John Lee · Cheng-Few Lee

Essentials of Excel VBA, Python, and R

Volume I: Financial Statistics and Portfolio Analysis

Second Edition

Springer

Essentials of Excel VBA, Python, and R

John Lee • Cheng-Few Lee

Essentials of Excel VBA, Python, and R

Volume I: Financial Statistics and Portfolio Analysis

Second Edition

Springer

John Lee
Center for PBBEF Research
Morris Plains, NJ, USA

Cheng-Few Lee
Rutgers Business School
The State University of New Jersey
North Brunswick, NJ, USA

ISBN 978-3-031-14235-2 ISBN 978-3-031-14236-9 (eBook)
https://doi.org/10.1007/978-3-031-14236-9

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

In the new edition of this book, there are 49 chapters, and they are divided into two volumes. Volume I, entitled "Essentials of Excel VBA, Python, and R: Volume I: Financial Statistics and Portfolio Analysis," contains 26 chapters. Volume II entitled, "Microsoft Excel VBA, Python and R For Financial Derivatives, Financial Management, and Machine Learning," contains 23 chapters. Volume I is divided into two parts: Part I, Financial Statistics, contains 21 chapters and Part II, Portfolio Analysis, contains five chapters. Volume II is divided into five parts: Part I, Excel VBA, contains three chapters; Part II, Financial Derivatives, contains six chapters; Part III, Applications of Python, Machine Learning for Financial Derivatives, and Risk Management, contains six chapters; Part IV, Financial Management, contains four chapters; and Part V, Applications of R Programs for Financial Analysis and Derivatives, contains three chapters.

In Volume I, we replaced Minitab and SAS, which were used in the first edition, with Python and R. In addition, we used individual companies' updated stock price's EPS, DPS, rate of return, and market rate of return to show how to understand the tool of statistical analysis. Therefore, these 26 chapters not only show how to understand statistical analysis' methodology but also show how to analyze the data of an individual stock and market index. Hence, these 26 chapters lay the foundation for not only financial statistics but also portfolio analysis, which is presented in Part II. In sum, Volume I can be regarded as the most comprehensive book for finance majors to understand that not only statistical tools are needed but also how to use Excel VBA, Python, and R to perform portfolio analysis.

There are four possible applications of this book:

A. To supplement business statistics book, especially for a book such as "Statistics for Business and Financial Economics, 3rd edition" (by Cheng-Few Lee, John C. Lee, and Alice C. Lee)
B. Chapters 1–21 can be used independently to teach statistics courses. Under this approach, the instructor can use data and three alternative computer programs to teach a course. In other words, this approach can be regarded as a data analysis approach to teach statistics.
C. This book can be used to learn how to apply Excel VBA, Python, and R to financial data analysis.
D. This book can be used to supplement the courses, such as portfolio management and financial management.

This book can be used by academic courses and for practitioners in the financial industry. Finally, we appreciate the extensive help of our assistants Xiaoyi Huang and Natalie Krawczyk.

Morris Plains, USA — John Lee
North Brunswick, USA — Cheng-Few Lee
2022

Contents

1 Introduction . . . 1
- 1.1 Introduction . . . 1
- 1.2 Microsoft Excel 2019 Versus Microsoft Excel 365 . . . 2
- 1.3 Power Query . . . 2
- 1.4 Microsoft Excel and Power Query . . . 3
- 1.5 Microsoft Excel 64-Bit Versus Microsoft Excel 32-Bit . . . 3
  - 1.5.1 Microsoft Excel 64-Bit and Power Query . . . 5
- 1.6 Statistical Environment of Microsoft Excel 365 . . . 5
- 1.7 Python Programming Language . . . 8
  - 1.7.1 Python Libraries for Statistics . . . 9
  - 1.7.2 Python Development Environment . . . 9
- 1.8 *R* Programming Language . . . 10
- 1.9 Web Scraping for Market and Financial Data . . . 11
  - 1.9.1 Microsoft Excel Power Query . . . 11
- 1.10 Case Study, Google Study, and Active Study Approach . . . 14
- 1.11 Structure of the Book . . . 15

Bibliography . . . 15

Part I Financial Statistics

2 Data Collection, Presentation, and Yahoo! Finance . . . 19
- 2.1 Introduction . . . 19
- 2.2 Data Presentation . . . 19
- 2.3 Yahoo! Finance . . . 19
- 2.4 Market Indexes . . . 20
  - 2.4.1 Dow Jones Industrial Average . . . 20
  - 2.4.2 S&P 500 . . . 20
  - 2.4.3 NASDAQ . . . 20
- 2.5 JSON Data Format . . . 21
  - 2.5.1 Quandl Data Provider . . . 21
- 2.6 Ticker Attributes . . . 25
  - 2.6.1 Yahoo! Finance API . . . 25
- 2.7 Historical Data . . . 44
  - 2.7.1 Yahoo! Finance API . . . 44
  - 2.7.2 Epoch Time . . . 45
  - 2.7.3 Power Query . . . 46
  - 2.7.4 Python . . . 65
- 2.8 Charting Historical Data . . . 67
  - 2.8.1 Microsoft Excel 365 Chart Wizard . . . 67
  - 2.8.2 Power Query M Code . . . 73

2.9 Using Python to Graph Johnson & Johnson's Historical Prices 78
2.10 Summary 80
Bibliography 80

3 Histograms, Rate of Returns, and Financial Statements 81
3.1 Introduction 81
3.2 Rate of Return 81
3.2.1 Power Query 81
3.2.2 Dynamic Power Query 98
3.2.3 Python 102
3.3 Histograms 104
3.3.1 Sturge's Rule 104
3.3.2 Microsoft Excel 105
3.4 Using Python to Create Johnson & Johnson's Rate of Return Histogram 111
3.4.1 Code 111
3.4.2 Output 113
3.5 Financial Statements 114
3.5.1 Power Query 114
3.5.2 Python 129
3.6 Summary 132
Bibliography 132

4 Numerical Summary Measures on Rate of Returns of Stocks and Market Indexes 133
4.1 Introduction 133
4.1.1 Summary Measures Excel Workbook 133
4.1.2 Summary Worksheet 133
4.2 Measure of Central Tendency 143
4.2.1 Arithmetic Mean (Average) 143
4.2.2 Annualized Monthly Returns 143
4.2.3 Median 143
4.2.4 Excel Functions 143
4.3 Measure of Dispersion 144
4.3.1 Variance 144
4.3.2 Annualized Monthly Variance 145
4.3.3 Standard Deviation 145
4.3.4 Annualized Monthly Standard Deviation 145
4.3.5 Coefficient of Variation 146
4.3.6 Excel Functions 146
4.4 Measure of Relative Position 147
4.4.1 Quartiles 147
4.4.2 Interquartile 147
4.4.3 Outliers 147
4.4.4 Z-Score 147
4.4.5 Excel Functions 147
4.5 Measure of Shape 148
4.5.1 Skewness 148
4.5.2 Kurtosis 149
4.5.3 Excel Functions 149
4.6 Measure of Linear Relationship 150
4.6.1 Coefficient of Correlation 150
4.6.2 Excel Functions 150

4.7 Box and Whisker Plot . . . 151
4.7.1 Outliers . . . 151
4.7.2 Extreme Outliers . . . 152
4.7.3 Vertical Whiskers . . . 152
4.7.4 Median and Mean . . . 152
4.8 Excel Rate of Return Box and Whisker Workbook . . . 152
4.8.1 Summary Worksheet . . . 152
4.8.2 Ticker1, Ticker2, Ticker3 Worksheets . . . 153
4.8.3 Ticker123 Worksheet . . . 154
4.9 Creating Box and Whisker Plot in Excel . . . 155
4.9.1 Single Box and Whisker Plot . . . 155
4.9.2 Combined Box and Whisker Plot . . . 157
4.10 Using Python to Calculate the 5-Year Numerical Measures of the Rate of Return of AAPL, MSFT, and the S&P 500 . . . 163
4.11 Summary . . . 167
Bibliography . . . 167

5 Probability Concepts and Their Analysis . . . 169
5.1 Introduction . . . 169
5.2 Data Presentation . . . 169
5.3 Probability . . . 169
5.3.1 Probability Simulation with Excel VBA . . . 170
5.3.2 Probability Simulation in R . . . 180
5.3.3 Probability Simulation in Python . . . 184
5.4 Combinations . . . 188
5.4.1 Combination List with Excel VBA . . . 189
5.4.2 Combination List with R . . . 191
5.5 Permutations . . . 191
5.5.1 Permutation List with Excel VBA . . . 192
5.5.2 Permutation List with R . . . 194
5.6 Summary . . . 195
Bibliography . . . 195

6 Discrete Random Variables and Probability Distributions . . . 197
6.1 Introduction and Probability Distribution . . . 197
6.2 Cumulative Probability Distribution . . . 198
6.3 Binomial Distribution . . . 202
6.3.1 Binomial Distribution in Excel . . . 203
6.3.2 Binomial Distribution in R . . . 207
6.3.3 Binomial Distribution in Python . . . 210
6.4 Poisson Random Variable . . . 211
6.4.1 Poisson Distribution in Excel . . . 212
6.4.2 Poisson Distribution in R . . . 215
6.4.3 Poisson Distribution in Python . . . 217
6.5 Excel 4.0 Macro Functions and Excel Names . . . 218
6.6 Examples . . . 219
6.6.1 Files Function . . . 219
6.7 Stephen Bullen's Charting Method . . . 224
6.7.1 Binomial Distribution . . . 228
6.7.2 Poisson Distribution . . . 229
6.8 Summary . . . 230
Bibliography . . . 230

7 Normal and Lognormal Distributions . . . 231
7.1 Introduction . . . 231
7.2 Uniform Distribution . . . 231
7.2.1 Uniform Distribution in *R* . . . 231
7.3 Normal Distribution . . . 234
7.3.1 Normal Distribution in *R* . . . 234
7.3.2 Normal Distribution in *Python* . . . 237
7.4 Standard Normal Distribution . . . 239
7.4.1 Standard Normal Distribution in *R* . . . 239
7.4.2 Standard Normal Distribution in Excel . . . 240
7.5 Lognormal Distribution . . . 243
7.5.1 Lognormal Distribution in R . . . 243
7.5.2 Lognormal Distribution in Python . . . 246
7.6 Normal Quantile–Quantile (QQ) Plot in Excel . . . 247
7.7 Normal Quantile–Quantile (QQ) Plot in Python . . . 250
7.8 Summary . . . 253
Bibliography . . . 254

8 Sampling Distributions and Central Limit Theorem . . . 255
8.1 Introduction . . . 255
8.2 Sample Distribution in Excel . . . 255
8.3 Mean of Sample Distribution Equals Mean of Population . . . 261
8.4 Sample Distribution in Python . . . 266
8.5 Central Limit Theorem . . . 267
8.5.1 Uniform Distribution in *R* . . . 268
8.5.2 Normal Distribution in *R* . . . 270
8.5.3 Lognormal Distribution in *R* . . . 272
8.5.4 Binomial Distribution in *R* . . . 273
8.5.5 Poisson Distribution in *R* . . . 275
8.6 Summary . . . 277
Bibliography . . . 277

9 Other Continuous Distributions . . . 279
9.1 Introduction . . . 279
9.2 *t*-Distribution . . . 279
9.2.1 *t*-Distribution in *R* . . . 279
9.2.2 *t*-Distribution in *Python* . . . 280
9.2.3 Student's *t*-Distribution in Excel . . . 281
9.3 Chi-Square (χ^2) Distribution . . . 284
9.3.1 Chi-Square (χ^2) Distribution in *R* . . . 285
9.3.2 Chi-Square (χ^2) Distribution in *Python* . . . 286
9.3.3 Chi-Square (χ^2) Distribution in Excel . . . 287
9.4 *F*-Distribution . . . 290
9.4.1 *F*-Distribution in *R* . . . 290
9.4.2 *F*-Distribution in *Python* . . . 291
9.4.3 *F*-Distribution in Excel . . . 292
9.5 Exponential Distribution . . . 296
9.5.1 Exponential Probability Density Function in Excel . . . 296
9.5.2 Exponential Cumulative Density Function in Excel . . . 300
9.6 Summary . . . 301
Bibliography . . . 301

10 Estimation . . . 303
10.1 Introduction . . . 303
10.2 Confidence Interval Simulation in Python . . . 304
10.2.1 Python Code . . . 305
10.2.2 Confidence Interval Simulation Data . . . 307
10.3 Interval Estimates for μ When σ^2 is Known . . . 308
10.3.1 Z Confidence Intervals . . . 308
10.4 Confidence Intervals for μ When σ^2 is Unknown . . . 311
10.4.1 T Confidence Intervals . . . 311
10.5 Confidence Intervals for the Population Proportion . . . 312
10.5.1 Example G . . . 312
10.5.2 Example H . . . 313
10.5.3 Example I . . . 314
10.5.4 Example J . . . 314
10.6 Confidence Intervals for the Variance . . . 315
10.6.1 Example K . . . 315
10.7 Summary . . . 316
Bibliography . . . 316

11 Hypothesis Testing . . . 317
11.1 Introduction . . . 317
11.2 One-Tailed Tests of Mean for Large Samples . . . 317
11.2.1 Example 11.1 . . . 318
11.3 Z-Test . . . 318
11.4 Hypothesis Testing and the *p*-Value . . . 319
11.4.1 Example 11.2 . . . 319
11.5 One-Tailed Tests of Mean for Large Samples: Two-Sample Test of Means . . . 320
11.5.1 Example 11.3 . . . 320
11.6 Two-Tailed Tests of Mean for Large Samples . . . 322
11.6.1 Example 11.4 . . . 322
11.6.2 Example 11.5 . . . 323
11.7 One-Tailed Tests of Mean for Small Samples . . . 324
11.7.1 Example 11.6 . . . 324
11.8 Hypothesis Testing for a Population Proportion . . . 325
11.8.1 Example 11.7 . . . 325
11.9 The Power of a Test and Power Function . . . 326
11.9.1 Example 11.8 . . . 326
11.10 Power and Sample Size . . . 327
11.11 Power and Alpha Size . . . 327
11.12 Comparing the Average EPS of AAPL and MSFT in Python . . . 328
11.13 Summary . . . 330
Bibliography . . . 330

12 Analysis of Variance and Chi-Square Tests . . . 331
12.1 Introduction . . . 331
12.2 One-Way Analysis of Variance . . . 333
12.2.1 Example 12.1 . . . 333
12.2.2 Example 12.2 . . . 341
12.3 Two-Way Analysis of Variance . . . 344
12.3.1 Example 12.3 . . . 344
12.4 Chi-Square Test . . . 347

12.5 Goodness of Fit . . . 347
12.5.1 Example 12.4 . . . 348
12.6 Test of Independence . . . 349
12.6.1 Example 12.5 . . . 349
12.7 Using the Chi-Square Test and Python to Determine if the Rate of Return of Apple Inc. Is a Normal Distribution . . . 350
12.8 Summary . . . 352
Bibliography . . . 352

13 Simple Linear Regression and the Correlation Coefficient . . . 353
13.1 Introduction . . . 353
13.2 Regression Analysis . . . 353
13.3 Retrieving Data Using Power Query . . . 356
13.4 Combining Power Query Data Sets . . . 359
13.5 Scatter Chart . . . 361
13.6 Deterministic Relationship and Stochastic Relationship . . . 364
13.7 Least Square Method . . . 365
13.8 Standard Assumptions for Linear Regression . . . 365
13.9 Standard Error of Estimate . . . 366
13.10 The Coefficient of Determination . . . 367
13.11 Correlation Coefficient . . . 368
13.12 Regression Analysis in Excel . . . 370
13.12.1 Correlation and Coefficient of Determination . . . 372
13.12.2 Regression Line . . . 373
13.12.3 Residuals of the Regression Line . . . 373
13.12.4 Fit Plot of the Data Set . . . 374
13.13 *INTERCEPT* and *SLOPE* Excel Functions . . . 374
13.14 Oil and Gasoline Regression Analysis in Python . . . 375
13.15 Summary . . . 378
Bibliography . . . 378

14 Simple Linear Regression and Correlation: Analyses and Applications . . . 379
14.1 Introduction . . . 379
14.2 Standard Error of Estimate . . . 380
14.3 Two-Tailed t-Test for β . . . 380
14.4 Two-Tailed t-Test for α . . . 382
14.5 Confidence Interval of β . . . 384
14.6 F Test . . . 384
14.7 The Relationship Between the F-Test and the t-Test . . . 385
14.8 Market Model . . . 386
14.9 Yahoo! Finance Beta Screener . . . 386
14.10 Historical Monthly Data from Yahoo! Finance . . . 386
14.10.1 Excel's Import Text Wizard . . . 387
14.11 Market Model of Apple Inc. in Excel . . . 389
14.11.1 Data Analysis and Regression Report . . . 390
14.11.2 Yahoo! Finance Beta and Power Query . . . 392
14.11.3 Yahoo! Finance Ticker Historical Data . . . 393
14.11.4 Yahoo! Finance S&P500 Historical Data . . . 394
14.11.5 Calculating Rate of Return . . . 394
14.11.6 Date, Time, and Epoch Time . . . 396
14.11.7 Converting to and from Epoch Time . . . 401
14.11.8 Other Power Queries . . . 402

14.12 Market Model of the Clorox Company in Excel . . . 405
14.12.1 Regression Report . . . 405
14.12.2 Yahoo! Finance Beta and the Market Model . . . 406
14.12.3 Sectors and Industry . . . 407
14.13 Market Model in Python . . . 408
14.14 Summary . . . 410
Bibliography . . . 410

15 Multiple Linear Regression . . . 411
15.1 Introduction . . . 411
15.2 R-Square . . . 413
15.3 *F*-Test . . . 414
15.4 Confidence Interval of B . . . 415
15.5 *t*-Test . . . 415
15.6 Analyzing the Determination of Price Per Share . . . 416
15.6.1 Regression Analysis . . . 416
15.6.2 Workbook Sources . . . 419
15.6.3 Data Source . . . 421
15.7 Power Query Resource Issue . . . 427
15.8 Excel 365 and OneDrive . . . 428
15.9 Using *R* to Predict . . . 431
15.10 Summary . . . 432
Bibliography . . . 433

16 Residual and Regression Assumption Analysis . . . 435
16.1 Introduction . . . 435
16.2 Regression Analysis . . . 435
16.3 Linearity . . . 438
16.4 The Expected Value of the Residual Term is Zero . . . 441
16.5 The Variance of the Error Term is Constant . . . 443
16.6 Autocorrelation Durbin–Watson Test . . . 450
16.6.1 VBA Code . . . 450
16.6.2 Durbin–Watson 1% Table . . . 452
16.7 Autocorrelation Walmart's Dividend and EPS from 2019 to 2000 . . . 453
16.7.1 Data Source . . . 454
16.8 Using VBA to Retrieve a Ticker's Name . . . 455
16.9 Durbin–Watson Test Market Model Python Code . . . 458
16.10 The Independent Variables Are Uncorrelated: Multicollinearity . . . 459
16.11 Variance Inflationary Factor (VIF) . . . 461
16.12 Summary . . . 463
Bibliography . . . 463

17 Nonparametric Statistics . . . 465
17.1 Introduction . . . 465
17.2 Mann–Whitney *U* Test . . . 465
17.2.1 Calculation in Microsoft Excel . . . 466
17.2.2 Calculation in *R* . . . 468
17.3 Kruskal–Wallis Test . . . 468
17.3.1 Calculation in Microsoft Excel . . . 469
17.3.2 Calculation in *R* . . . 470
17.3.3 Calculation in Python . . . 471
17.4 Spearman's Rank Correlation Test . . . 472

17.4.1 Calculation in *R* . . . 474
17.4.2 Calculation in Python . . . 475
17.5 Using Python to Test the Randomness of the Rate of Return of JNJ . . . 476
17.6 Using Python to Test the Randomness of the Rate of Return of MSFT . . . 478
17.7 Summary . . . 479
Bibliography . . . 479

18 Time Series: Analysis, Model, and Forecasting . . . 481
18.1 Introduction . . . 481
18.2 Moving Averages . . . 481
18.2.1 Moving Averages in Excel . . . 482
18.2.2 Moving Averages in *R* . . . 490
18.2.3 Moving Averages in Python . . . 491
18.2.4 Data Source . . . 492
18.3 Linear Trend . . . 493
18.3.1 Linear Trend Analysis in Excel . . . 493
18.3.2 Linear Trend Analysis in R . . . 499
18.3.3 Linear Trend Analysis in Python . . . 502
18.4 Exponential Smoothing . . . 507
18.4.1 Exponential Smoothing in Excel . . . 508
18.4.2 Exponential Smoothing in Python . . . 510
18.5 Summary . . . 511
Bibliography . . . 511

19 Index Numbers and Stock Market Indexes . . . 513
19.1 Introduction . . . 513
19.2 Simple Price Index . . . 513
19.2.1 Example 19.1 . . . 513
19.3 Laspeyres Price Index . . . 518
19.4 Paasche Price Index . . . 519
19.5 Fisher's Ideal Price Index . . . 520
19.6 Stock Indexes: S&P500 Index and NASDAQ Composite Index . . . 521
19.7 Stock Indexes: Dow Jones Industrial Average (DJIA) . . . 522
19.8 Components of the Dow Jones Industrial Average (DJIA) . . . 524
19.8.1 Using Power Query to Retrieve the Dow 30 Components . . . 525
19.9 Components of the S&P 500 Index . . . 526
19.9.1 Using Power Query to Retrieve the S&P 500 Components . . . 526
19.10 Components of the NASDAQ Composite Index . . . 530
19.10.1 Using Power Query to Retrieve the NASDAQ Composite Components . . . 533
19.11 Using Python to Calculate the Four Statistical Moments of the Rate of Returns of Every Component in the S&P 500 . . . 533
19.12 Summary . . . 536
Bibliography . . . 536

20 Sampling Surveys: Methods and Applications . . . 537
20.1 Introduction . . . 537
20.2 Random Number Tables . . . 537
20.2.1 Excel VBA . . . 537
20.2.2 Python Code . . . 538

20.3 Confidence Interval for the Population Mean . . . 538
20.3.1 Example 20.1 . . . 538
20.4 Confidence Interval for the Population Proportion . . . 540
20.4.1 Example 20.2 . . . 540
20.5 Determining Sample Size . . . 541
20.5.1 Example 20.3 . . . 541
20.6 Summary . . . 541
Bibliography . . . 542

21 Statistical Decision Theory . . . 543
21.1 Introduction . . . 543
21.2 Decision Trees and Expected Monetary Values . . . 543
21.3 NPV and IRR Method for Capital Budgeting Decision Under Certainty . . . 543
21.4 The Statistical Distribution Method . . . 547
21.4.1 Methodology . . . 547
21.4.2 Excel and VBA Application . . . 552
21.5 Summary . . . 555
Bibliography . . . 556

Part II Portfolio Analysis

22 Risk Classification, Estimation, and Diversification . . . 559
22.1 Introduction . . . 559
22.2 Risk Classification . . . 559
22.2.1 Business Risk . . . 559
22.2.2 Financial Risk . . . 561
22.2.3 Total Risk . . . 561
22.3 Portfolio Analysis and Application . . . 562
22.3.1 Expected Rate of Return on a Portfolio . . . 562
22.3.2 The Two-Asset Case . . . 564
22.3.3 The N-asset Case . . . 565
22.3.4 The Efficient Portfolios . . . 565
22.3.5 Corporate Application of Diversification . . . 568
22.4 Determination of Commercial Lending Rates . . . 568
22.5 The Dominance Principle and Performance EVALUATION . . . 570
22.6 Summary . . . 572
Bibliography . . . 586

23 Asset Allocation and Markowitz Portfolio-Selection Model . . . 587
23.1 Introduction . . . 587
23.2 Utility Theory, Utility Functions, and Indifference Curves . . . 587
23.2.1 Utility Theory . . . 588
23.2.2 Utility Functions . . . 588
23.2.3 Risk Aversion and Asset Allocation . . . 593
23.2.4 Indifference Curves . . . 594
23.3 Efficient Portfolios . . . 595
23.3.1 Portfolio Combinations . . . 596
23.3.2 Short Selling . . . 596
23.4 Techniques for Calculating the Efficient Frontier with Short Selling . . . 599
23.4.1 The Normal Distribution . . . 599
23.4.2 The Log-Normal Distribution . . . 600
23.4.3 Mathematical Method to Calculate Efficient Frontier . . . 601

23.4.4 Portfolio Determination with Specific Adjustment for Short Selling . . . 603
23.4.5 Portfolio Determination Without Short Selling . . . 605
23.5 Summary . . . 605
Bibliogrphy . . . 615

24 Capm, Beta Estimation, and Forecasting . . . 617
24.1 Introduction . . . 617
24.2 A Graphical Approach to the Derivation of the Capm . . . 617
24.2.1 The Lending, Borrowing, and Market Portfolios . . . 617
24.2.2 The Capital Market Line . . . 618
24.2.3 The Security Market Line—The Capital Asset Pricing Model . . . 620
24.3 Mathematical Approach to the Derivation of the Capm . . . 622
24.4 The Market Model and Risk Decomposition . . . 623
24.4.1 The Market Model . . . 623
24.4.2 Risk Decomposition . . . 623
24.4.3 Why Beta is Important for Security Analysis . . . 624
24.4.4 Determination of Systematic Risk . . . 625
24.5 Growth Rates, Accounting Betas, and Variance in Ebit . . . 626
24.5.1 Sustainable Growth Rates . . . 626
24.5.2 Accounting Beta . . . 628
24.5.3 Variance in EBIT . . . 628
24.5.4 Capital–Labor Ratio . . . 628
24.5.5 Fixed Costs and Variable Costs . . . 629
24.5.6 Beta Forecasting . . . 629
24.5.7 Market-Based Versus Accounting-Based Beta Forecasting . . . 630
24.6 Some Applications and Implications of the Capm . . . 633
24.7 Summary . . . 634
Bibliography . . . 640

25 Portfolio Selection Methods: Theory and Application . . . 643
25.1 Introduction . . . 643
25.2 The Single-Index Model . . . 643
25.2.1 Deriving the Single-Index Model . . . 645
25.2.2 Portfolio Analysis and the Single-Index Model <S2> . . . 648
25.2.3 The Market Model and Beta . . . 652
25.3 Multiple Indexes and the Multiple-Index Model . . . 653
25.4 Summary . . . 656
Bibliography . . . 668

26 Investment Performance Approach to Portfolio Selection . . . 671
26.1 Introduction . . . 671
26.2 Sharpe Performance-Measure Approach with Short Sales Allowed . . . 671
26.3 Sharpe Performance-Measure Approach with Short Sales and Upper Bound Constraints . . . 677
26.4 Treynor-Measure Approach with Short Sales Allowed . . . 679
26.5 Treynor-Measure Approach with Short Sales not Allowed . . . 681
26.6 Impact of Short Sales on Optimal-Weight Determination . . . 684
26.7 Economic Rationale of the Treynor Performance-Measure Method . . . 684
26.8 Summary . . . 685
Bibliography . . . 695

Introduction

1

1.1 Introduction

A person studies *statistics* to learn how to obtain information from a data set. Therefore, the discipline of statistics transforms data into information. But one big problem in studying statistics is that it is a very intense computational discipline. We can get a sense of the computational intensity of statistics by looking at the average statistical formula and the correlation coefficient statistical formula shown below:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} \qquad r = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{1/2}\left[\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2\right]^{1/2}}$$

Average *Correlation Coefficient*

The average formula looks fairly simple, but it becomes more and more difficult as n increases. The correlation coefficient formula is more complex and tedious to perform—and other computational formulas are even more complicated. Therefore, one of the biggest costs in both time and money in studying and using statistics is implementing statistical concepts on data sets that we are interested in studying, analyzing, and understanding. Statistical computer programs have been developed to combat these implementation costs. Some of the more popular statistical programs are SAS, SPSS, and R. A popular programming language to do statistical analysis is Python programming language. It is popular because it has free and powerful statistical libraries. In this book, we will look at Python and R.

Traditionally, schools use a statistical program to accompany the study of statistics. One problem with this is that most students will not become professional statisticians; this means that all the time and effort they spent learning how to use the statistical program has been wasted. And, if they do ever need to do some statistical analysis, the statistical program they did use in school is probably not available.

One solution to the above problem is to use Microsoft Excel as the computer program to do the statistical analysis. Excel can do statistical analysis. While Excel's statistical features are not as strong or as sophisticated as the traditional statistical programs, it does take care of 90% of the statistical analysis done in the business world. One of the most important characteristics of Excel is that it is virtually everywhere; this is important because it allows a person to perform the statistical analysis right away instead of spending time, effort, and money to acquire a traditional statistical program. Many times it could be weeks before they get the statistical program.

The benefits of using Excel have been recognized in the academic world; this is why you will find more and more statistical books using Excel to do the statistical analysis.

This book will use Microsoft Excel, Python, and R to demonstrate statistical concepts. We use Microsoft Excel because of the benefits stated above. We will use Python and R to double-check our work in Excel and do some statistical analysis that is impossible or not practical to do in Excel.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_1

In this book, we will use Microsoft Excel 365, Python 3.9, and R 4.0.3.

Most of the concepts demonstrated in this book will work in Excel 2016, Excel 2019, and Excel 365. This book will use Excel 365.

Section 1.2 discusses Microsoft Excel 2019 versus Microsoft Excel 365, Sect. 1.3 discusses power query, Sect. 1.4 discusses Microsoft Excel and power query, and Sect. 1.5 compares Microsoft Excel 64-bit and Microsoft 32-bit. In Sect. 1.6, the statistical environment of Microsoft Excel 365 is talked about. Python programming language is discussed in Sect. 1.7, while R programming language is discussed in Sect. 1.8. In Sect. 1.9, web scraping for market and financial data is discussed. A case study, Google study, and an active study approach are discussed in Sect. 1.10, and finally, in Sect. 1.11, we go over the structure of the book.

1.2 Microsoft Excel 2019 Versus Microsoft Excel 365

There are many versions of Excel currently in use. Some of the more popular versions are Microsoft Excel 2016, Microsoft Excel 2019, and Excel 365. As of July 2021, there are two current versions of Excel. The two current versions of Excel are Microsoft Excel 2019 and Microsoft Excel 365. At the core, Excel 2019 and Excel 365 have the same features. One main fundamental difference between Excel 2019 and Excel 365 is that Excel 365 will be upgraded with new features over time while Excel 2019 will not. Excel 2019 will be upgraded with only stability and bug fixes. The URL https://www.windowscentral.com/office-2019-or-office-365-which-one-you-should-get and the URL https://support.microsoft.com/en-us/office/what-s-the-difference-between-microsoft-365-and-office-2019-ed447ebf-6060-46f9-9e90-a239bd27eb96 discusses in detail the difference between Microsoft Excel 2019 and Microsoft Excel 365.

There are two notable non-Excel benefits by subscribing to Excel 365 versus buying Excel 2019. The first important non-Excel benefit of Excel 365 is that you can install Excel 365 on all your devices and sign in to five simultaneously. With Excel 365, you can install Excel 365 to more than one PC or more than one Mac. In contrast with Excel 2019, you can only install it on one PC or one Mac.

The second non-Excel benefit of Excel 365 is you get 1 terabyte of OneDrive cloud storage. With OneDrive, you can set it up, so you can sync your Excel files on your computer with OneDrive; this way, you can always work on your Excel file on any device.

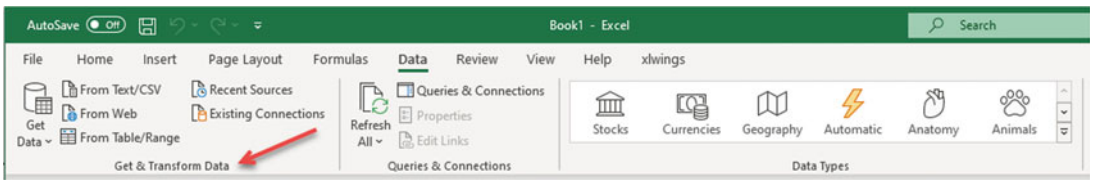
1.3 Power Query

Tableau and *Power BI* has recently been the hot applications in the data analytics world. Power BI is a Microsoft product. Power BI accomplishes its data analytics with *Power Query*. The URL https://docs.microsoft.com/en-us/power-query/power-query-what-is-power-query defines *Power Query* as a "data transformation and data preparation engine." This URL also indicates that with *Power Query*, "you can perform the extract, transform, and load (ETL) processing of data." Below is a table that lists Microsoft products and services where *Power Query* can be found.

Product	M engine[1]	Power Query Desktop[2]	Power Query Online[3]	Dataflows[4]
Excel for Windows	Yes	Yes	No	No
Excel for Mac	Yes	No	No	No
Power BI	Yes	Yes	Yes	Yes
Power Apps	Yes	No	Yes	Yes
Power Automate	Yes	No	Yes	No
Azure Data Factory	Yes	No	Yes	Yes
SQL Server Integration Services	Yes	No	No	No
SQL Server Analysis Services	Yes	Yes	No	No
Dynamics 365 Customer Insights	Yes	No	Yes	Yes

[1]M engine	The underlying query execution engine that runs queries expressed in the Power Query formula language ("M").
[2]Power Query Desktop	The Power Query experience found in desktop applications.
[3]Power Query Online	The Power Query experience found in web browser applications.
[4]Dataflows	Power Query as a service that runs in the cloud and is product-agnostic. The stored result can be used in other applications as services.

1.4 Microsoft Excel and Power Query



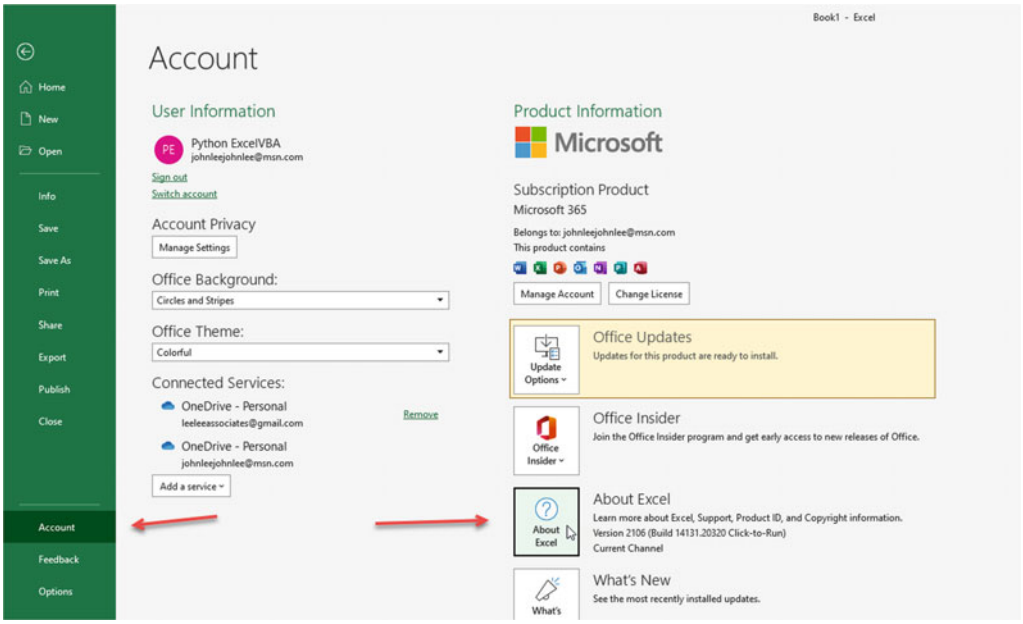
In Microsoft Excel 365 the *Get & Transform Data* section of the *Data* tab is *Power Query*. This book will use *Power Query* to retrieve data from the internet, and we will use *Power Query* to transform a data set. Because of *Power Query*, there is less need to do Excel VBA programming.

1.5 Microsoft Excel 64-Bit Versus Microsoft Excel 32-Bit

Until recently, software applications were available in 32-bit. Slowly more and more software applications are now available in 62-bit. The 64-bit applications can handle and process more data than the 32-bit applications. It is important to know if you are using Excel 32-bit or Excel 64-bit. The reason being 32-bit software applications can not communicate with 64-bit applications. Microsoft Excel 365 64-bit is installed by default in the retail market. But, in the large corporate world, 32-bit Excel is installed by default because large corporations have a lot of 32-bit applications that have not been converted to 64-bit.

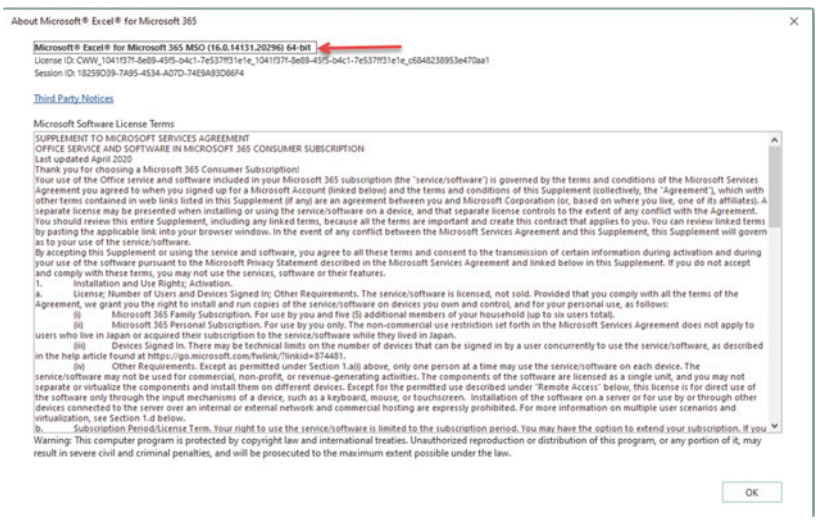
Do the following to see if you are using Microsoft Excel 64-bit or Microsoft Excel 32-bit.

(1) Click on *Account* from the *File* menu.



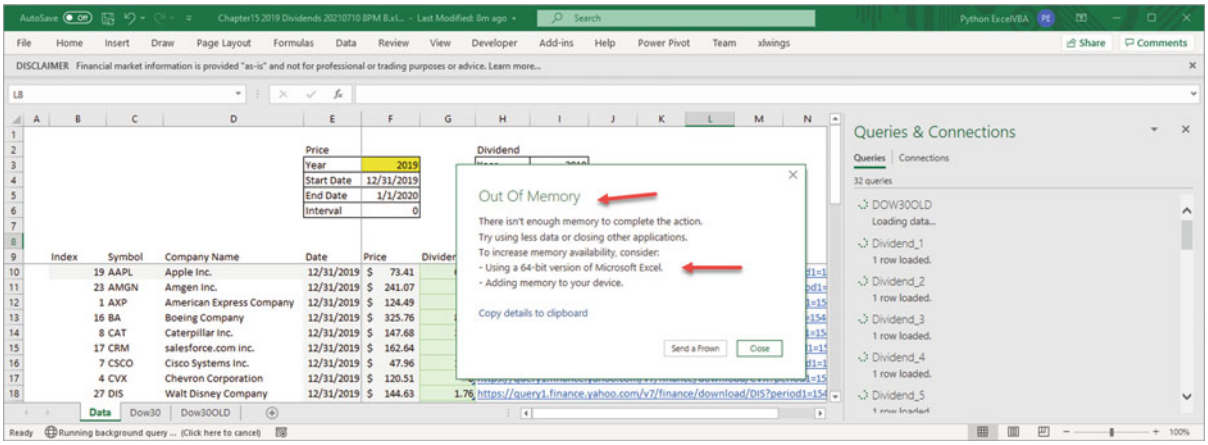
(2) Click on *About Excel*

The resulting screen will tell you if you have Excel 64-bit or Excel 32-bit.



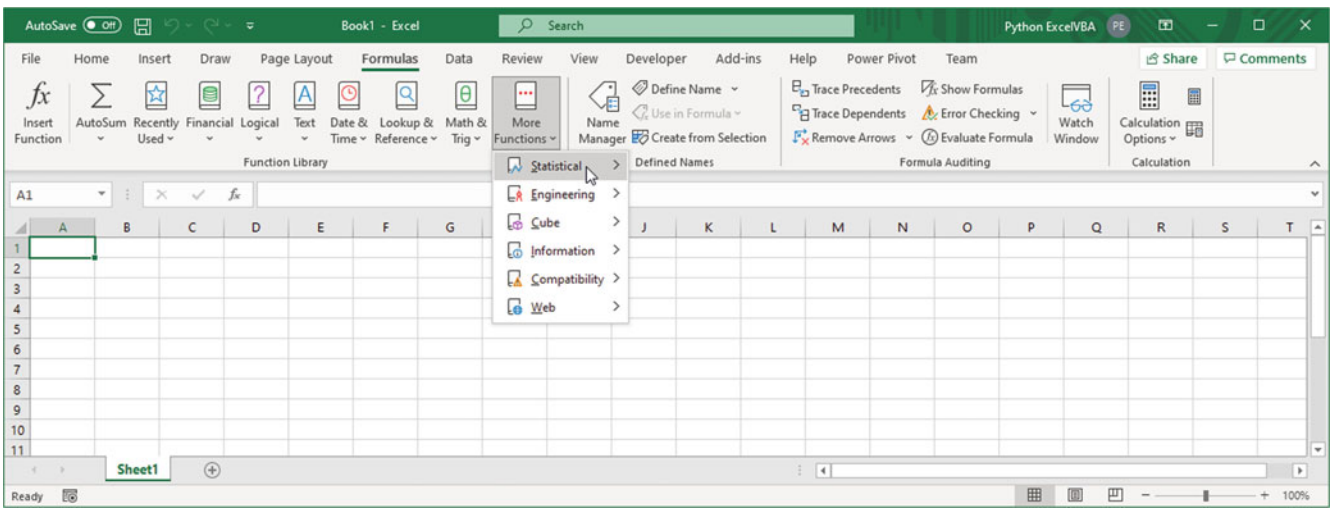
1.5.1 Microsoft Excel 64-Bit and Power Query

It's important to use Excel 64-bit if you are using Excel Power Query heavily. You will get the following message if you are using Excel 32-bit:



1.6 Statistical Environment of Microsoft Excel 365

Excel has many built-in statistical features. One statistical feature that Excel provides is a lot of statistical functions. To view the statistical functions, go to the *Formulas* tab and choose the *Statistical* item as shown below:



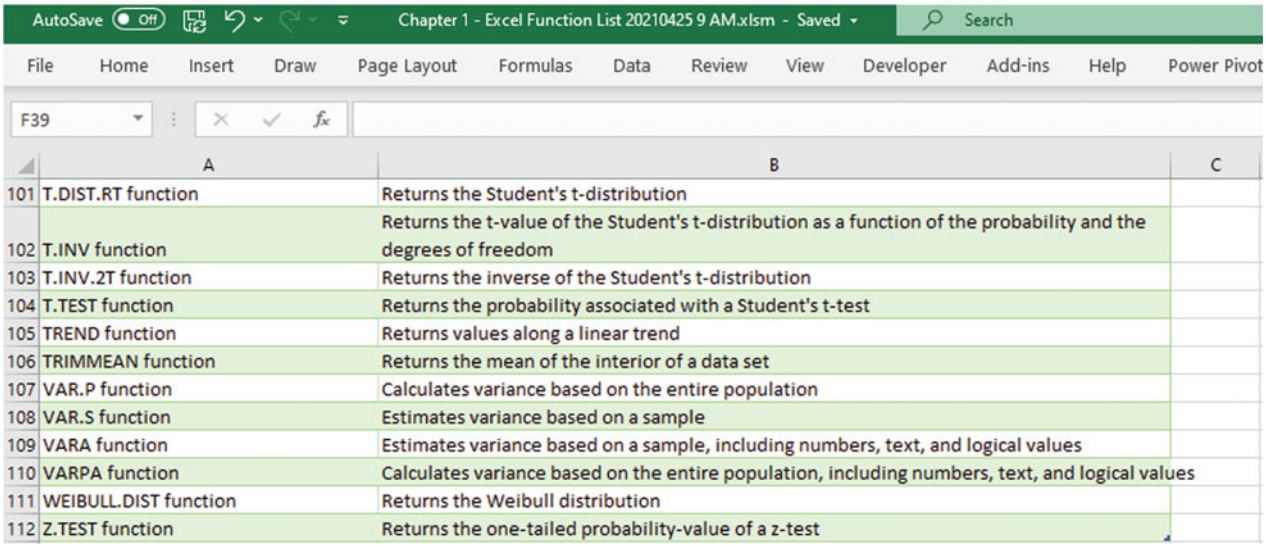
The URL https://www.excelfunctions.net/excel-statistical-functions.html is a good place to learn about the statistical Excel functions available. Microsoft is constantly adding new Excel statistical functions. This URL will tell you about new Excel statistical functions.

One nice feature of this website is it tells you when a specific statistical function became available and if a new statistical function replaces a previous older statistical function, and if the statistical function is available in the Excel Mac version.

The list below lists all the available statistical functions in Excel 365.

	A	B
1	Function	Description
2	AVEDEV function	Returns the average of the absolute deviations of data points from their mean
3	AVERAGE function	Returns the average of its arguments
4	AVERAGEA function	Returns the average of its arguments, including numbers, text, and logical values
5	AVERAGEIF function	Returns the average (arithmetic mean) of all the cells in a range that meet a given criteria
6	AVERAGEIFS function	Returns the average (arithmetic mean) of all cells that meet multiple criteria
7	BETA.DIST function	Returns the beta cumulative distribution function
8	BETA.INV function	Returns the inverse of the cumulative distribution function for a specified beta distribution
9	BINOM.DIST function	Returns the individual term binomial distribution probability
10	BINOM.DIST.RANGE function	Returns the probability of a trial result using a binomial distribution
11	BINOM.INV function	Returns the smallest value for which the cumulative binomial distribution is less than or equal to a criterion value
12	CHISQ.DIST function	Returns the cumulative beta probability density function
13	CHISQ.DIST.RT function	Returns the one-tailed probability of the chi-squared distribution
14	CHISQ.INV function	Returns the cumulative beta probability density function
15	CHISQ.INV.RT function	Returns the inverse of the one-tailed probability of the chi-squared distribution
16	CHISQ.TEST function	Returns the test for independence
17	CONFIDENCE.NORM function	Returns the confidence interval for a population mean
18	CONFIDENCE.T function	Returns the confidence interval for a population mean, using a Student's t distribution
19	CORREL function	Returns the correlation coefficient between two data sets
20	COUNT function	Counts how many numbers are in the list of arguments
21	COUNTA function	Counts how many values are in the list of arguments
22	COUNTBLANK function	Counts the number of blank cells within a range
23	COUNTIF function	Counts the number of cells within a range that meet the given criteria
24	COUNTIFS function	Counts the number of cells within a range that meet multiple criteria
25	COVARIANCE.P function	Returns covariance, the average of the products of paired deviations
26	COVARIANCE.S function	Returns the sample covariance, the average of the products deviations for each data point pair in two data sets
27	DEVSQ function	Returns the sum of squares of deviations
28	EXPON.DIST function	Returns the exponential distribution
29	F.DIST function	Returns the F probability distribution
30	F.DIST.RT function	Returns the F probability distribution
31	F.INV function	Returns the inverse of the F probability distribution
32	F.INV.RT function	Returns the inverse of the F probability distribution
33	F.TEST function	Returns the result of an F-test
34	FISHER function	Returns the Fisher transformation
35	FISHERINV function	Returns the inverse of the Fisher transformation
36	FORECAST function	Returns a value along a linear trend Note: In Excel 2016, this function is replaced with FORECAST.LINEAR as part of the new Forecasting functions, but it's still available for compatibility with earlier versions.
37	FORECAST.ETS function	Returns a future value based on existing (historical) values by using the AAA version of the Exponential Smoothing (ETS) algorithm
38	FORECAST.ETS.CONFINT function	Returns a confidence interval for the forecast value at the specified target date
39	FORECAST.ETS.SEASONALITY function	Returns the length of the repetitive pattern Excel detects for the specified time series
40	FORECAST.ETS.STAT function	Returns a statistical value as a result of time series forecasting
41	FORECAST.LINEAR function	Returns a future value based on existing values
42	FREQUENCY function	Returns a frequency distribution as a vertical array
43	GAMMA function	Returns the gamma function value
44	GAMMA.DIST function	Returns the gamma distribution
45	GAMMA.INV function	Returns the inverse of the gamma cumulative distribution
46	GAMMALN function	Returns the natural logarithm of the gamma function, Γ(x)
47	GAMMALN.PRECISE function	Returns the natural logarithm of the gamma function, Γ(x)
48	GAUSS function	Returns 0.5 less than the standard normal cumulative distribution
49	GEOMEAN function	Returns the geometric mean
50	GROWTH function	Returns values along an exponential trend

	A	B	C
51	HARMEAN function	Returns the harmonic mean	
52	HYPGEOM.DIST function	Returns the hypergeometric distribution	
53	INTERCEPT function	Returns the intercept of the linear regression line	
54	KURT function	Returns the kurtosis of a data set	
55	LARGE function	Returns the k-th largest value in a data set	
56	LINEST function	Returns the parameters of a linear trend	
57	LOGEST function	Returns the parameters of an exponential trend	
58	LOGNORM.DIST function	Returns the cumulative lognormal distribution	
59	LOGNORM.INV function	Returns the inverse of the lognormal cumulative distribution	
60	MAX function	Returns the maximum value in a list of arguments	
61	MAXA function	Returns the maximum value in a list of arguments, including numbers, text, and logical values	
62	MAXIFS function	Returns the maximum value among cells specified by a given set of conditions or criteria	
63	MEDIAN function	Returns the median of the given numbers	
64	MIN function	Returns the minimum value in a list of arguments	
65	MINIFS function	Returns the minimum value among cells specified by a given set of conditions or criteria.	
66	MINA function	Returns the smallest value in a list of arguments, including numbers, text, and logical values	
67	MODE.MULT function	Returns a vertical array of the most frequently occurring, or repetitive values in an array or range of data	
68	MODE.SNGL function	Returns the most common value in a data set	
69	NEGBINOM.DIST function	Returns the negative binomial distribution	
70	NORM.DIST function	Returns the normal cumulative distribution	
71	NORM.INV function	Returns the inverse of the normal cumulative distribution	
72	NORM.S.DIST function	Returns the standard normal cumulative distribution	
73	NORM.S.INV function	Returns the inverse of the standard normal cumulative distribution	
74	PEARSON function	Returns the Pearson product moment correlation coefficient	
75	PERCENTILE.EXC function	Returns the k-th percentile of values in a range, where k is in the range 0..1, exclusive.	
76	PERCENTILE.INC function	Returns the k-th percentile of values in a range	
77	PERCENTRANK.EXC function	Returns the rank of a value in a data set as a percentage (0..1, exclusive) of the data set	
78	PERCENTRANK.INC function	Returns the percentage rank of a value in a data set	
79	PERMUT function	Returns the number of permutations for a given number of objects	
80	PERMUTATIONA function	Returns the number of permutations for a given number of objects (with repetitions) that can be selected from the total objects	
81	PHI function	Returns the value of the density function for a standard normal distribution	
82	POISSON.DIST function	Returns the Poisson distribution	
83	PROB function	Returns the probability that values in a range are between two limits	
84	QUARTILE.EXC function	Returns the quartile of the data set, based on percentile values from 0..1, exclusive	
85	QUARTILE.INC function	Returns the quartile of a data set	
86	RANK.AVG function	Returns the rank of a number in a list of numbers	
87	RANK.EQ function	Returns the rank of a number in a list of numbers	
88	RSQ function	Returns the square of the Pearson product moment correlation coefficient	
89	SKEW function	Returns the skewness of a distribution	
90	SKEW.P function	Returns the skewness of a distribution based on a population: a characterization of the degree of asymmetry of a distribution around its mean	
91	SLOPE function	Returns the slope of the linear regression line	
92	SMALL function	Returns the k-th smallest value in a data set	
93	STANDARDIZE function	Returns a normalized value	
94	STDEV.P function	Calculates standard deviation based on the entire population	
95	STDEV.S function	Estimates standard deviation based on a sample	
96	STDEVA function	Estimates standard deviation based on a sample, including numbers, text, and logical values	
97	STDEVPA function	Calculates standard deviation based on the entire population, including numbers, text, and logical values	
98	STEYX function	Returns the standard error of the predicted y-value for each x in the regression	
99	T.DIST function	Returns the Percentage Points (probability) for the Student t-distribution	
100	T.DIST.2T function	Returns the Percentage Points (probability) for the Student t-distribution	



AutoSave Off | Chapter 1 - Excel Function List 20210425 9 AM.xlsm - Saved | Search

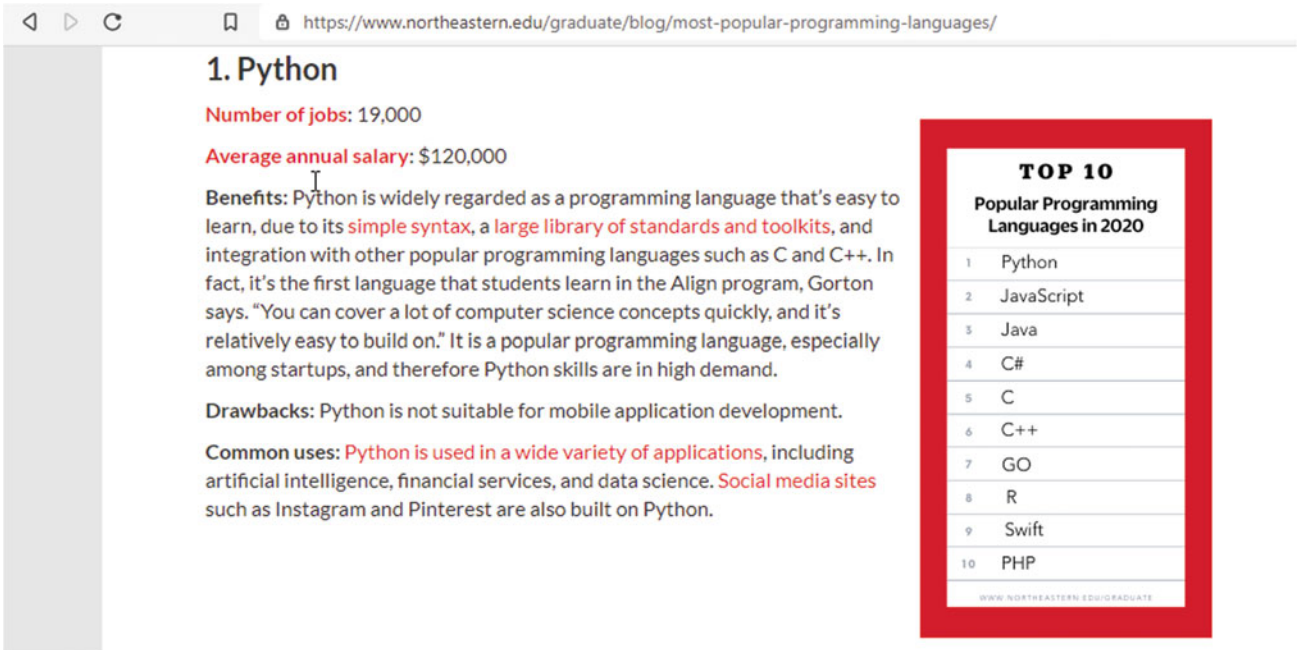
File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins Help Power Pivot

F39

	A	B	C
101	T.DIST.RT function	Returns the Student's t-distribution	
102	T.INV function	Returns the t-value of the Student's t-distribution as a function of the probability and the degrees of freedom	
103	T.INV.2T function	Returns the inverse of the Student's t-distribution	
104	T.TEST function	Returns the probability associated with a Student's t-test	
105	TREND function	Returns values along a linear trend	
106	TRIMMEAN function	Returns the mean of the interior of a data set	
107	VAR.P function	Calculates variance based on the entire population	
108	VAR.S function	Estimates variance based on a sample	
109	VARA function	Estimates variance based on a sample, including numbers, text, and logical values	
110	VARPA function	Calculates variance based on the entire population, including numbers, text, and logical values	
111	WEIBULL.DIST function	Returns the Weibull distribution	
112	Z.TEST function	Returns the one-tailed probability-value of a z-test	

1.7 Python Programming Language

Many websites rank Python as the most popular programming language. The URL https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/ ranks Python as the most popular programming language in 2021.



https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/

1. Python

Number of jobs: 19,000

Average annual salary: $120,000

Benefits: Python is widely regarded as a programming language that's easy to learn, due to its simple syntax, a large library of standards and toolkits, and integration with other popular programming languages such as C and C++. In fact, it's the first language that students learn in the Align program, Gorton says. "You can cover a lot of computer science concepts quickly, and it's relatively easy to build on." It is a popular programming language, especially among startups, and therefore Python skills are in high demand.

Drawbacks: Python is not suitable for mobile application development.

Common uses: Python is used in a wide variety of applications, including artificial intelligence, financial services, and data science. Social media sites such as Instagram and Pinterest are also built on Python.

TOP 10
Popular Programming Languages in 2020

1. Python
2. JavaScript
3. Java
4. C#
5. C
6. C++
7. GO
8. R
9. Swift
10. PHP

WWW.NORTHEASTERN.EDU/GRADUATE

The Python programming language is open-source programming. The URL https://www.python.org/ is the official website for the Python programming language.

1.7.1 Python Libraries for Statistics

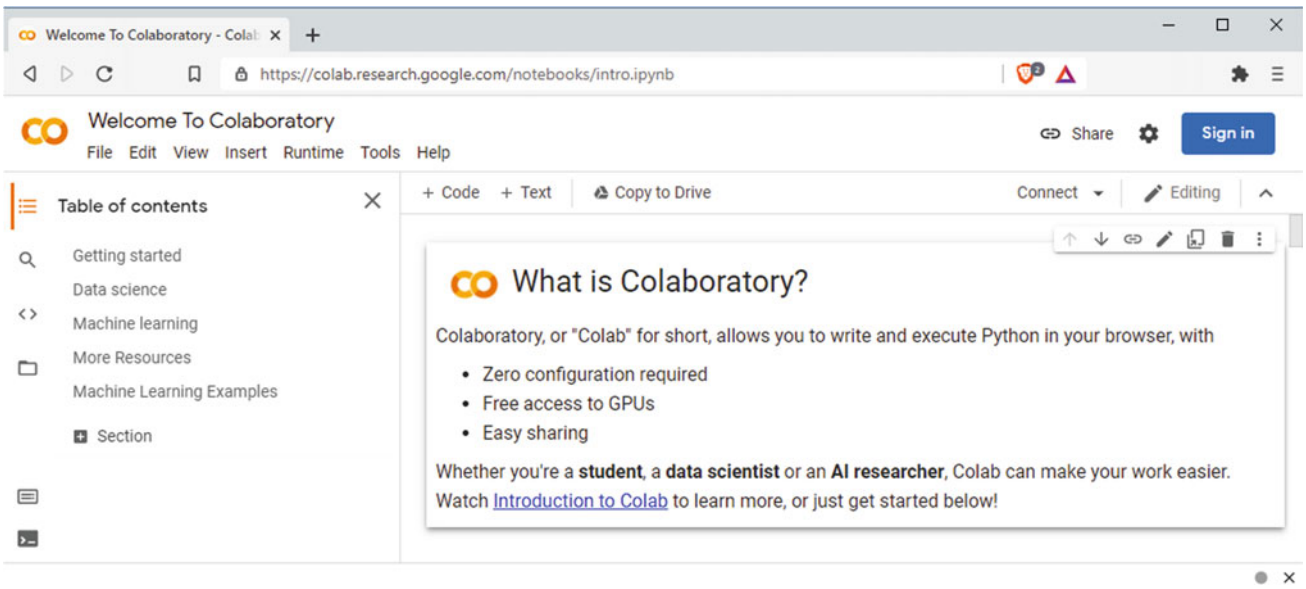
Python is not a statistical program; instead, it's a general-purpose programming language. What makes Python so powerful and popular is that there are a lot of libraries *for free* available for Python. The following are some popular Python libraries to do statistical analysis.

(1) Numpy. The official URL of Numpy is https://numpy.org/.
(2) Pandas. The official URL of Pandas is https://pandas.pydata.org/.
(3) SciPy. The official URL of SciPy is https://www.scipy.org/.
(4) Statsmodels. The official URL of Statsmodel is https://www.statsmodels.org/stable/index.html.

1.7.2 Python Development Environment

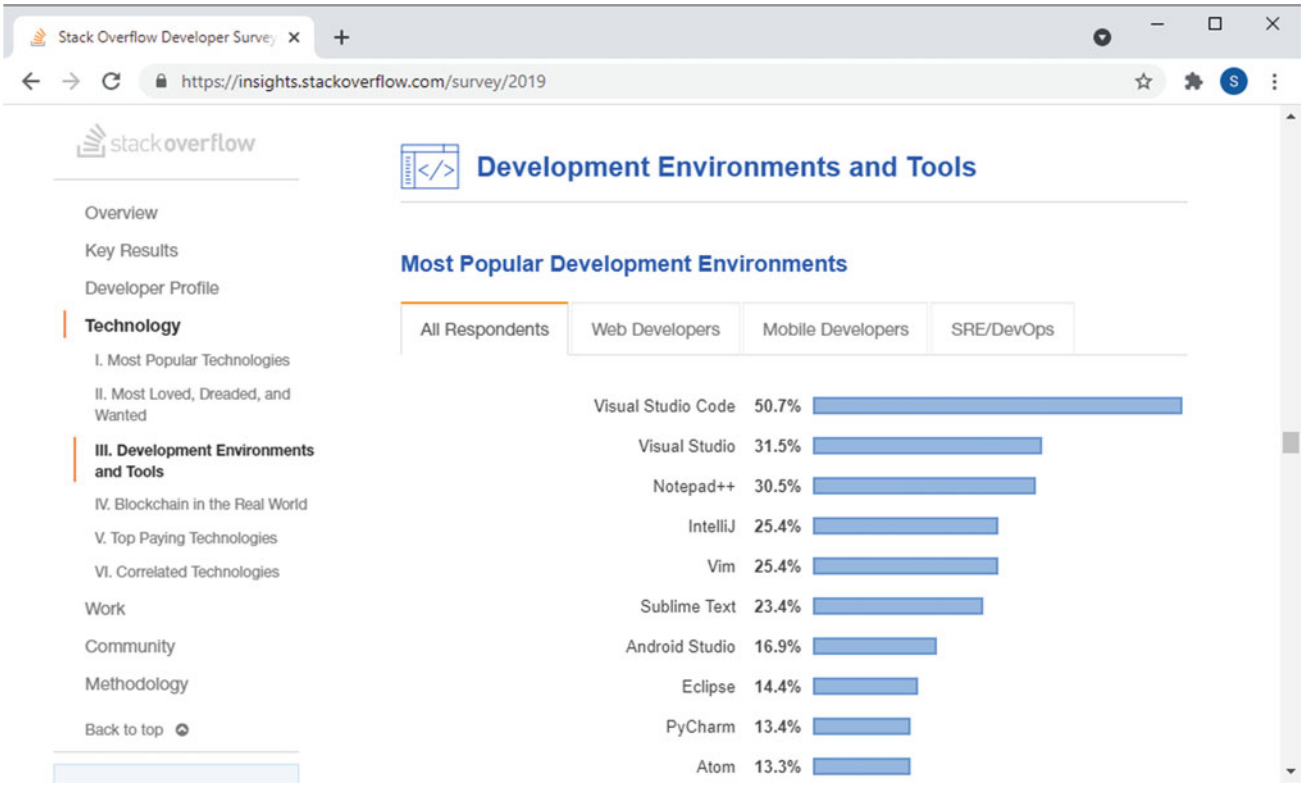
1.7.2.1 Google Colaboratory

There are many different popular development environments for Python. Most of the time, this book will use *Google Colaboratory* as the environment to run Python code. The official URL for Google Colaboratory is https://colab.research.google.com/notebooks/intro.ipynb.



1.7.2.2 Visual Studio Code

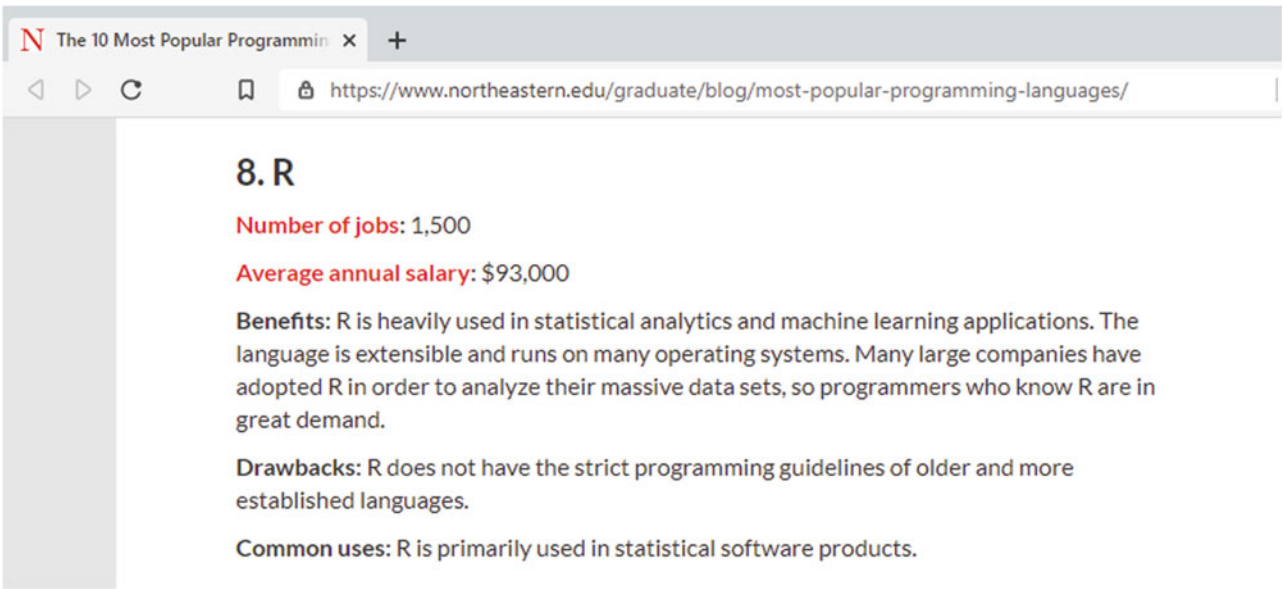
Google Colaboratory is popular with data scientists who want to explore data sets. But Google Colaboratory is not suitable for serious Python development. The most popular development environment is *Visual Studio Code*. The URL https://insights.stackoverflow.com/survey/2019 shows the top 2019 development environment.



The official website for *Visual Studio Code* is https://code.visualstudio.com.

1.8 *R* Programming Language

R is an open-source programming language dedicated to statistical analysis. According to the URL https://www.northeastern.edu/graduate/blog/most-popular-programming-languages/, R is the eighth most popular programming language.



The URL https://www.r-project.org/ is the official website for R. The most popular development environment for *R* is R Studio. The official website for R Studio is https://www.rstudio.com/.

1.9 Web Scraping for Market and Financial Data

One of the toughest parts of doing statistical analysis is to obtain the data to do the statistical analysis. Recently, it has become much easier to get data to do the statistical analysis. It is now easier to get data because of the internet. Many websites have the data of interest to users.

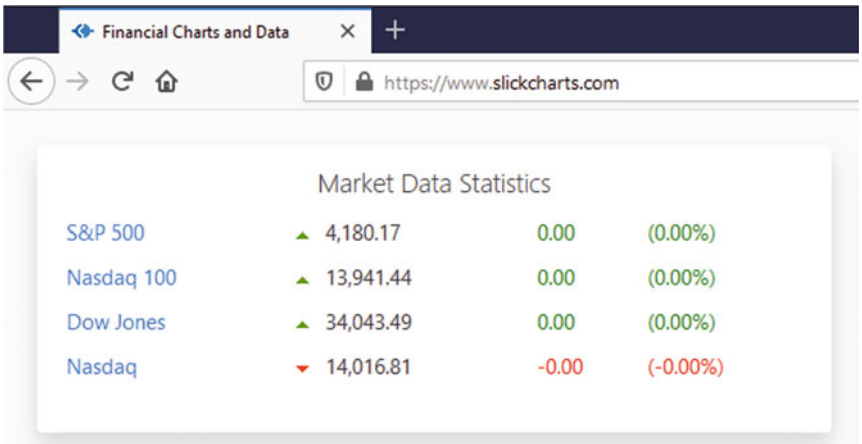
There are two general ways to get data sets from the internet. The first way to get the data from the internet is to web scrap the data from the internet by using a computer program or computer language to read a website to retrieve the data.

The second way to get the data from the internet is to subscribe to data providers. Most data providers sell specific data for a fee. Users pay a fee to the data provider to retrieve their data. A user would use a computer program or computer language to request specific data from the data provider. Most data providers indicate the frequency, amount allowed to be requested, and time period allowed for a specific price. Many data providers allow users to sample the type of data they have by limiting the frequency, amount of data, and time period a user can request. Governmental data providers usually have free data sets.

1.9.1 Microsoft Excel Power Query

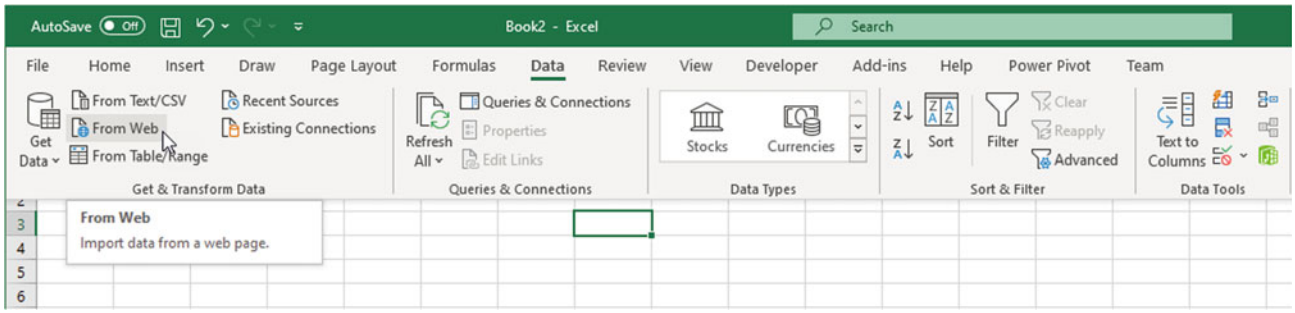
Excel's *Power Query* is a feature that can web scrape many types of data from the internet.

On April 25, 2021, URL https://www.slickcharts.com/ showed the balances of the following market indexes.



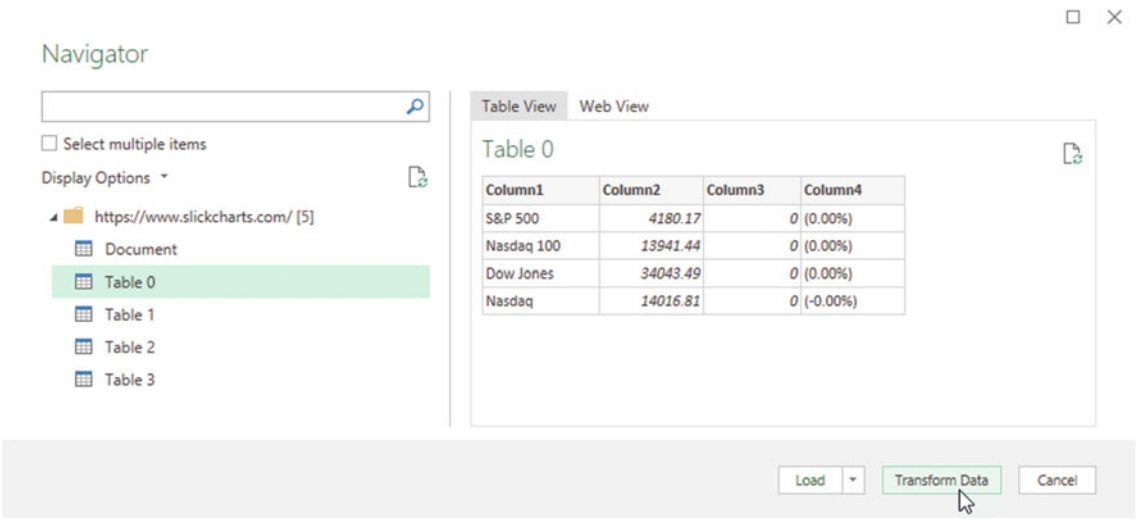
We can use *Power Query* to retrieve the balances of these market indexes. The following are the steps to do this in *Power Query*.

First, go to the *Data* tab and click on the *From Web* button.

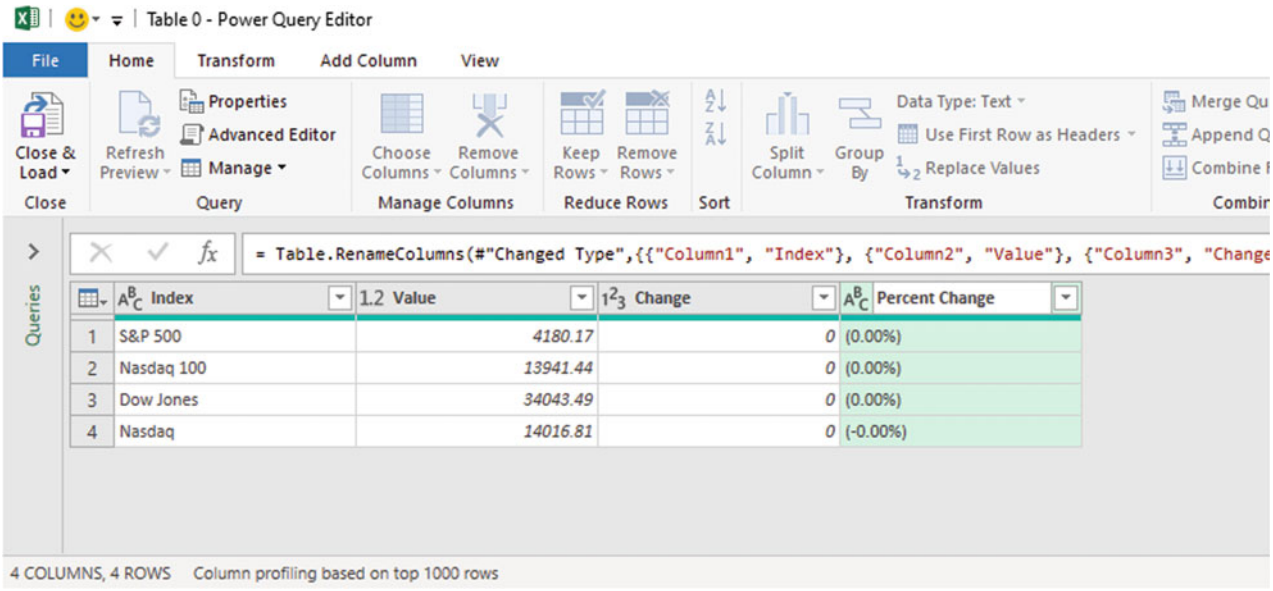


Next, enter the URL https://www.slickcharts.com/.

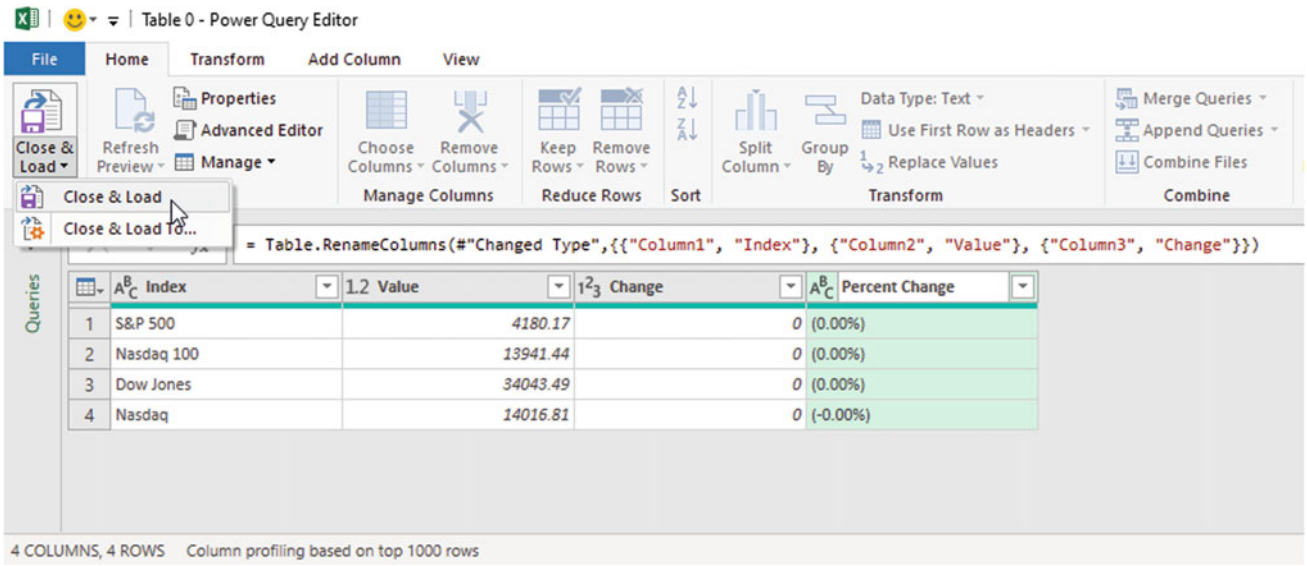
Next, select *Table 0* and then click the *Transform Data*.



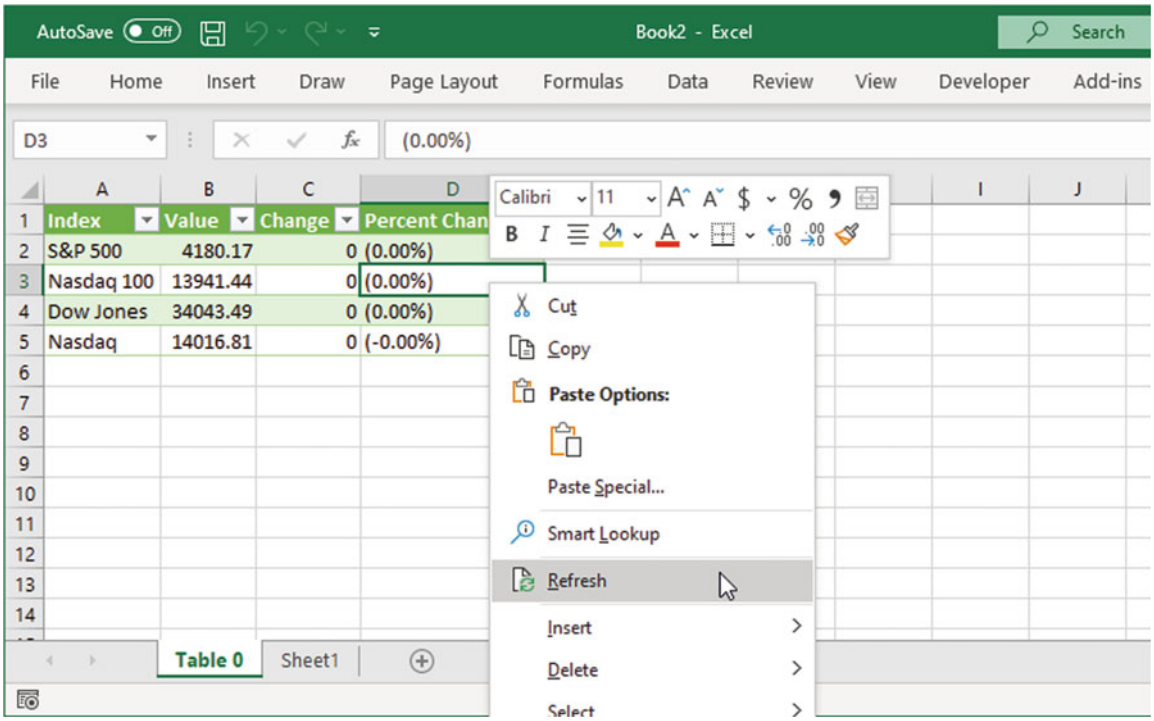
Next, change the column headers to *Index, Value, Change,* and *Percent Change*.



Next, click on *Close & Load*.



The data is saved in Excel.



You can refresh the data by selecting a cell in the table and right-mouse click to refresh the data. Refreshing is a *power query* feature. Before this refreshing feature, a lot of VBA code was required to do this. Instead of doing a lot of VBA programming, a user only need to set up this refreshing feature only once.

Note that the data source of the data is the URL https://www.slickcharts.com/.

1.10 Case Study, Google Study, and Active Study Approach

Microsoft Excel is the overwhelming default go-to program of the business user. Business users try to use Excel to solve all their business needs. Most business users only use a fraction of the features available in Excel and inefficiently use Excel. The main problem is that Excel is a very large, sophisticated program, which is shown by the many large "encyclopedias" like Excel books available. Most people do not read an encyclopedia from cover to cover. For the same reason, most people do not read these "encyclopedia" Excel books from cover to cover.

The best way to learn Excel is by applying Excel to problems. "Learning by doing" is the best way to learn Excel. Therefore, this book will "kill two birds with one stone" This book will discuss the concepts of statistics and simultaneously discuss how to use Excel to implement the concepts of statistics. This approach of learning Excel is the case study approach.

In this book, the case study approach will be used to demonstrate what can be achieved in Excel. It is well known that the best way to learn a topic is to actively study a topic instead of passively study a topic. To actively study a topic one should always ask questions about what is being presented and then find the answer on the internet, i.e., *Google search* the topic. Experienced computer programmers often solve problems by searching for the answers on *Google*.

1.11 Structure of the Book

There are seven parts in this book. Volume I includes financial statistics and portfolio analysis. There are 21 chapters in the financial statistics part, which show how Excel VBA, Python, and R can be used for financial statistical analysis. The portfolio analysis portion, which includes five chapters, shows how Excel VBA can be used to perform portfolio analysis.

There are 23 chapters, which are divided into 5 parts, in Volume II, entitled *Microsoft Excel VBA, Python and R For Financial Derivatives, Financial Analysis, and Machine Learning*. They are Excel VBA; financial derivatives, applications of Python, machine learning for financial derivatives, and risk management; financial management; and applications of R programs for financial analysis.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Part I
Financial Statistics

Data Collection, Presentation, and Yahoo! Finance

2

2.1 Introduction

To learn statistics, the first step we need to do is collect data, then we need to present the data in either tables or graphs. The source of data collection is very important. In this chapter, we will show how data can be collected, then we will show how data can be presented in either a table or graph. In addition, we show how we can use Python to get the data for Johnson & Johnson from Yahoo! Finance. We also present this data in a graph. In this section, we show how to get the data from the Dow Jones Industrial Average, S&P 500, and NASDAQ from Yahoo! Finance and perform some analysis.

In Sect. 2.1 of this chapter, we discuss data presentation and we then discuss Yahoo! Finance in Sect. 2.2. Market indexes such as the Dow Jones Industrial Average, S&P 500, and NASDAQ are discussed in Sect. 2.3. JSON data format is examined in Sect. 2.4. In Sect. 2.5, we look at ticker attributes, and in Sect. 2.6, we discuss historical data. Section 2.7 discusses charting this historical data. In Sect. 2.8, we use Python to graph Johnson & Johnson's historical prices. Finally, in Sect. 2.9, we summarize the chapter.

2.2 Data Presentation

Statistics is mainly associated with numbers. A complete statistical analysis of a data set consists of both numbers and graphs. Graphs many times allow users to understand a data set that is very hard to capture with a number.

Prior to Microsoft Excel, it has been very hard to create professional-looking graphs. There is a feature in Excel called a *chart wizard* that has made creating graphs easy. We will look at the chart wizard in detail in this chapter.

Until very recently, it has been really hard to get market, financial and economic data to use for data analysis. Now with the advent of the Internet, it has been much easier to get market, financial and economic data for data analysis. The go-to place to get market and financial data is Yahoo! Finance. Yahoo! Finance's Uniform Resource Locator or URL is https://finance.yahoo.com/. The URL https://www.verisign.com/en_US/website-presence/online/what-is-a-url/index.xhtml discusses more the definition of a URL. In this chapter, we will look at how to retrieve market and financial data from Yahoo! Finance.

2.3 Yahoo! Finance

In this chapter, we will look at financial data that spans through time. The most popular technique to analyze financial data that spans through time is to use line graphs.

In this chapter, we will analyze the stock price of Apple Inc. from January 1, 2000, to January 1, 2021.

One of the most popular places to get financial data is Yahoo! Finance. To get Apple Inc.'s historical stock price, you would go to https://finance.yahoo.com/quote/AAPL/history. It is interesting to note that to get the historical stock price of Johnson & Johnson, you would go to https://finance.yahoo.com/quote/JNJ/history. To get the historical price of Microsoft, you would go to https://finance.yahoo.com/quote/MSFT/history. Note the pattern in constructing the URL to get historical prices for a stock.

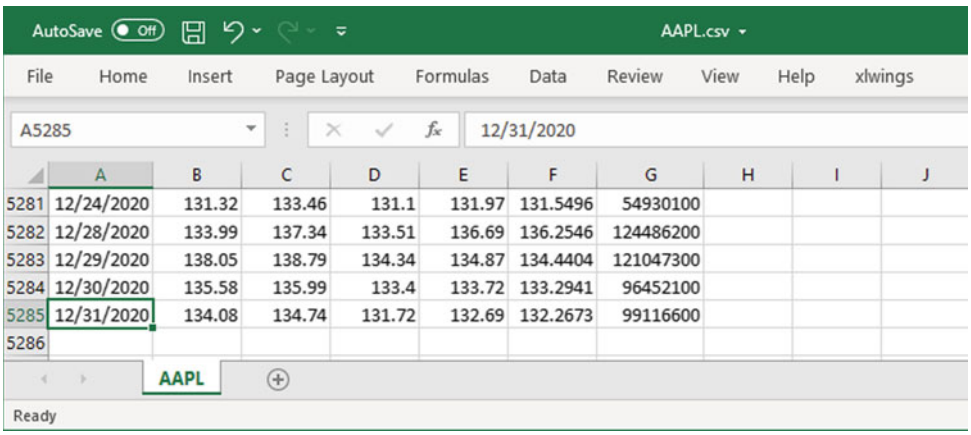
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_2

The following are the steps to get AAPL's historical stock prices from January 1, 2000, to January 1, 2021:

1. Using a web browser like Microsoft Edge, Firefox, or Google Chrome, go to the location https://finance.yahoo.com/quote/AAPL/history. Yahoo! Finance will show you the available date ranges. The earliest available historical data available for AAPL is December 11, 1980. For this example, we will use the Microsoft Edge web browser.
2. At the top of the web page, set the date range of interest. Let's set it from January 1, 2000, to January 1, 2001.
3. Click on the *Apply* button.
4. Below the *Apply* button, click on the *Download* link.
5. The historical file is downloaded to the top of Microsoft Edge. The name of the historical file is *APPL.CSV*. The Microsoft Excel icon indicates that Microsoft Excel will open by default open the *APPL.CSV* file.

If you go to the bottom of the worksheet, you will notice that there are 5,285 historical prices for AAPL from January 1, 2000, to January 1, 2021.



	A	B	C	D	E	F	G	H	I	J
5281	12/24/2020	131.32	133.46	131.1	131.97	131.5496	54930100			
5282	12/28/2020	133.99	137.34	133.51	136.69	136.2546	124486200			
5283	12/29/2020	138.05	138.79	134.34	134.87	134.4404	121047300			
5284	12/30/2020	135.58	135.99	133.4	133.72	133.2941	96452100			
5285	12/31/2020	134.08	134.74	131.72	132.69	132.2673	99116600			
5286										

Besides getting historical stock prices from Yahoo! Finance, you can get historical prices of popular market indexes like the Dow Jones Industrial Average, the S&P500, and the NASDAQ.

2.4 Market Indexes

2.4.1 Dow Jones Industrial Average

The weblink for the Dow Jones industrial average historical prices is https://finance.yahoo.com/quote/^DJI/history.

2.4.2 S&P 500

The weblink for the S&P 500 historical prices is https://finance.yahoo.com/quote/^GSPC/history.

2.4.3 NASDAQ

The weblink for NASDAQ's historical prices is https://finance.yahoo.com/quote/^IXIC/history.

2.5 JSON Data Format

The preferred way for computers to communicate with each other is through an *Application Program Interface* call or API call. The URL https://blogs.mulesoft.com/biz/tech-ramblings-biz/what-are-apis-how-do-apis-work discusses more on what an API is. Often when one computer requests data from another computer, the requesting computer sends the computer that has the data an URL. There is often logic in how to create the requesting URL. This requesting URL is an API.

There needs to be an agreement between the two computers on the format of the data being requested and returned. For the longest time, the preferred format of the returned data was the XML data format. Now the preferred data format to send data between computers is the JSON data format. This preference is discussed in the URL, https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html The URL https://www.guru99.com/json-vs-xml-difference.html discusses the difference between JSON and XML.

The computer that is requesting the data is making an "API request." The computer that returns the data is returning an "API response" or returning a "JSON response," or returning an "XML Response."

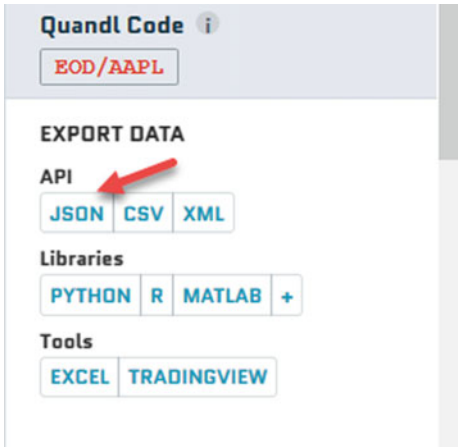
2.5.1 Quandl Data Provider

Most data providers on the Internet only return a JSON response. One of the few data providers that provide both a JSON response and an XML response for the same data is *Quandl*. Quandl is owned by NASDAQ. The URL to *Quandl* is https://www.quandl.com/. Quandl has both free data and premium data.

The URL to Quandl's free Apple Inc. historical data is https://www.quandl.com/data/EOD/AAPL-Apple-Inc-AAPL-Stock-Prices-Dividends-and-Splits.

Because it's only free data, Quandl only shows AAPL's historical price from September 3, 2013, to December 28, 2017.

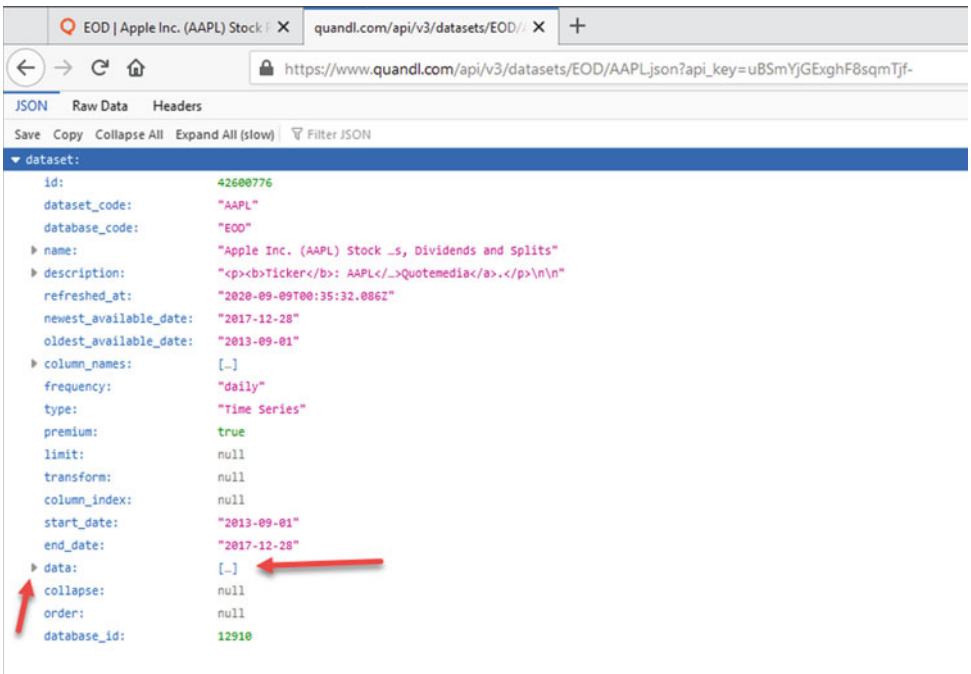
2.5.1.1 JSON Data Format



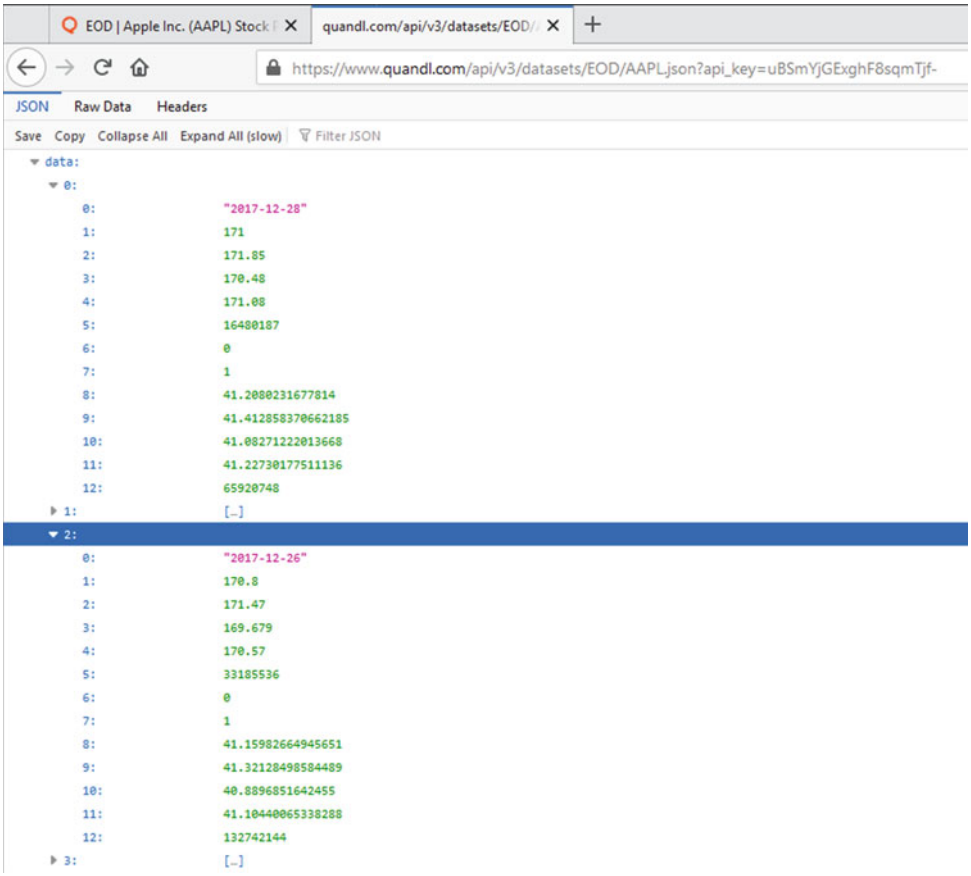
Clicking on the *JSON* label results in Quandl showing the JSON API request to the JSON AAPL data.

Pretty JSON Data Format

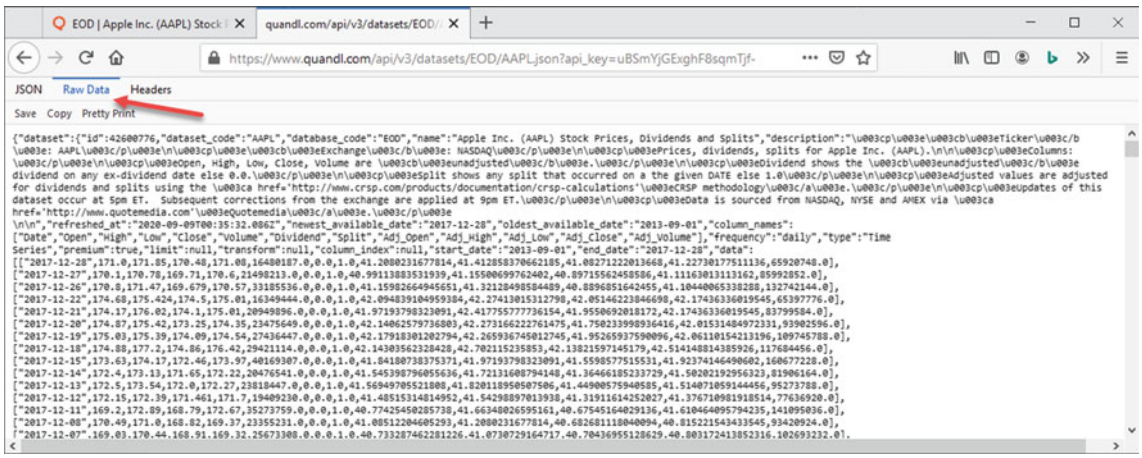
The preferred web browser to review JSON data is Firefox because it formats the JSON data to a *pretty format* so humans can more easily read and understand the data.



The historical AAPL prices are part of the *data* element. The three dots next to the *data* element indicate that the data element's child data is hidden. Click on the gray array next to the *data* element to see AAPL historical prices.

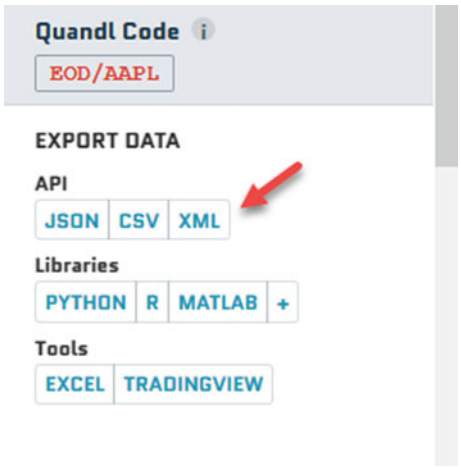


2.5.1.2 RAW JSON Data Format

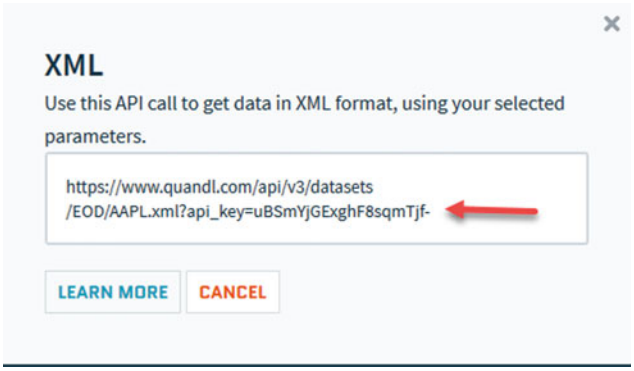


It is also possible to view AAPL's "raw" JSON data set by clicking on the *Raw Data* label in Firefox.

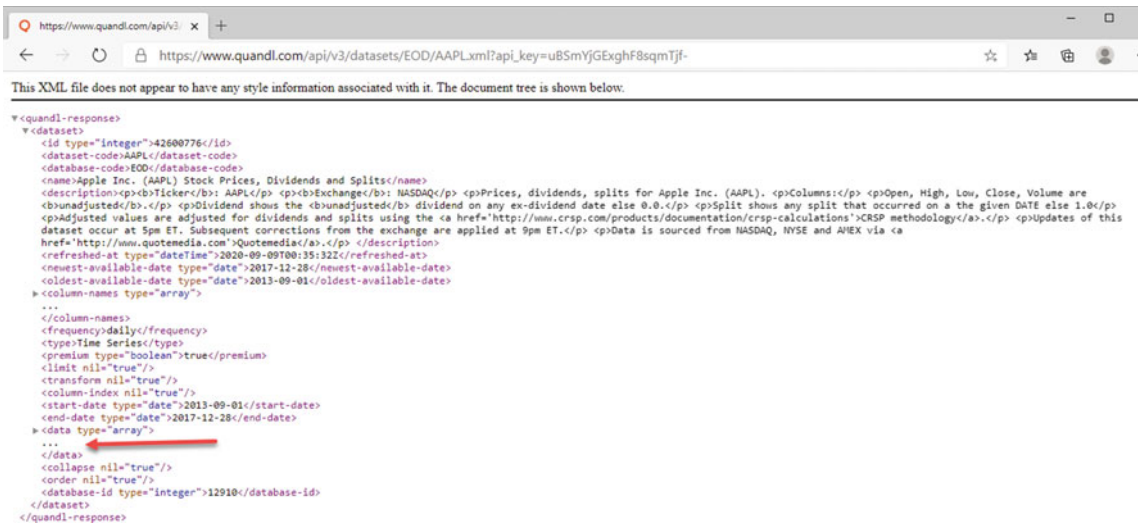
2.5.1.3 XML Data Format



Clicking on the *XML* label results in Quandl showing the XML API request to the XML AAPL data.



For XML data, the Microsoft Edge web browser does a better job in formatting XML.

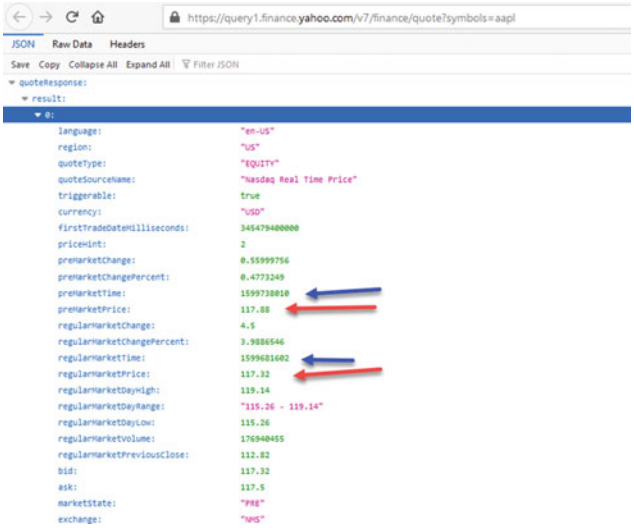


Click on the gray arrow to expand the *data* elements to see the historical prices of AAPL.

2.6 Ticker Attributes

2.6.1 Yahoo! Finance API

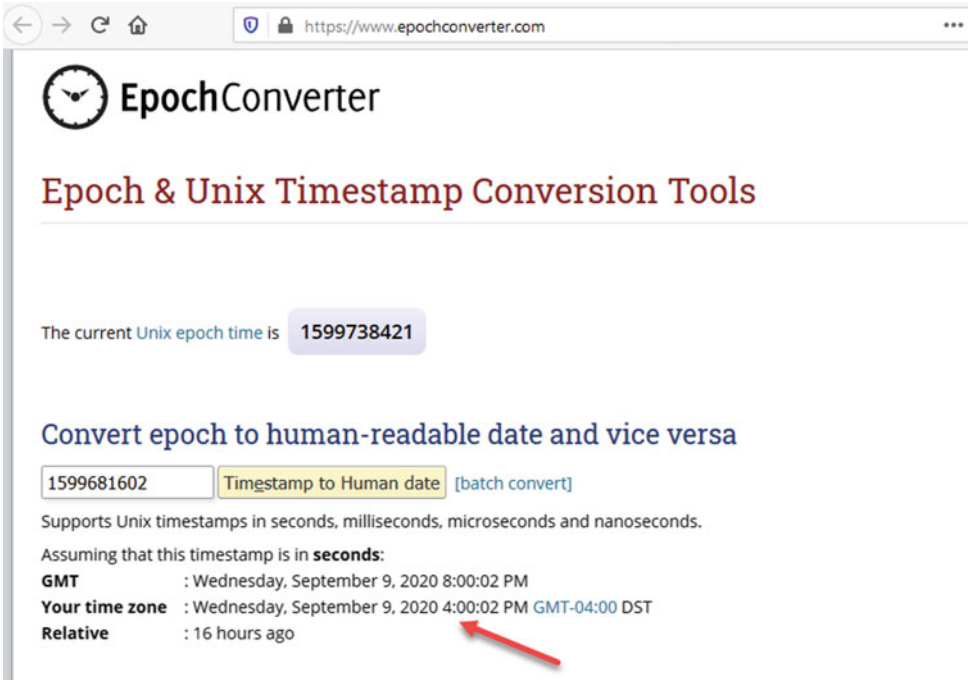
The URL https://query1.finance.yahoo.com/v7/finance/quote?symbols=aapl is the Yahoo! Finance API that shows the attributes of the AAPL ticker. On the early morning of September 10, 2020, the following shows the first 26 attributes in Firefox for the ticker AAPL. This API sometimes returns 74 attributes for stocks and sometimes 70 attributes for stocks. The attributes are displayed in random order; with every refresh, the order of the attributes is different.



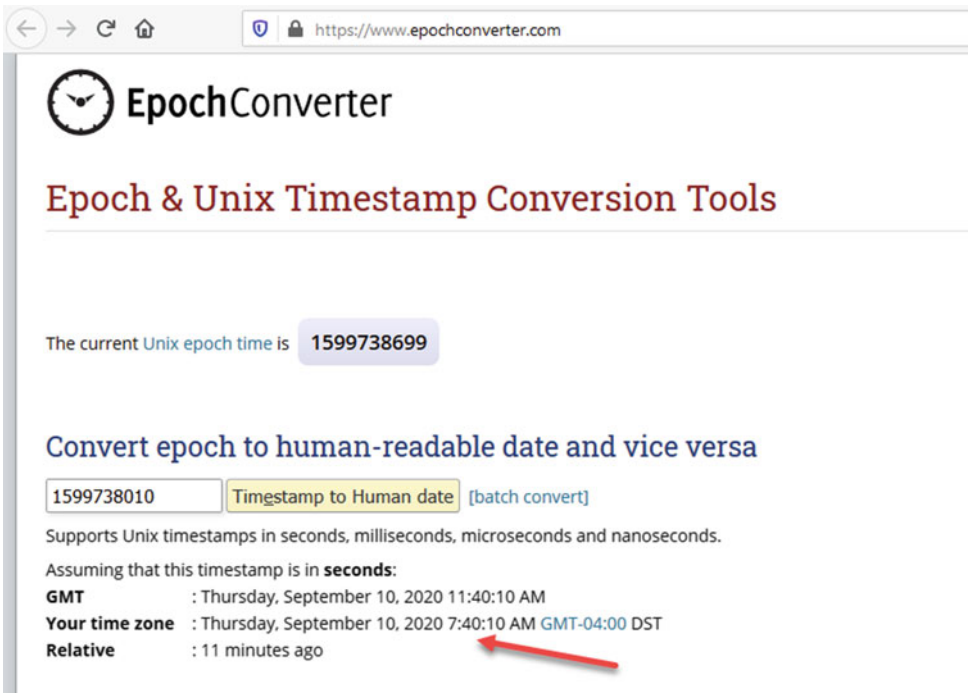
We can confirm these attributes are correct by going to Yahoo! Finance URL https://finance.yahoo.com/quote/AAPL.

The ticker attribute *regularMarketPrice* value was 117.32. This price agrees with the price shown in the URL https://finance.yahoo.com/quote/AAPL. The ticker attribute *preMarketPrice* value was 117.88. This price agrees with the price shown in the URL https://finance.yahoo.com/quote/AAPL.

The ticker attribute *regularMarketTime* value was 1599681602. This value is *epoch* time. Going to the URL https://www.epochconverter.com/, we can see the epoch time 1599681602 is September 4, 4:00 PM DST. This time agrees with the September 9, 4:00 PM EDT shown in the URL https://finance.yahoo.com/quote/AAPL.

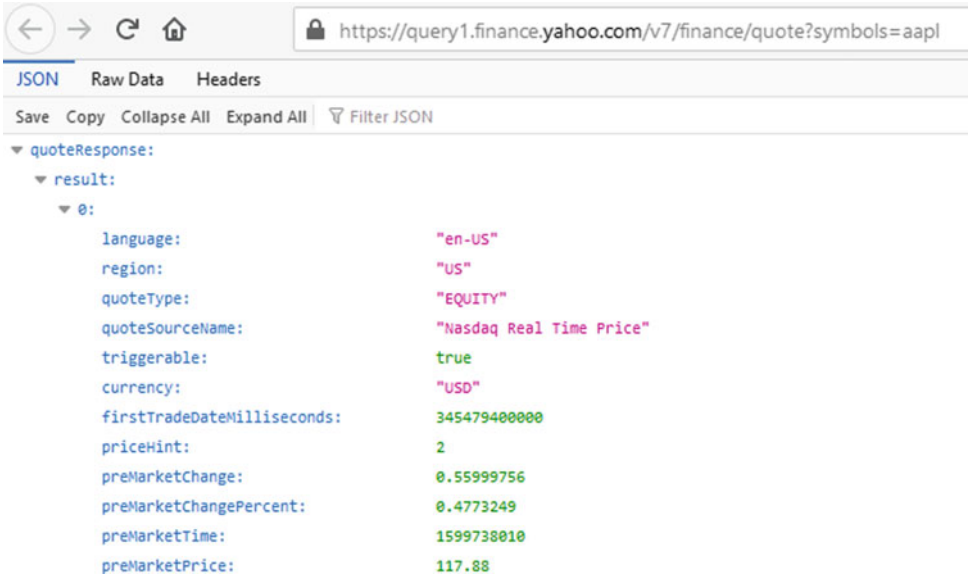


The ticker attribute *preMarketTime* value was 1599738010. Going to the URL https://www.epochconverter.com/, we can see the epoch time 1599738010 is September 10, 7:40 AM DST. This time agrees with the September 10, 7:40 AM EDT shown in the URL https://finance.yahoo.com/quote/AAPL.



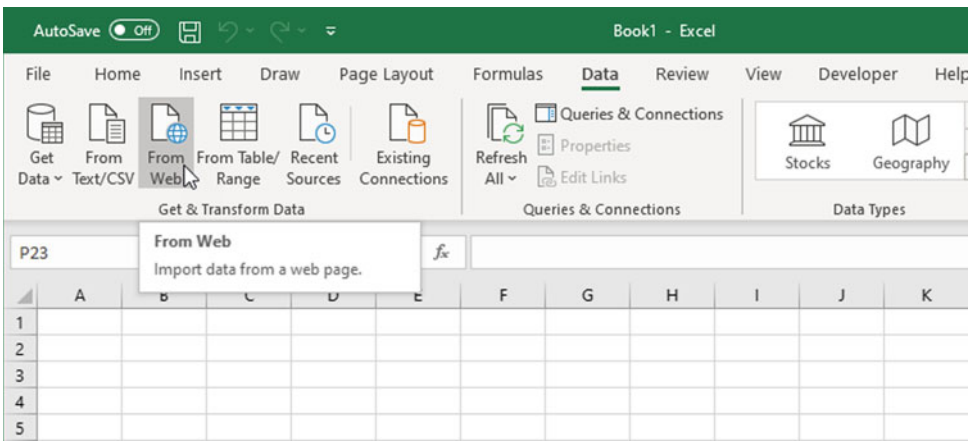
2.6.1.1 Power Query

Before configuring *Power Query,* it is best to review an API in Firefox to understand the JSON data set structure. Understanding the structure of a JSON data set helps guide us in how to configure *Power Query.* We can see that the attributes of the AAPL ticker are three levels below the *quoteReponse* element.

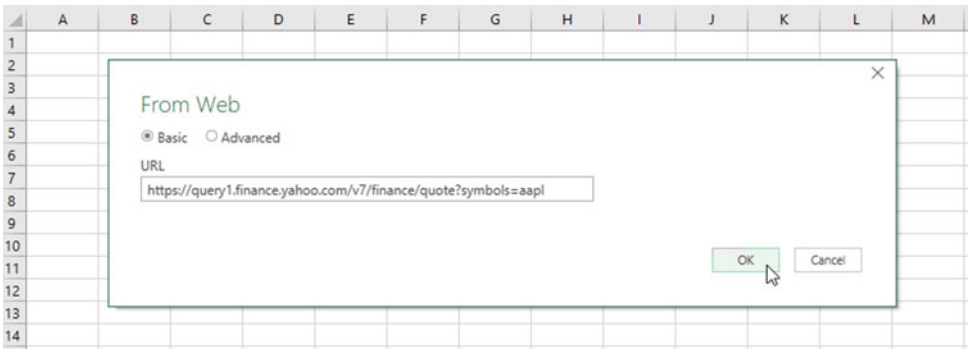


The following are the steps to retrieve all the attributes for the stock ticker AAPL. We will break up the 74 attributes into two lists by creating two queries in *Power Query.*

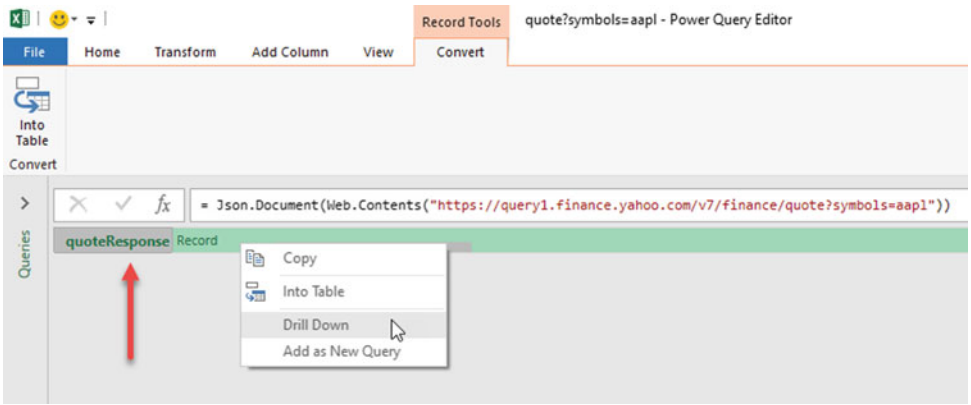
(1) Click on *From Web Page* in the *Data* tab.



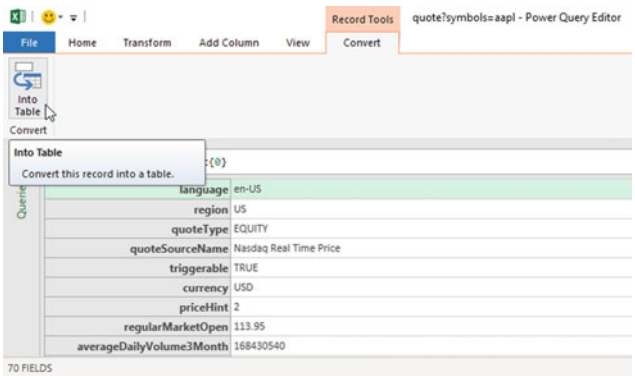
(2) Enter the URL https://query1.finance.yahoo.com/v7/finance/quote?symbols=aapl.



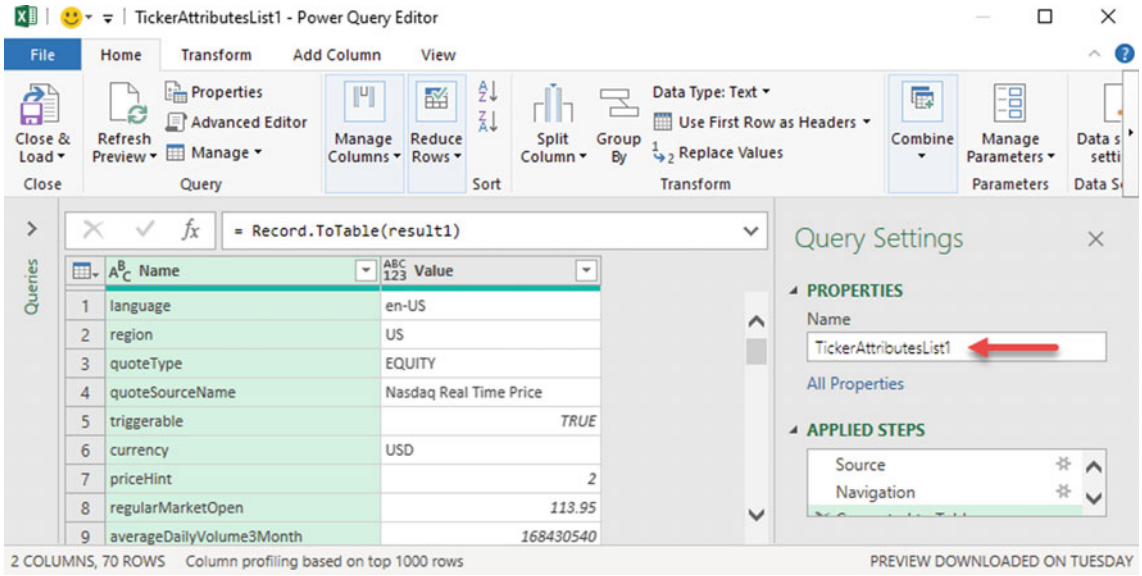
(3) Power query first shows us a *quoteResponse* row. The *quoteResponse* row matches the first element of the JSON data set shown above. This is why we should drill down on the word *Record*.



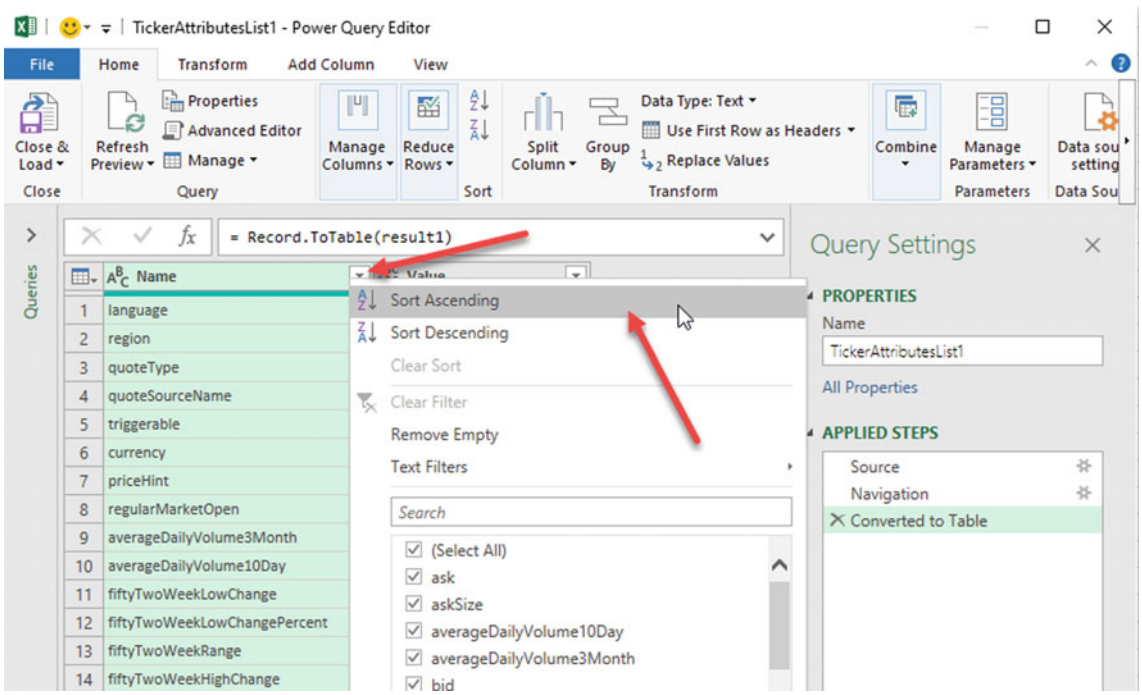
(4) We should keep drilling down until we see the attributes of the AAPL ticker. Once we see the attributes click on *Into Table*.



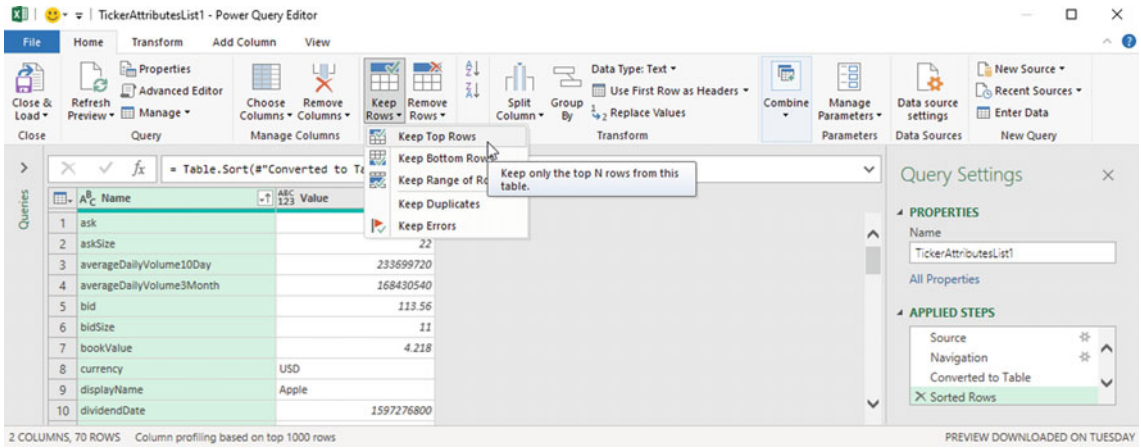
(5) Change the query name to *TickerAttributesList1*.



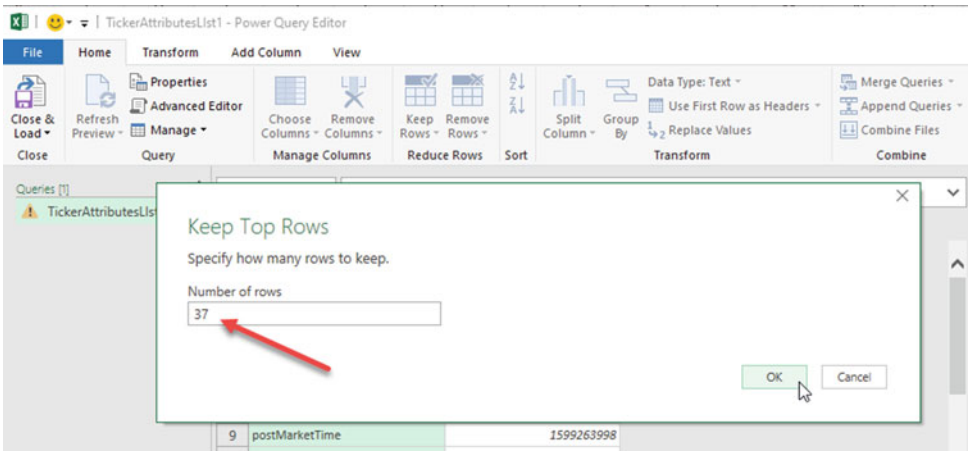
(6) Because the attributes are delivered in random order, let's sort the attributes in ascending order. Click on the down arrow in the *Name* field and then click on *Sort Ascending*.



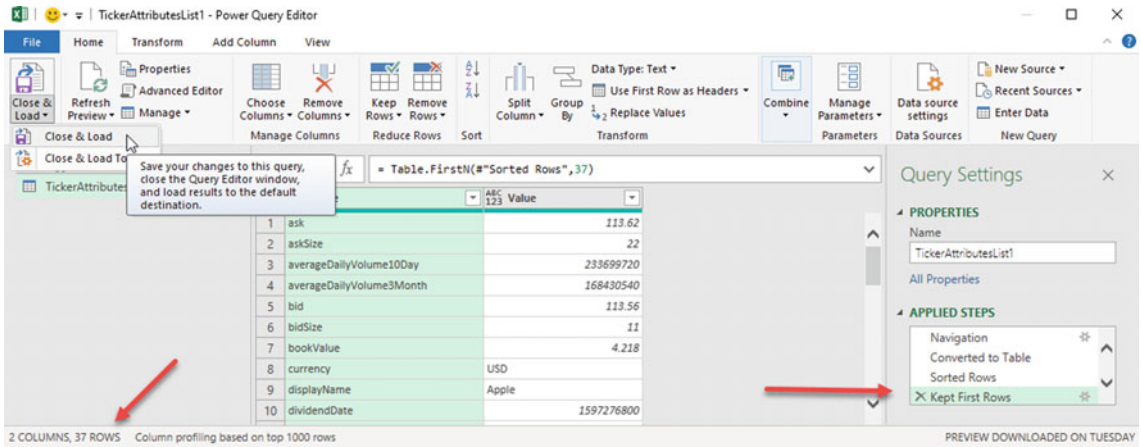
(7) The next step is to restrict the list to 37 attributes. We will do this by keeping the top 37 attributes on the list. To do this, click on *Keep Top Rows* in the *Home* tab.



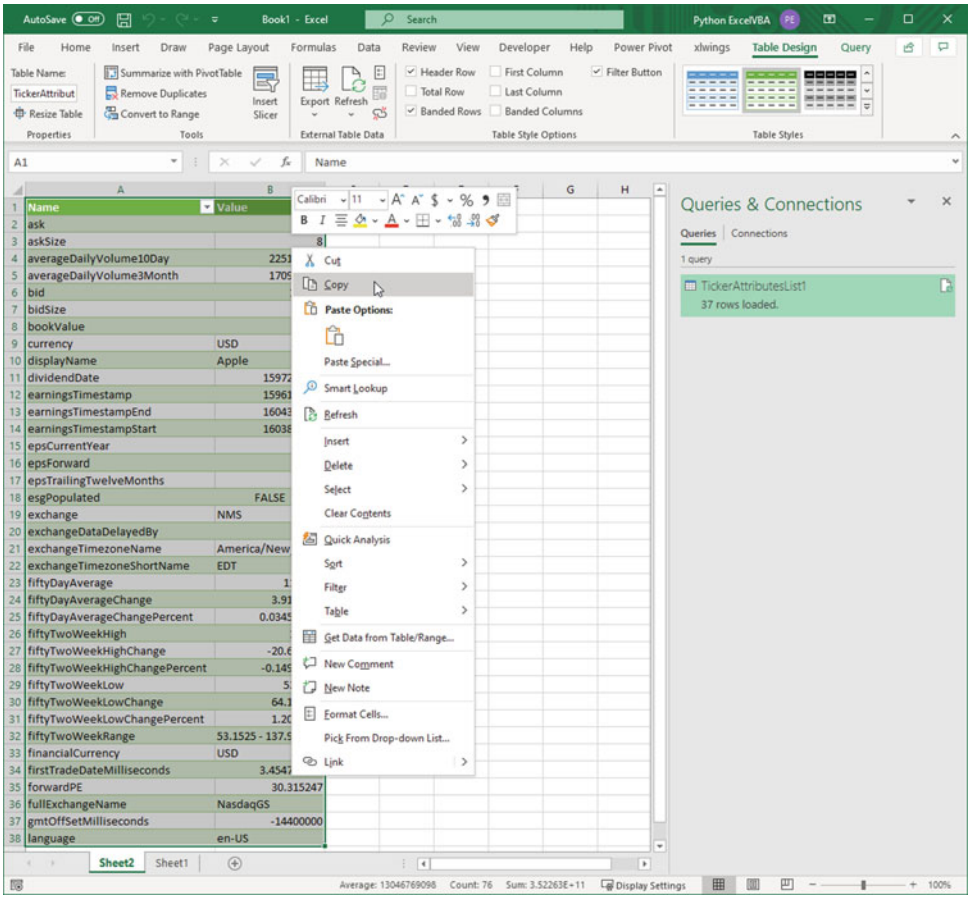
(8) Enter 37.



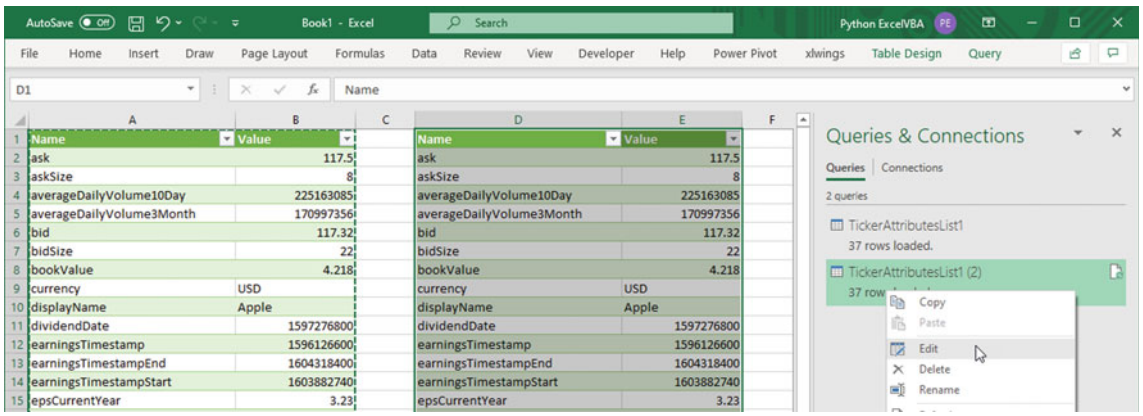
(9) Click on *Close & Load*. Notice that the status bar indicated only 37 fields, which means there are only 37 attributes. Also, notice that the *APPLIED STEPS* section is keeping track of the manual steps that we have applied to the query.



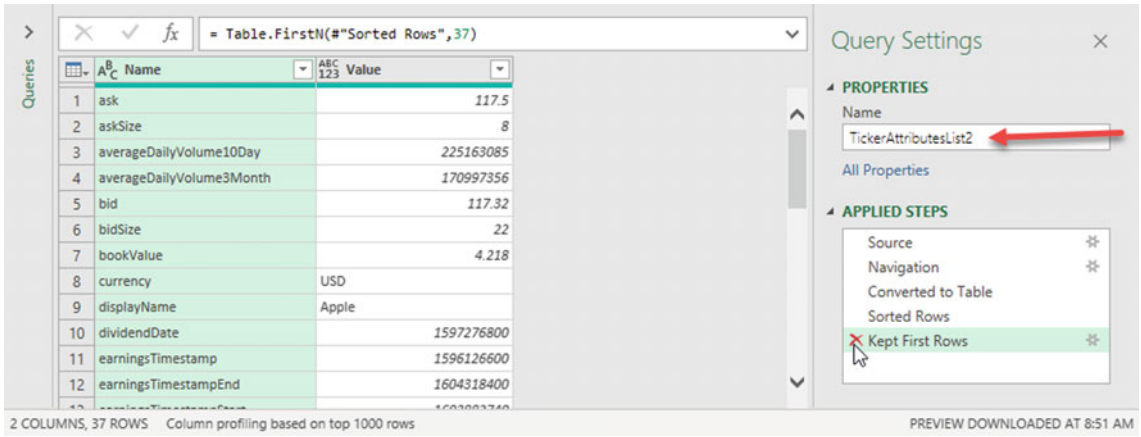
(10) Excel loaded the query to an Excel table in Sheet2. It also shows the query that the Excel table is using. Let's now select the Excel table and copy it to cell d1.



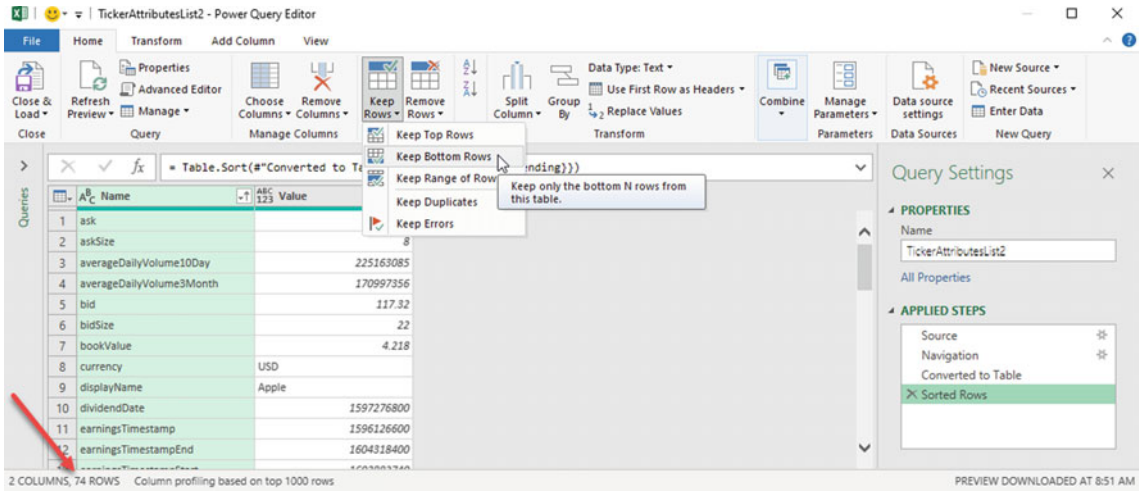
(11) The second Excel table caused the second duplicate query to be created. Let's edit the second query by right-mousing on the send query and selecting *edit*.



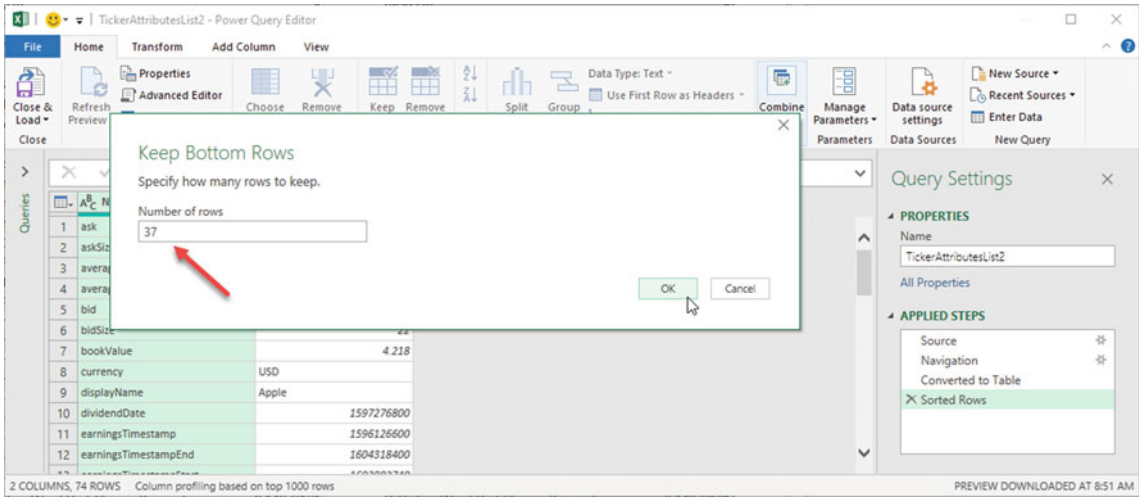
(12) Power query recorded all the steps in the *APPLIED STEPS* section. Let's remove the *Kept First Rows* step by clicking on the red X shown below. Let's also rename the query to *TickerAttributesList2*, as shown below.



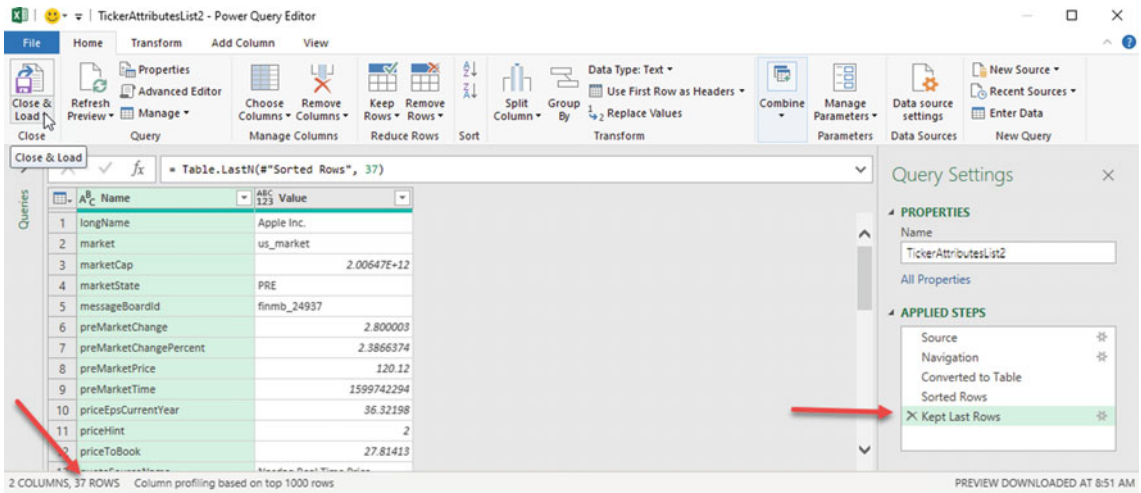
(13) This time let's keep the bottom 37 attributes. To do this, click on *Keep Bottom Rows* in the *Home* tab. Notice that the query is now returning 74 rows, while in step 7, it was only returning 70 rows.



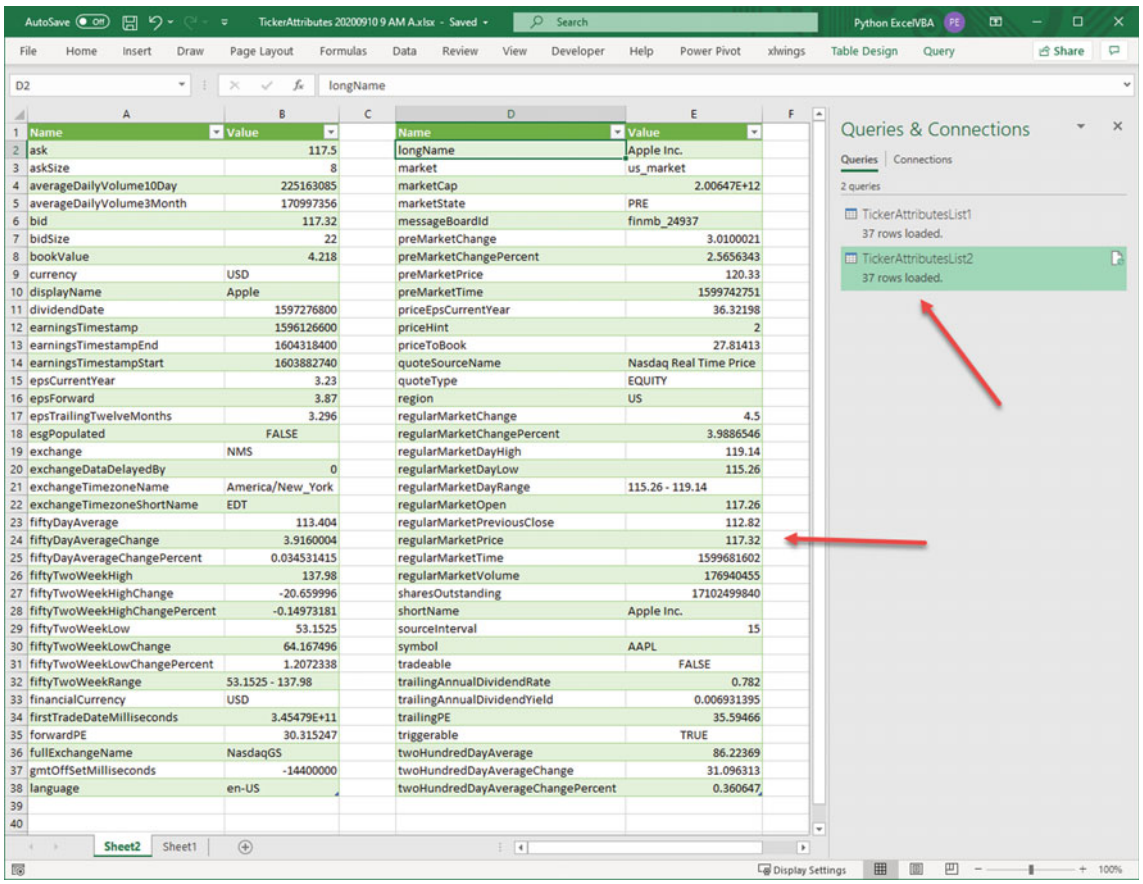
(14) Enter 37 in the *Keep Bottom Rows* window.



(15) Click on *Close & Load*. Notice the *Kept Last Rows* in the *APPLIED STEPS* area and the *37 ROWS* in the status bar.



(16) Clicking on the *Close & Load* button resulted in the second Excel table showing the last 37 attributes for the ticker AAPL. Also, we can see that the second query was renamed to *TickerAttributesList2*.



Name	Value
ask	117.5
askSize	8
averageDailyVolume10Day	225163085
averageDailyVolume3Month	170997356
bid	117.32
bidSize	22
bookValue	4.218
currency	USD
displayName	Apple
dividendDate	1597276800
earningsTimestamp	1596126600
earningsTimestampEnd	1604318400
earningsTimestampStart	1603882740
epsCurrentYear	3.23
epsForward	3.87
epsTrailingTwelveMonths	3.296
esgPopulated	FALSE
exchange	NMS
exchangeDataDelayedBy	0
exchangeTimezoneName	America/New_York
exchangeTimezoneShortName	EDT
fiftyDayAverage	113.404
fiftyDayAverageChange	3.9160004
fiftyDayAverageChangePercent	0.034531415
fiftyTwoWeekHigh	137.98
fiftyTwoWeekHighChange	-20.659996
fiftyTwoWeekHighChangePercent	-0.14973181
fiftyTwoWeekLow	53.1525
fiftyTwoWeekLowChange	64.167496
fiftyTwoWeekLowChangePercent	1.2072338
fiftyTwoWeekRange	53.1525 - 137.98
financialCurrency	USD
firstTradeDateMilliseconds	3.45479E+11
forwardPE	30.315247
fullExchangeName	NasdaqGS
gmtOffSetMilliseconds	-14400000
language	en-US

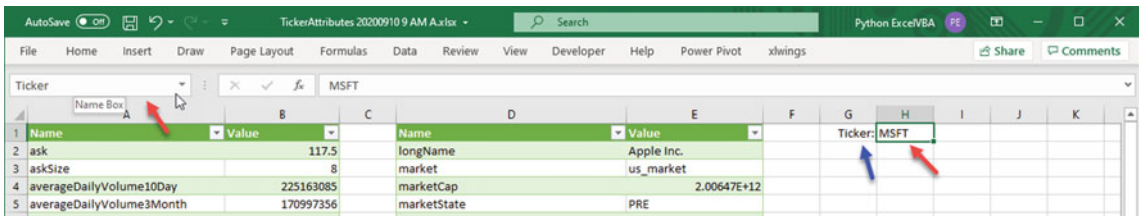
Name	Value
longName	Apple Inc.
market	us_market
marketCap	2.00647E+12
marketState	PRE
messageBoardId	finmb_24937
preMarketChange	3.0100021
preMarketChangePercent	2.5656343
preMarketPrice	120.33
preMarketTime	1599742751
priceEpsCurrentYear	36.32198
priceHint	2
priceToBook	27.81413
quoteSourceName	Nasdaq Real Time Price
quoteType	EQUITY
region	US
regularMarketChange	4.5
regularMarketChangePercent	3.9886546
regularMarketDayHigh	119.14
regularMarketDayLow	115.26
regularMarketDayRange	115.26 - 119.14
regularMarketOpen	117.26
regularMarketPreviousClose	112.82
regularMarketPrice	117.32
regularMarketTime	1599681602
regularMarketVolume	176940455
sharesOutstanding	17102499840
shortName	Apple Inc.
sourceInterval	15
symbol	AAPL
tradeable	FALSE
trailingAnnualDividendRate	0.782
trailingAnnualDividendYield	0.006931395
trailingPE	35.59466
triggerable	TRUE
twoHundredDayAverage	86.22369
twoHundredDayAverageChange	31.096313
twoHundredDayAverageChangePercent	0.360647

2.6.1.2 Power Query M Code

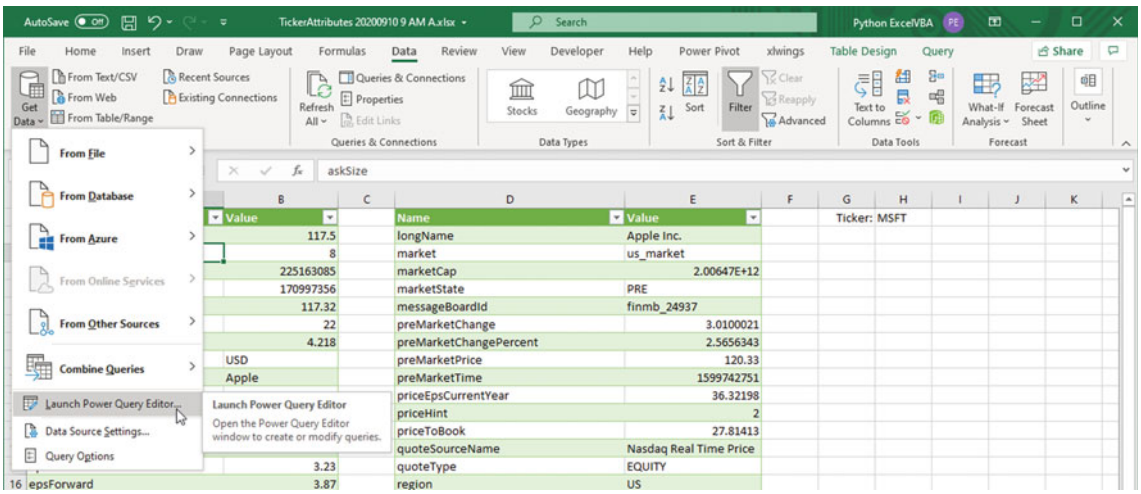
In the previous section, we perform a lot of manual steps to set up *Power Query* to retrieve the ticker attributes of AAPL. To prevent having to do this for every ticker, we will, in this section, show how to modify AAPL's *Power Query* so we can easily retrieve the ticker attributes of any stock. In essence, AAPL's *Power Query* setup will be a template for stock tickers.

We will set it up so we can enter a ticker in cell h1 and then have *Power Query* retrieve the ticker attributes based on the ticker entered. All the manual steps are recorded in *Power Query's* scripting language called *M Code*. We will modify the *M Code* to use the ticker entered in cell h1 to retrieve that ticker's attributes. To do this, *M Code* will reference the range name of the cell.

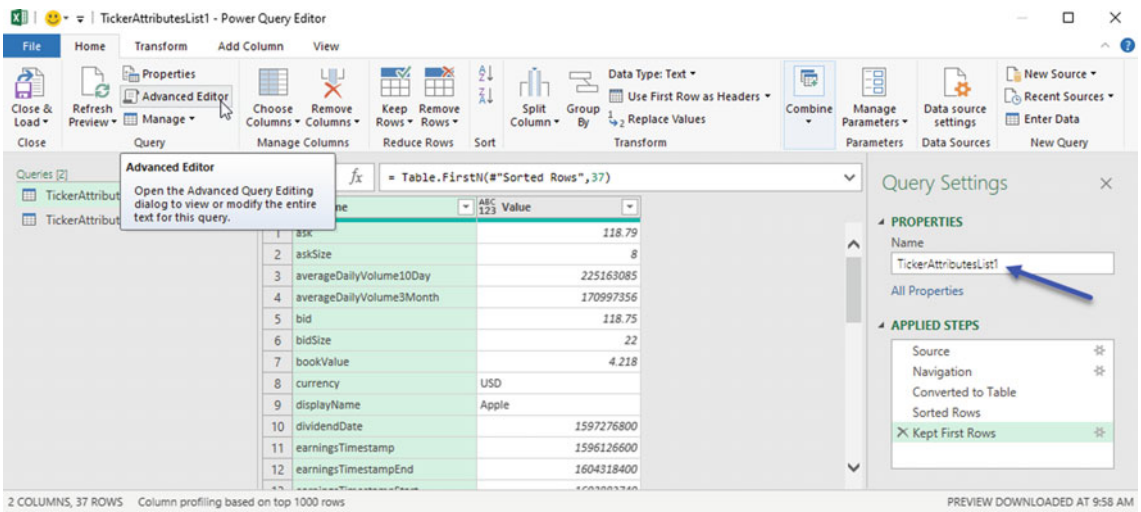
(1) Let's first name cell h1 as *Ticker*. To do this, select cell h1, and then in the *Name Box* enter "Ticker."
   a. Enter the value "Ticker:" in cell g1.
   b. Enter the value "MSFT" in cell h1.



(2) Select the first Excel data table and then do Data → Get Data → Launch Power Query Editor.



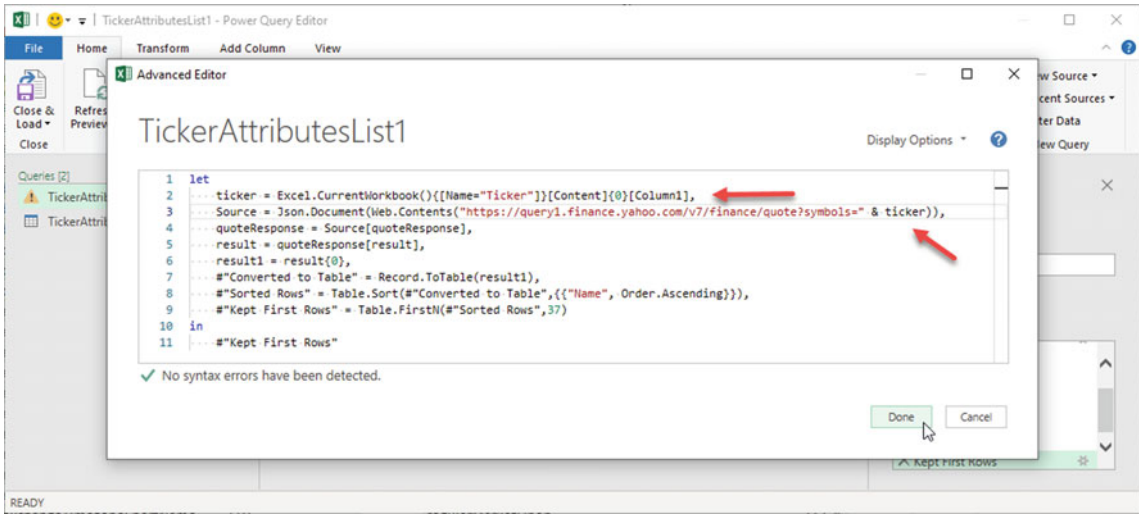
(3) Select the query *TickerAttributeList1* and then click on *Advanced Editor*.



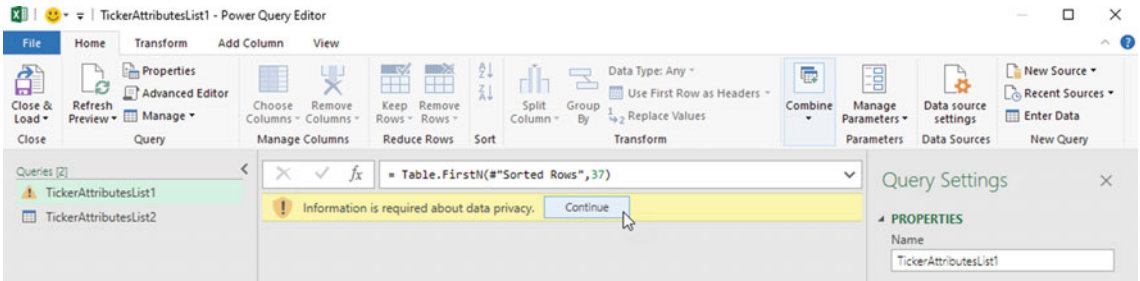
(4) Do the following:

a. Add line 2. Line 2 will retrieve the ticker of interest range the worksheet range name *Ticker* and store it in the *M Code* variable "ticker."

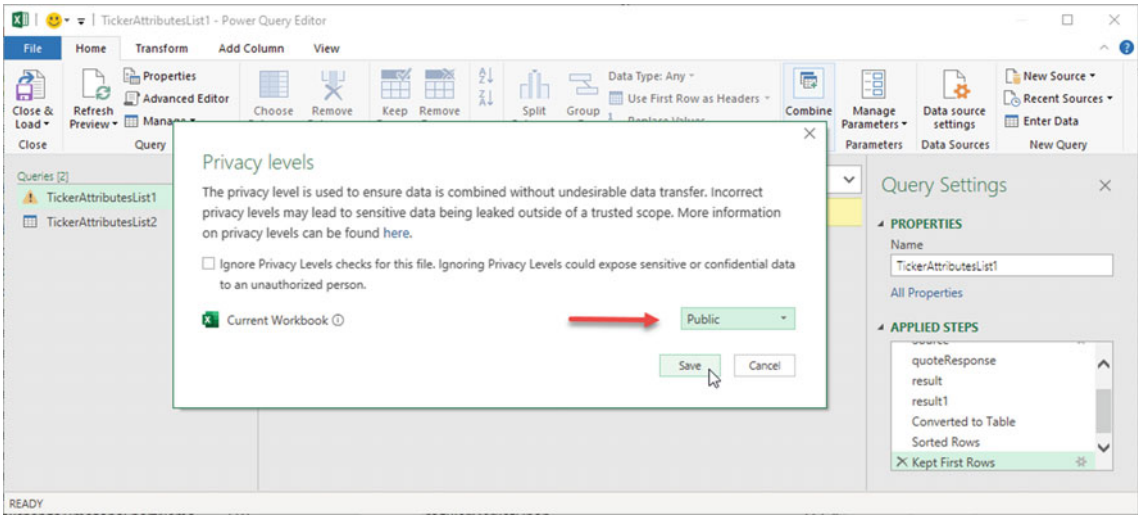
b. Update line 3. Appending the variable "ticker" will make the Yahoo! Finance API dynamic.



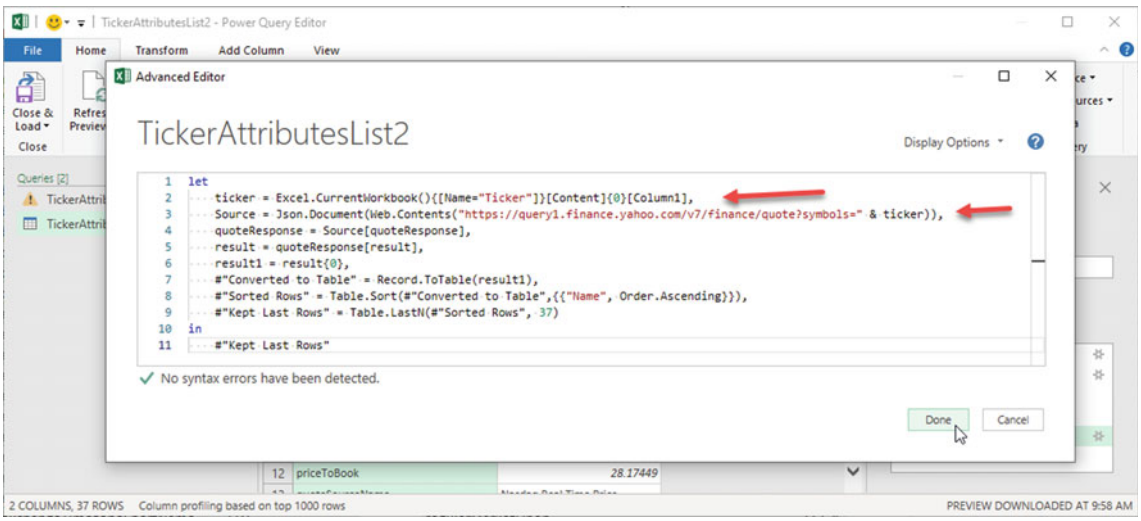
(5) Click on *Continue*.



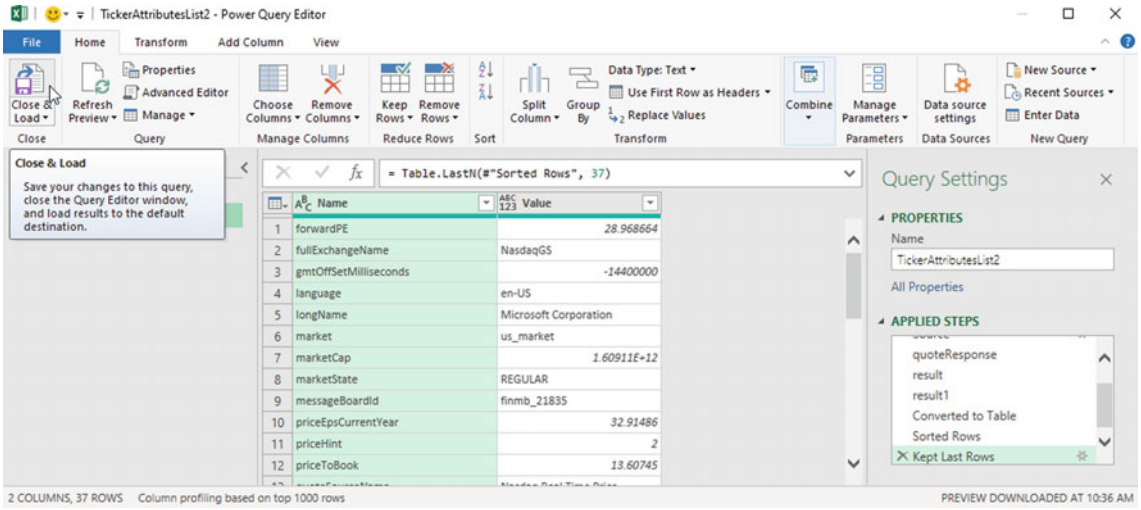
(6) Change privacy to “public.”



(7) Let’s do the same thing for the *TickerAttributesList2* query.

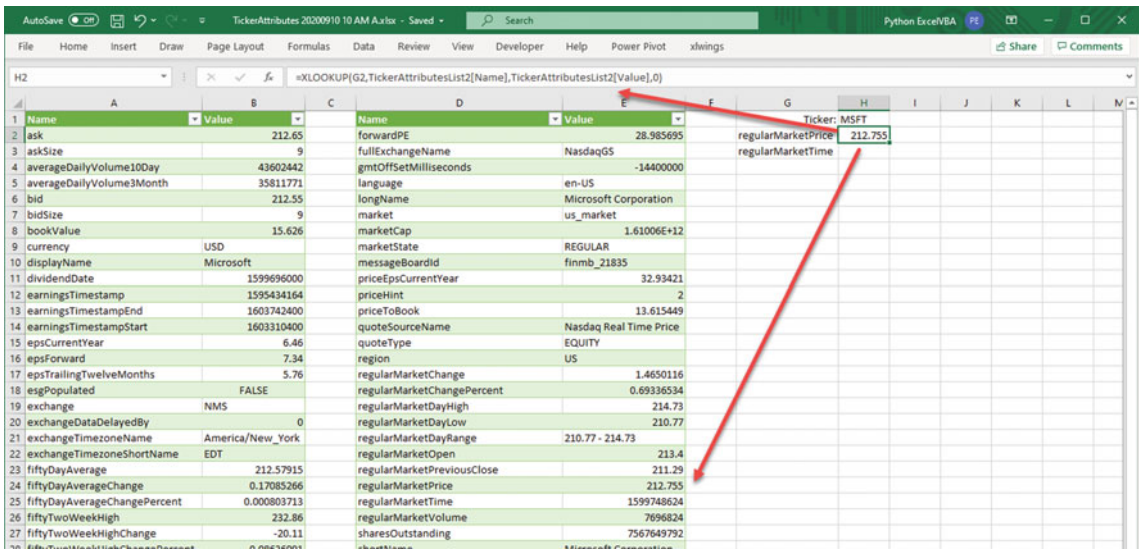


(8) Click *Close and Load.*

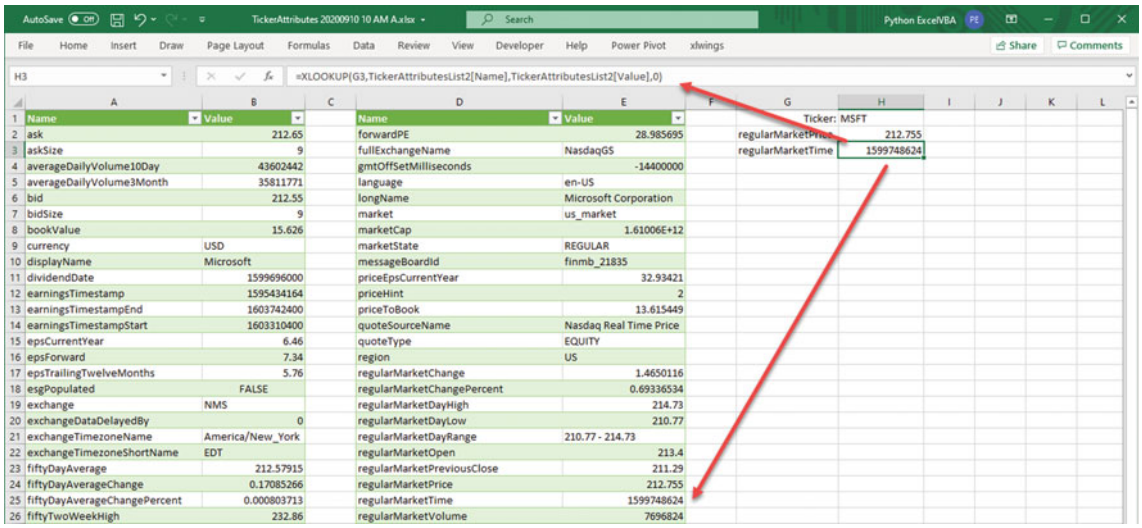


(9) In cell h2, lets put in the following Excel formula to retrieve the stock price,

a. =XLOOKUP(G2,TickerAttributesList2[Name],TickerAttributesList2[Value],0)

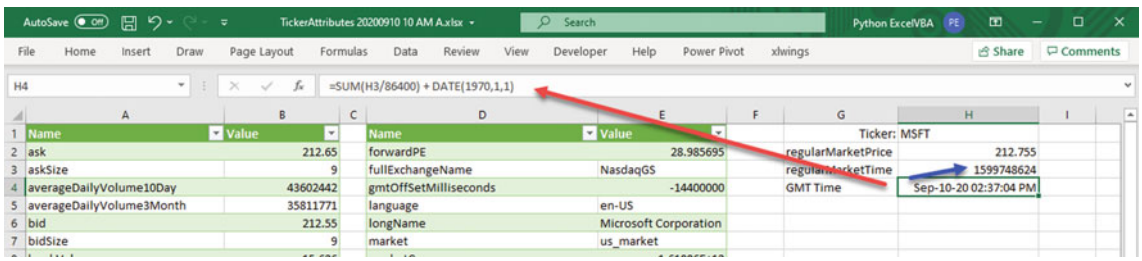


(10) In cell h3, lets put in the following Excel formula to retrieve the market time,
   a. =XLOOKUP(G3,TickerAttributesList2[Name],TickerAttributesList2[Value],0)

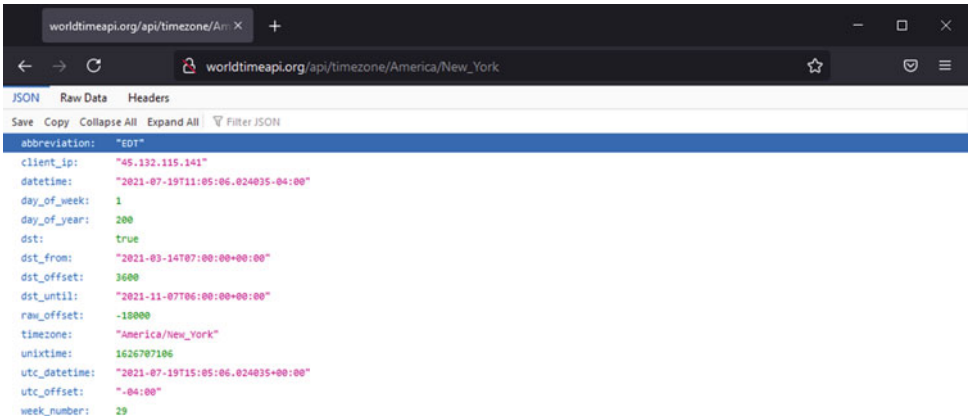


All financial system report time in GMT epoch time, which is London time. GMT time is known as Coordinated Universal Time or UTC time.

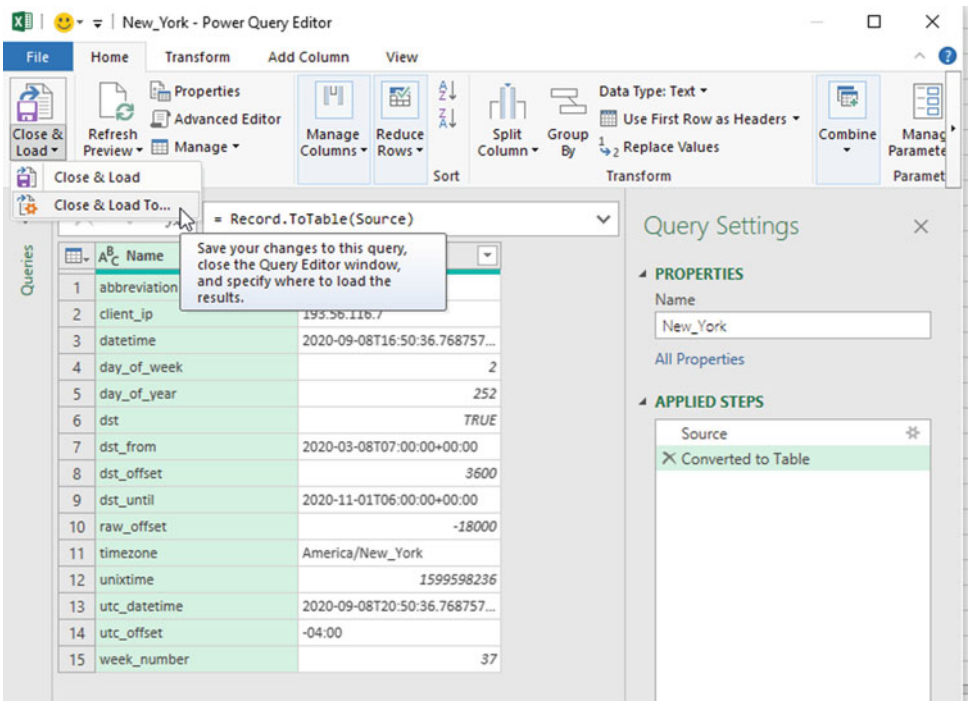
(11) The formula in cell h3 indicates the London time for the epoch time in cell h2.



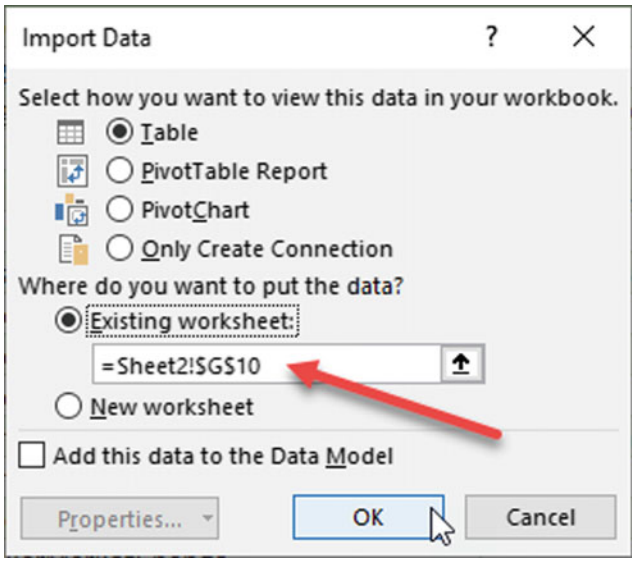
When converting GMT/UTC time to New York time, we either need to subtract by 5 or 4 h depending on if it's daylight savings time or not. The API http://worldtimeapi.org/api/timezone/America/New_York indicates the UTC adjustment needs to get New York time. This API returns a JSON response.



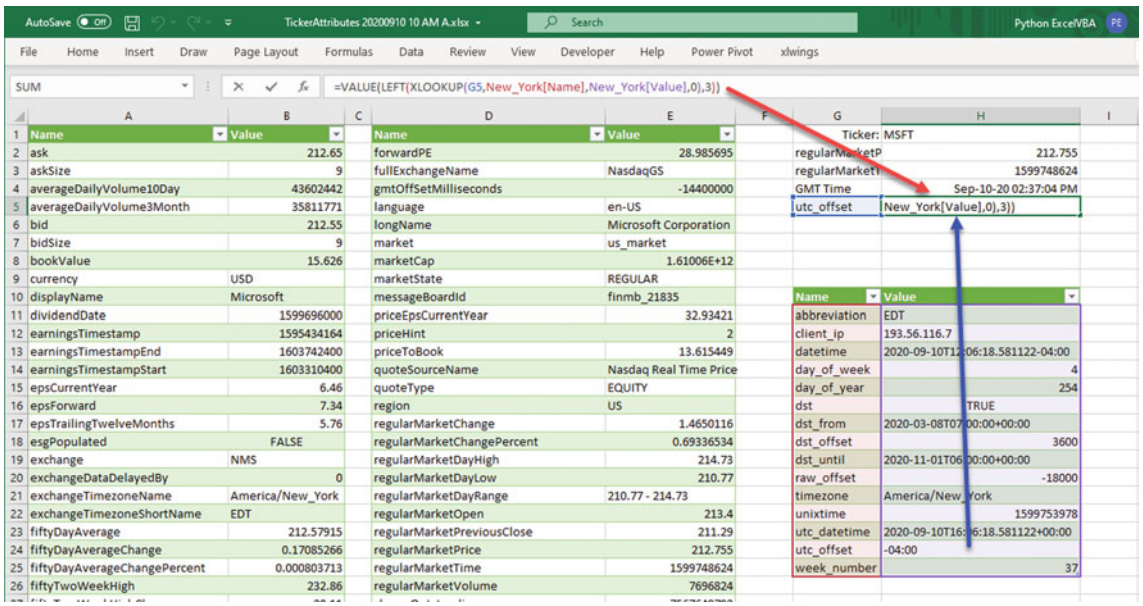
When loading the JSON response, click on the *Close & Load To*.



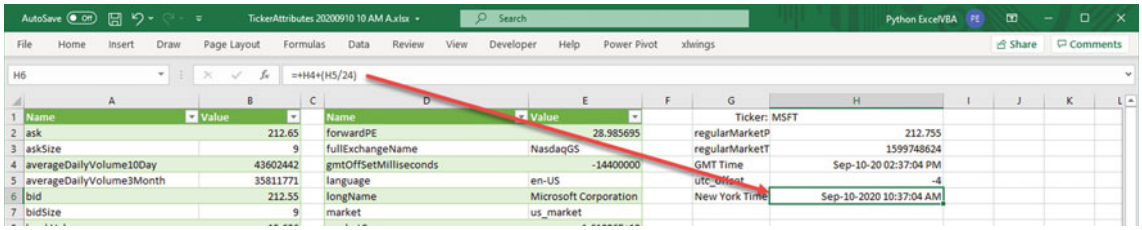
Indicate cell g10 on Sheet2.



(12) The utc_offset value in cell h24 is a string. The formula in cell h5 converts the cell h24 to a number.



(13) Below shows the New York time.



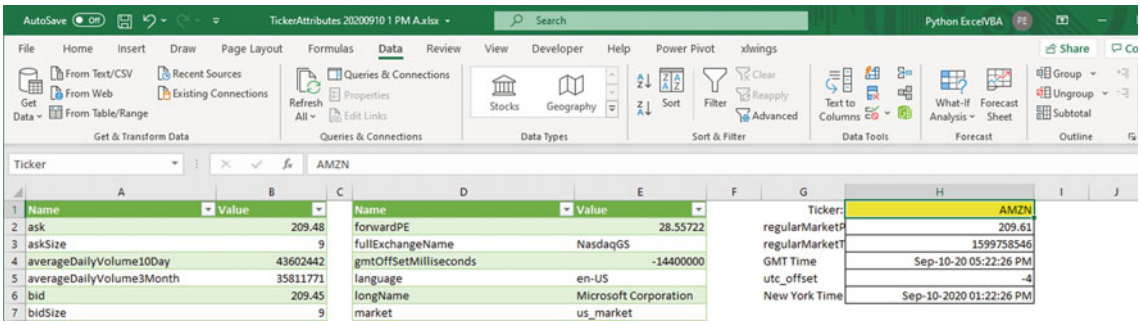
2.6.1.3 Microsoft Excel Ticker Attribute Template

We have spent a lot of manual time and effort setting up 3 *Power Queries* to retrieve a stock ticker's attributes. We configured it so *Power Query* will retrieve the ticker attributes based on range name.

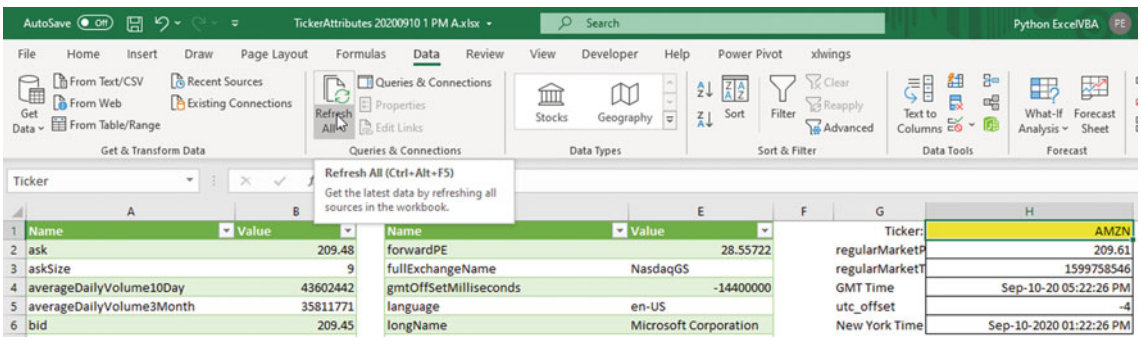
Amazon Ticker Attributes

In this section, we will retrieve Amazon ticker attributes for September 10, 2020. The ticker symbol for Amazon is *AMZN*.

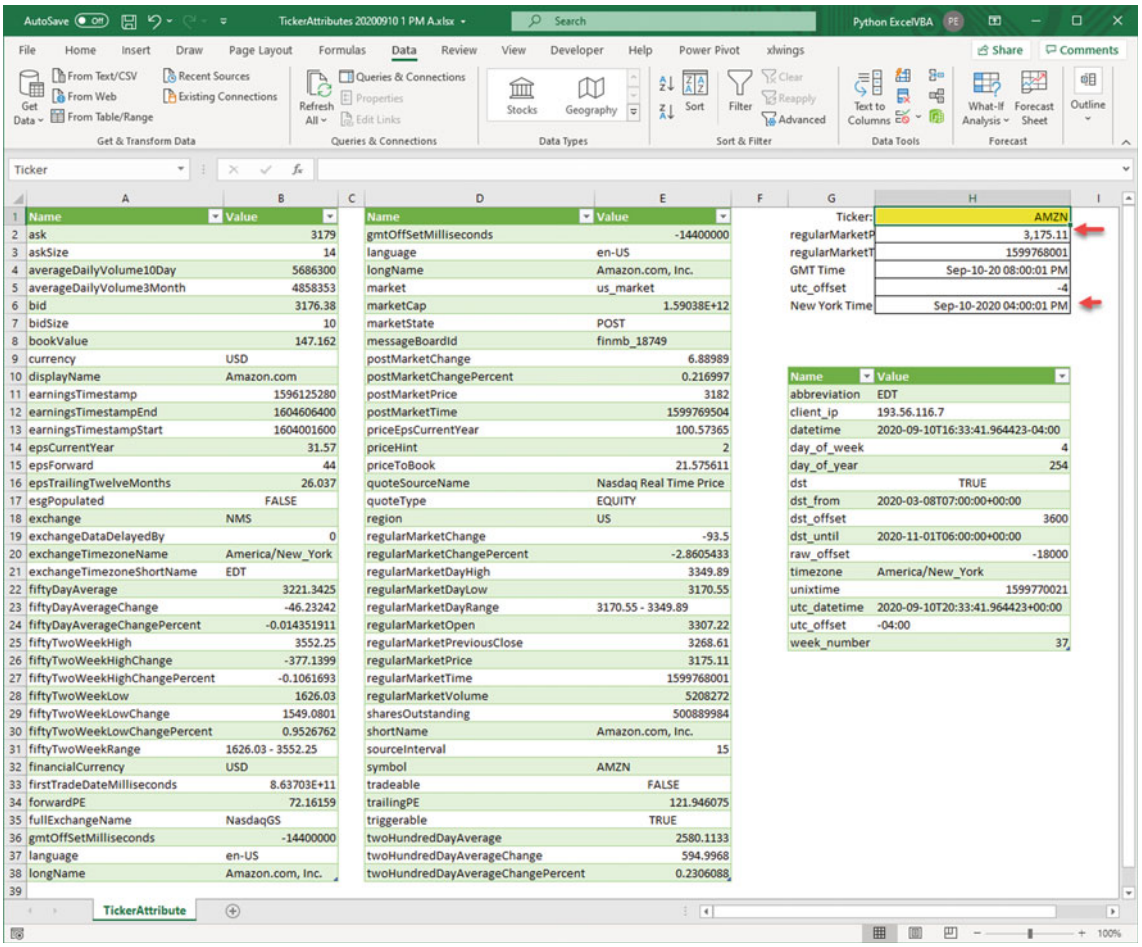
(1) In cell H1, enter *AMZN*.



(2) In the *Data* tab, click on *Refresh All*.



(3) The following are AMZN's ticker attributes for Thursday, September 10, 2020, after the refresh.



Name	Value
ask	3179
askSize	14
averageDailyVolume10Day	5686300
averageDailyVolume3Month	4858353
bid	3176.38
bidSize	10
bookValue	147.162
currency	USD
displayName	Amazon.com
earningsTimestamp	1596125280
earningsTimestampEnd	1604606400
earningsTimestampStart	1604001600
epsCurrentYear	31.57
epsForward	44
epsTrailingTwelveMonths	26.037
esgPopulated	FALSE
exchange	NMS
exchangeDataDelayedBy	0
exchangeTimezoneName	America/New_York
exchangeTimezoneShortName	EDT
fiftyDayAverage	3221.3425
fiftyDayAverageChange	-46.23242
fiftyDayAverageChangePercent	-0.014351911
fiftyTwoWeekHigh	3552.25
fiftyTwoWeekHighChange	-377.1399
fiftyTwoWeekHighChangePercent	-0.1061693
fiftyTwoWeekLow	1626.03
fiftyTwoWeekLowChange	1549.0801
fiftyTwoWeekLowChangePercent	0.9526762
fiftyTwoWeekRange	1626.03 - 3552.25
financialCurrency	USD
firstTradeDateMilliseconds	8.63703E+11
forwardPE	72.16159
fullExchangeName	NasdaqGS
gmtOffSetMilliseconds	-14400000
language	en-US
longName	Amazon.com, Inc.

Name	Value
gmtOffSetMilliseconds	-14400000
language	en-US
longName	Amazon.com, Inc.
market	us_market
marketCap	1.59038E+12
marketState	POST
messageBoardId	finmb_18749
postMarketChange	6.88989
postMarketChangePercent	0.216997
postMarketPrice	3182
postMarketTime	1599769504
priceEpsCurrentYear	100.57365
priceHint	2
priceToBook	21.575611
quoteSourceName	Nasdaq Real Time Price
quoteType	EQUITY
region	US
regularMarketChange	-93.5
regularMarketChangePercent	-2.8605433
regularMarketDayHigh	3349.89
regularMarketDayLow	3170.55
regularMarketDayRange	3170.55 - 3349.89
regularMarketOpen	3307.22
regularMarketPreviousClose	3268.61
regularMarketPrice	3175.11
regularMarketTime	1599768001
regularMarketVolume	5208272
sharesOutstanding	500889984
shortName	Amazon.com, Inc.
sourceInterval	15
symbol	AMZN
tradeable	FALSE
trailingPE	121.946075
triggerable	TRUE
twoHundredDayAverage	2580.1133
twoHundredDayAverageChange	594.9968
twoHundredDayAverageChangePercent	0.2306088

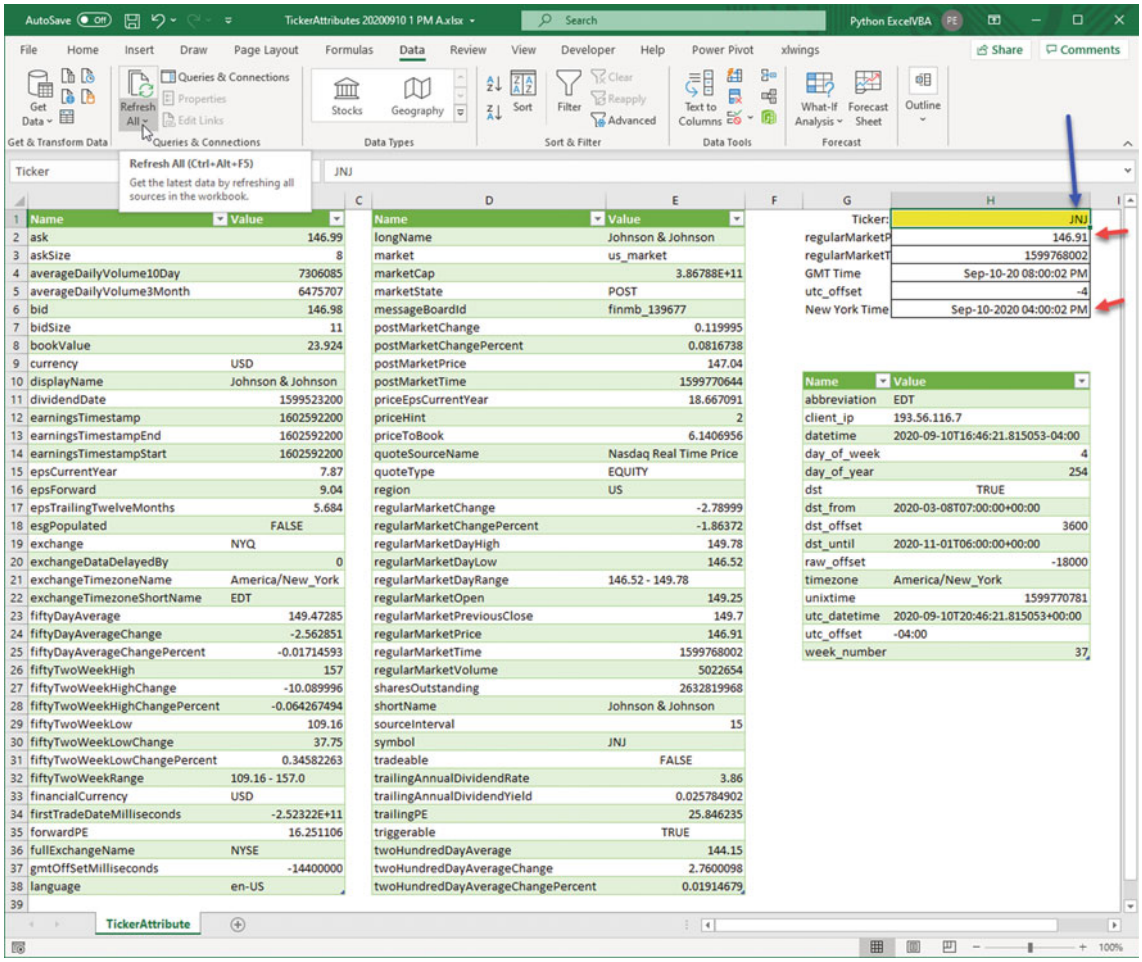
Ticker:	AMZN
regularMarketP	3,175.11
regularMarketT	1599768001
GMT Time	Sep-10-20 08:00:01 PM
utc_offset	-4
New York Time	Sep-10-2020 04:00:01 PM

Name	Value
abbreviation	EDT
client_ip	193.56.116.7
datetime	2020-09-10T16:33:41.964423-04:00
day_of_week	4
day_of_year	254
dst	TRUE
dst_from	2020-03-08T07:00:00+00:00
dst_offset	3600
dst_until	2020-11-01T06:00:00+00:00
raw_offset	-18000
timezone	America/New_York
unixtime	1599770021
utc_datetime	2020-09-10T20:33:41.964423+00:00
utc_offset	-04:00
week_number	37

We can confirm that the ticker attributes are correct for AMZN by going to https://finance.yahoo.com/quote/amzn.

Johnson and Johnson Ticker Attributes

This section will use the Microsoft Excel ticker attribute that we created to retrieve Johnson and Johnson's ticker attributes. The ticker for Johnson and Johnson is JNJ. To do this, enter the ticker JNJ in cell H1 and then click on *Refresh All* in the *Data* tab.



We can confirm that the ticker attributes are correct for JNJ by going to https://finance.yahoo.com/quote/jnj.

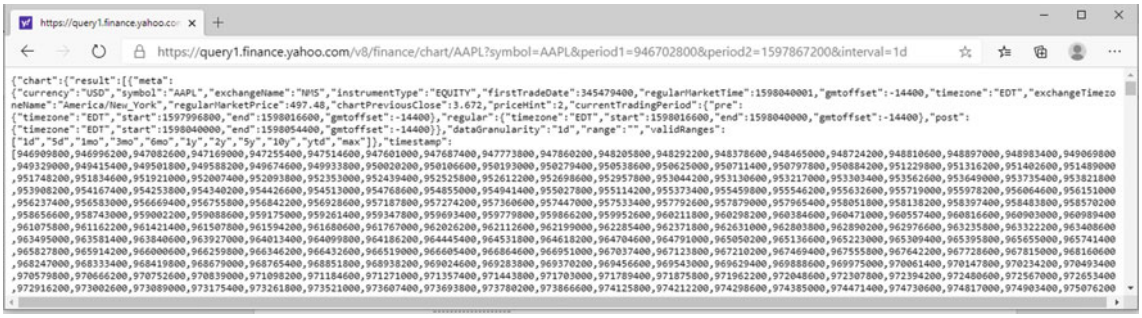
2.7 Historical Data

2.7.1 Yahoo! Finance API

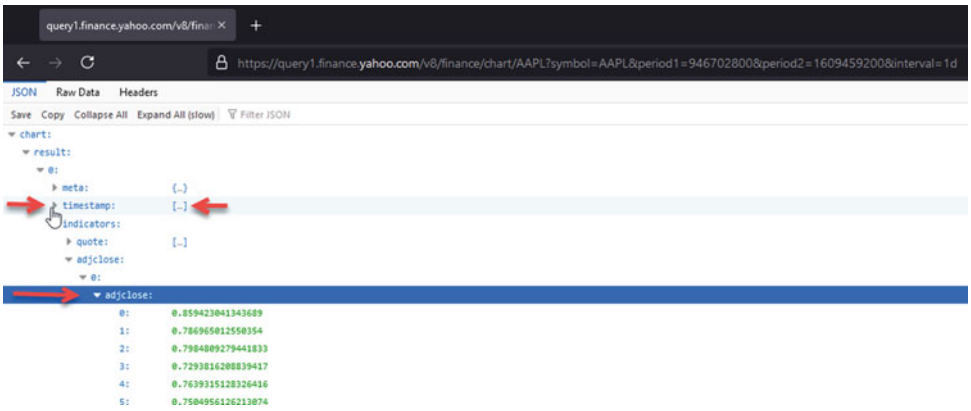
In this section, we will demonstrate the historical data Yahoo! Finance API. The following is the URL to return Apple Inc.'s historical daily prices from January 1, 2000, to January 01, 2021.

https://query1.finance.yahoo.com/v8/finance/chart/AAPL?symbol=AAPL&period1=946702800&period2=1609459200&interval=1d.

Below shows Apple Inc.'s JSON data set in Microsoft Edge web browser. Microsoft Edge shows Apple Inc.'s JSON data set unformatted; this makes it very hard to review and analyze Apple Inc.'s data set.



Below shows Apple Inc.'s JSON data set in the Firefox web browser.



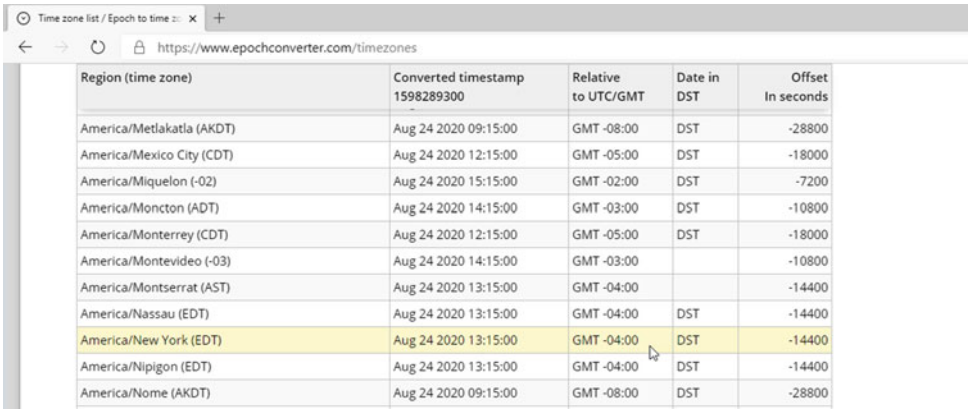
Firefox will auto-detect a JSON data set. Once Firefox detects a JSON data set, it will format it in a human-readable form, as shown above. It will also put a horizontal or vertical gray arrow next to each element that has a child element. For example, the *timestamp* element has a horizontal gray arrow to its left; this indicates that there are child elements under the *timestamp* element. One can click on the horizontal gray arrow to see the child elements of the *timestamp* element. Also, there are three dots inside two square brackets next to the *timestamp* element; this also indicates that there are child elements under the *timestamp* element.

The horizontal arrow will become a vertical gray arrow when the child elements are displayed. Each child element is numbered, beginning with the number 0; this makes it easier to identify each child element. The element *adjclose* has a vertical gray arrow to indicate that all child elements are displayed.

2.7.2 Epoch Time

The Apple Inc's. Yahoo! Finance API uses Epoch time to indicate period 1 and period 2 of the data set. Go to the URL https://www.maketecheasier.com/what-is-epoch-time/ to learn more about Epoch time. For period 1, we used 946702800 epoch time to indicate January 1, 2000, 12:00:00 AM. For period 2 we used 1609459200 to indicate January 01, 2021, 12:00:00 AM. Go to the URL https://www.epochconverter.com/ to convert between epoch time and regular time.

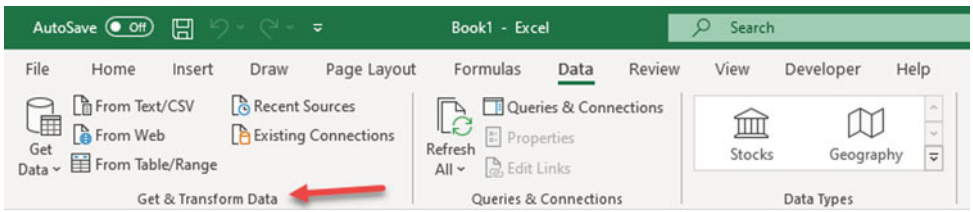
It is important to note that *Epoch Time* is based on Greenwich Mean time or GMT time. London is in the GMT time zone. All other time zones must adjust to GMT time. To get New York City time, one would subtract 5 or 4 h from the stated GMT time. The adjustment of 4 or 5 h is based on whether the date is daylight saving time. The URL https://www.epochconverter.com/timezones has a table for GMT time adjustment for all the major cities around the world.



Region (time zone)	Converted timestamp 1598289300	Relative to UTC/GMT	Date in DST	Offset In seconds
America/Metlakatla (AKDT)	Aug 24 2020 09:15:00	GMT -08:00	DST	-28800
America/Mexico City (CDT)	Aug 24 2020 12:15:00	GMT -05:00	DST	-18000
America/Miquelon (-02)	Aug 24 2020 15:15:00	GMT -02:00	DST	-7200
America/Moncton (ADT)	Aug 24 2020 14:15:00	GMT -03:00	DST	-10800
America/Monterrey (CDT)	Aug 24 2020 12:15:00	GMT -05:00	DST	-18000
America/Montevideo (-03)	Aug 24 2020 14:15:00	GMT -03:00		-10800
America/Montserrat (AST)	Aug 24 2020 13:15:00	GMT -04:00		-14400
America/Nassau (EDT)	Aug 24 2020 13:15:00	GMT -04:00	DST	-14400
America/New York (EDT)	Aug 24 2020 13:15:00	GMT -04:00	DST	-14400
America/Nipigon (EDT)	Aug 24 2020 13:15:00	GMT -04:00	DST	-14400
America/Nome (AKDT)	Aug 24 2020 09:15:00	GMT -08:00	DST	-28800

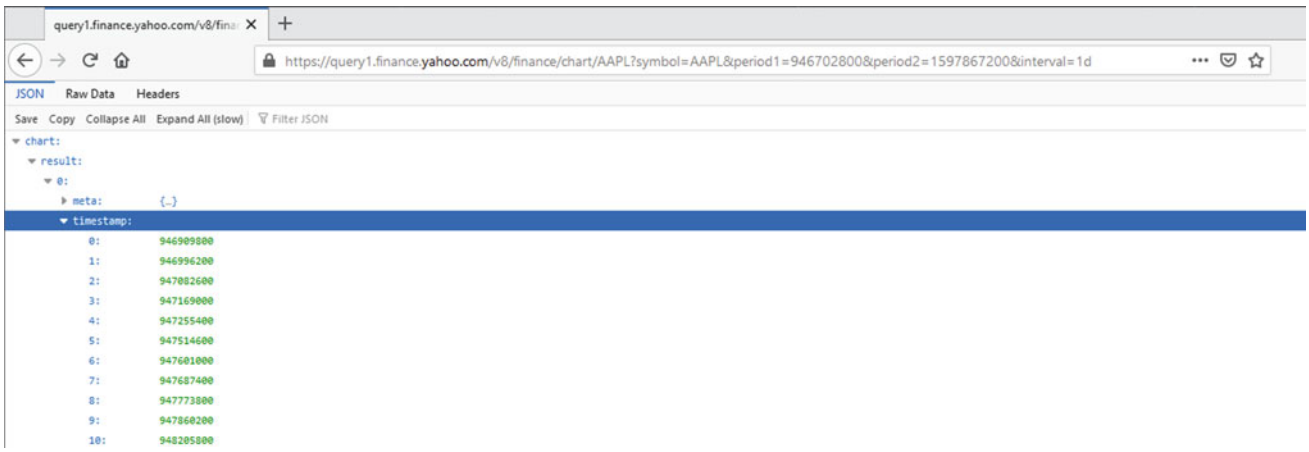
2.7.3 Power Query

We will use *Power Query* to retrieve, format, and understand a JSON data set in Microsoft Excel. *Power Query* became a standard feature in Microsoft Excel 2016. *Power Query* is part of the *Get & Transform Data* section in the *Data* tab.

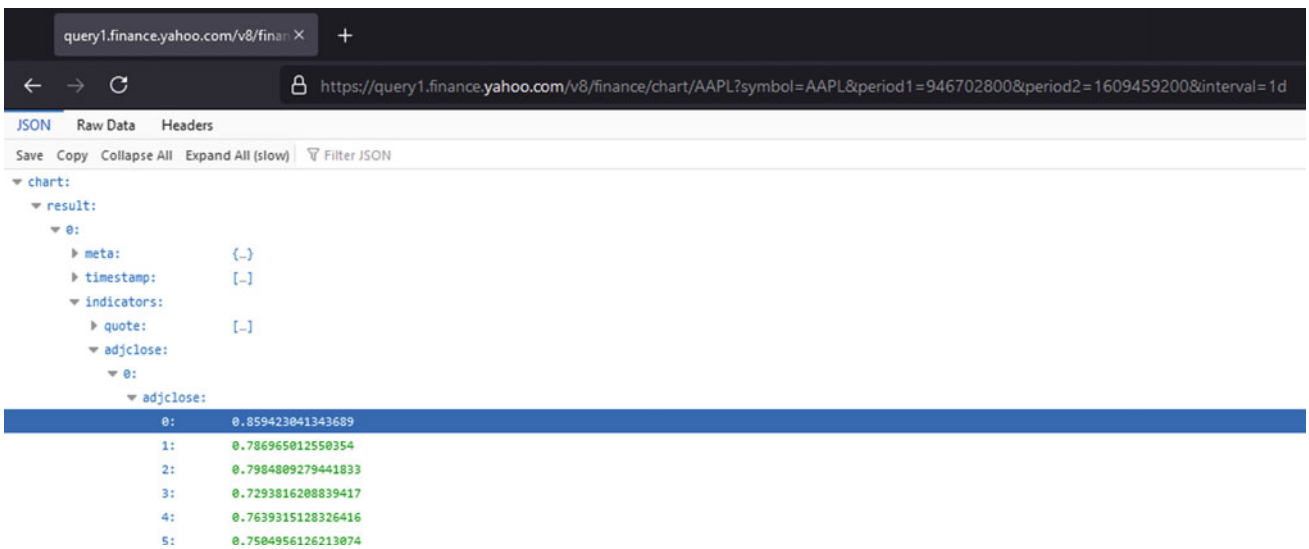


Apple Inc.'s JSON data set is large and complex. We will use Firefox to help guide us on how to configure *Power Query* to retrieve AAPL's historical prices. Inexperience *Power Query* users would try to create one query to retrieve both the date and price; this is a wrong approach. Separate queries should be created for the date and price. We will configure two queries in *Power Query*. One query will retrieve the date, and one query will retrieve the price. After creating the two queries, we will create another query to merge the date query and price query.

The following view of AAPL's JSON data set in Firefox will help guild us configure *Power Query* to retrieve the dates.



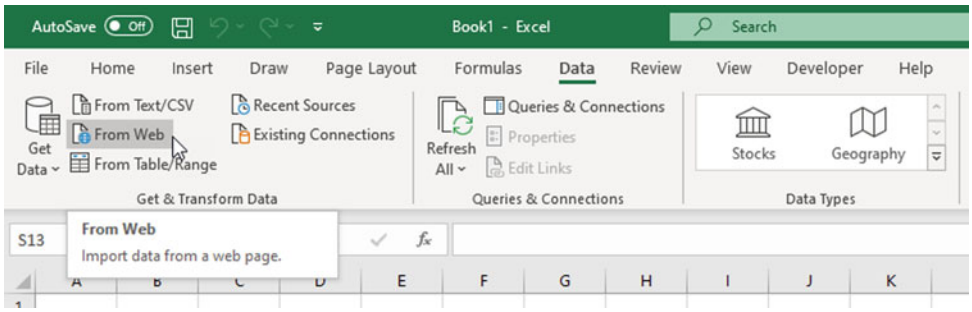
The following view of AAPL's JSON data set in Firefox will help guild us configure *Power Query* to retrieve the historical prices.



2.7.3.1 Date Query

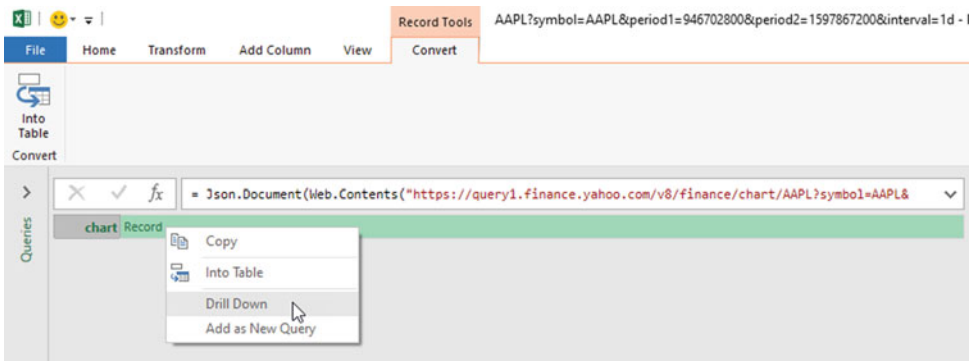
The following are the steps in configuring *Power Query* to retrieve AAPL historical dates:

(1) Click on the *From Web* button in the *Data* tab.

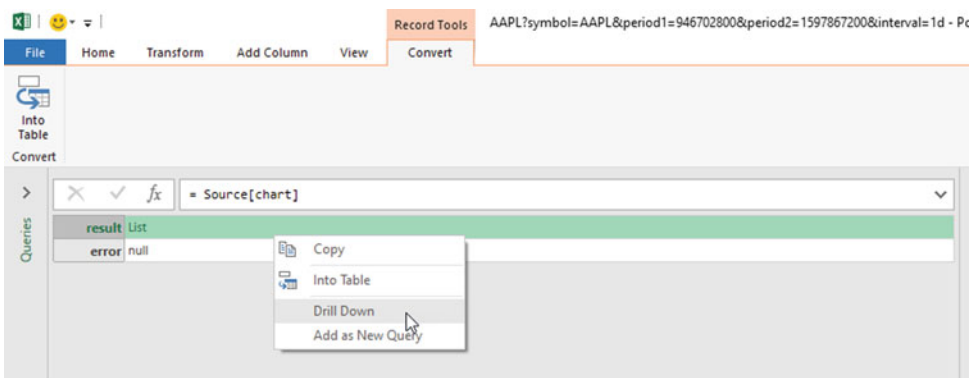


(2) Enter URL to retrieve AAPL's historical prices.

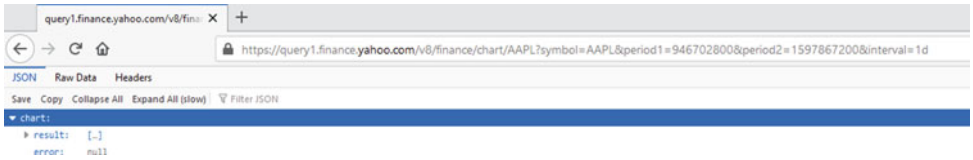
(3) Right-mouse click on *Record* and choose *Drill Down*.



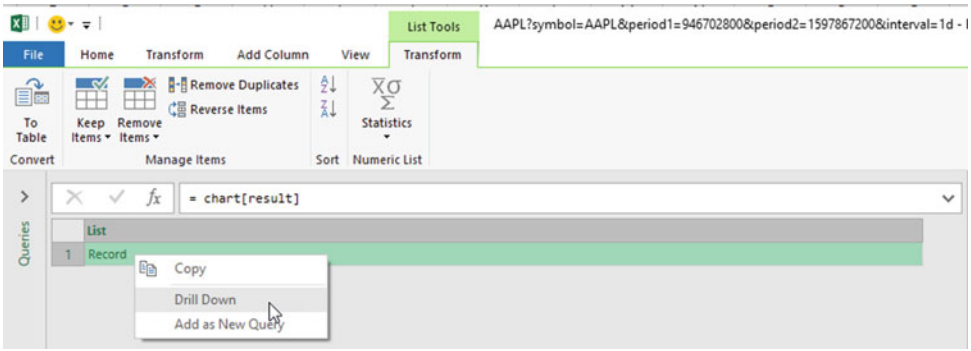
(4) Right-mouse click on *List* and choose *Drill Down*.



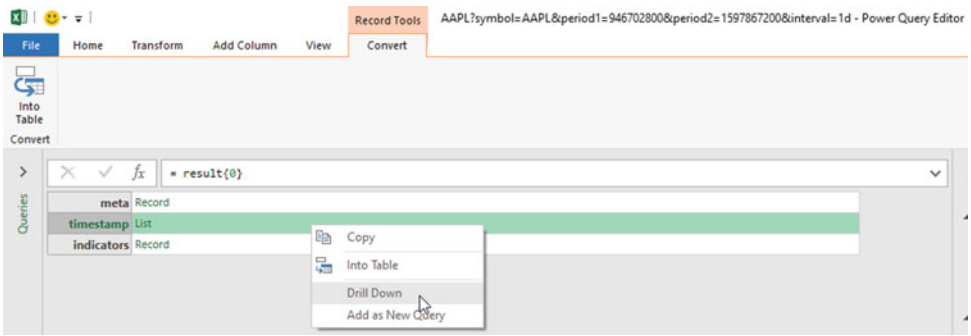
Note that the *Power Query* view mimics the Firefox view.



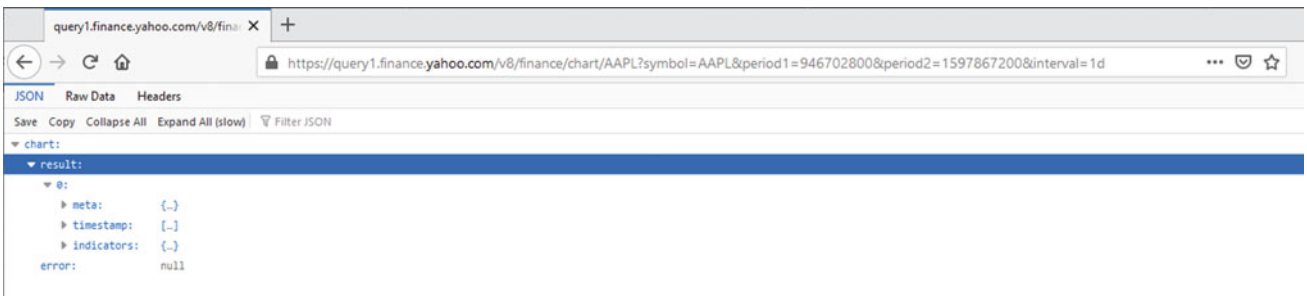
(5) Right-mouse click on *Record* and choose *Drill Down*.



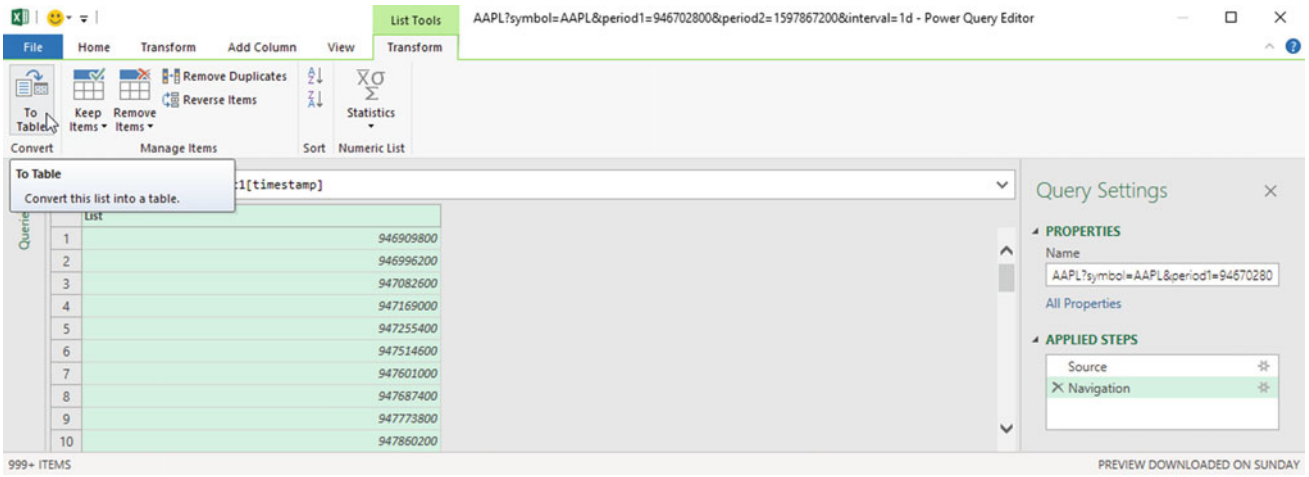
(6) Right-mouse click on *timestamp* and choose *Drill Down.*



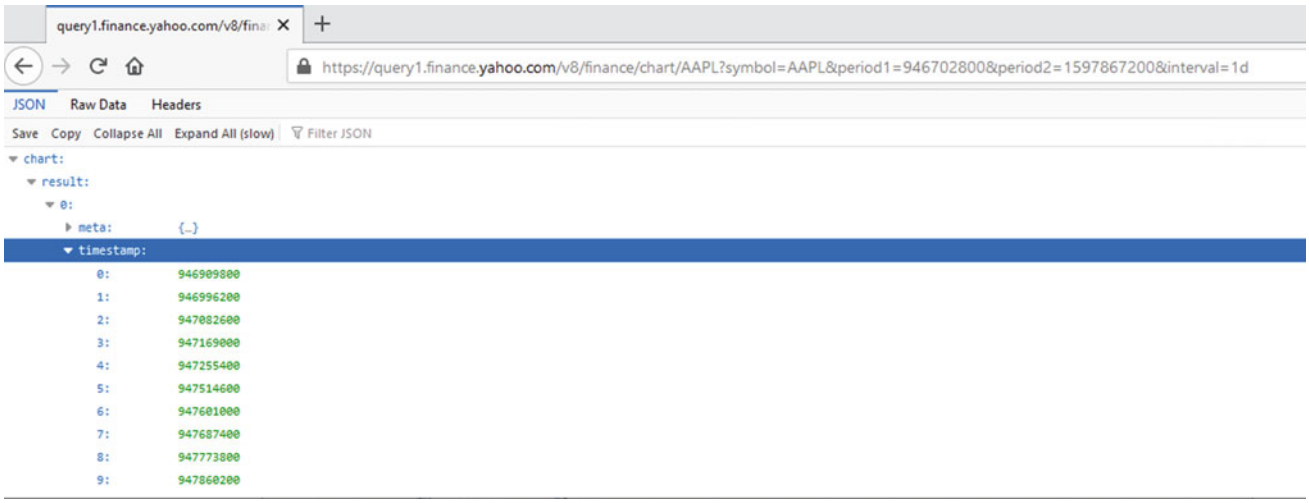
Note that the *Power Query* view mimics the Firefox view.



(7) Click on the *To Table* button.

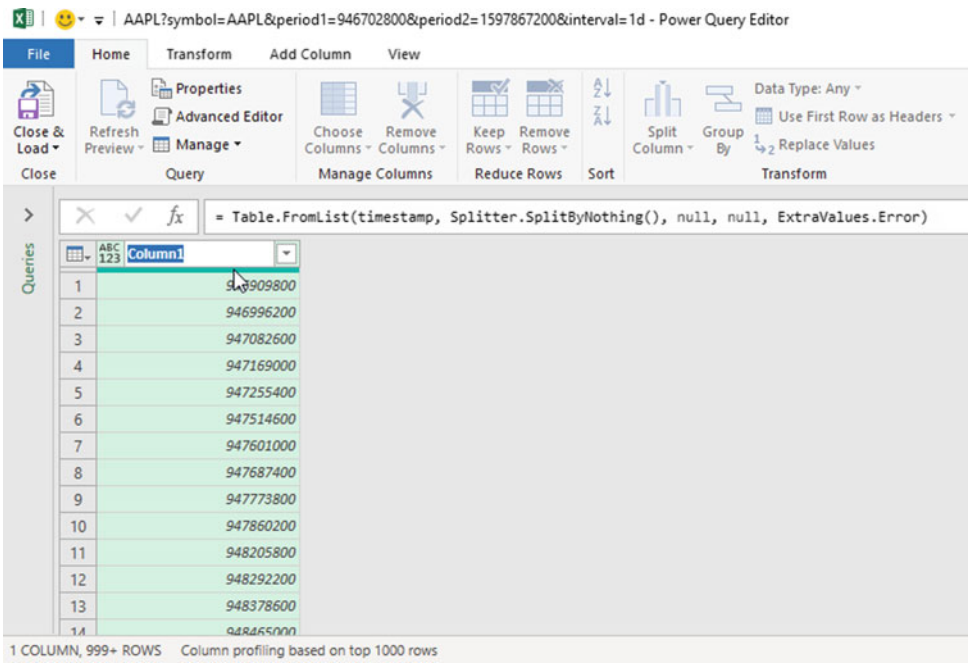


Note that the first 10 Epoch times in *Power Query* match the Firefox view.



(8) Click on *OK*.

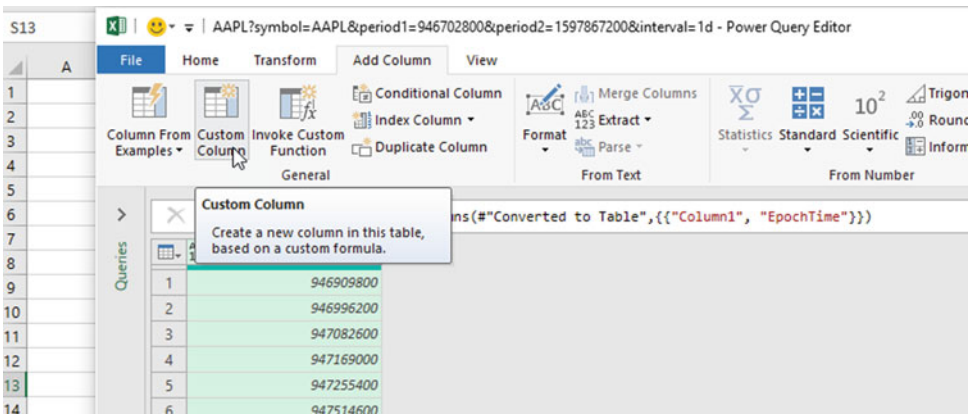
(9) Double-click on the "Column1" field name and rename it "EpochTime."



Custom Query Column

Let's now create a custom query column to convert the epoch time to regular time.

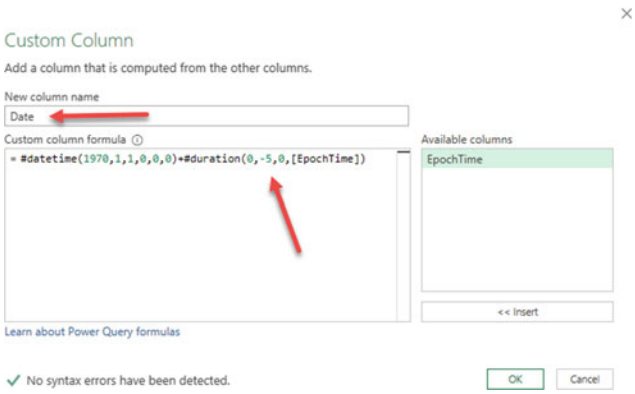
(1) Click on the *Custom Column* button in the *Add Column* tab.



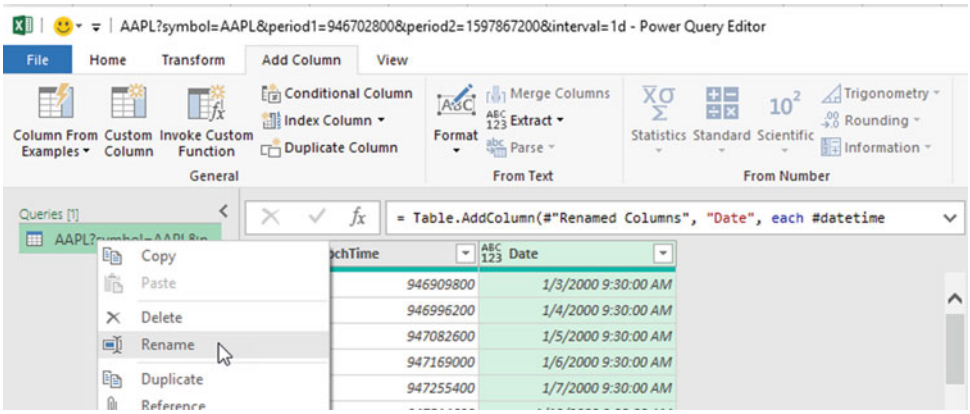
(2) First, name the new column as *Date*. Second, enter the formula,

#datetime(1970,1,1,0,0,0)+#duration(0,−5,0,[EpochTime])

Note the −5 in the formula. This is to convert the GMT time to New York time.



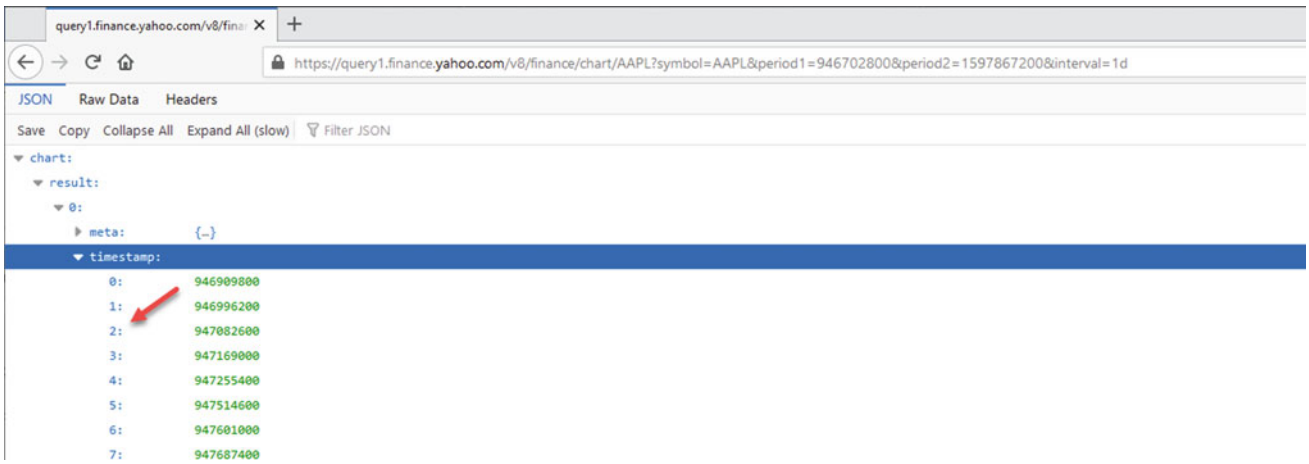
(3) Rename the query to *Date*.



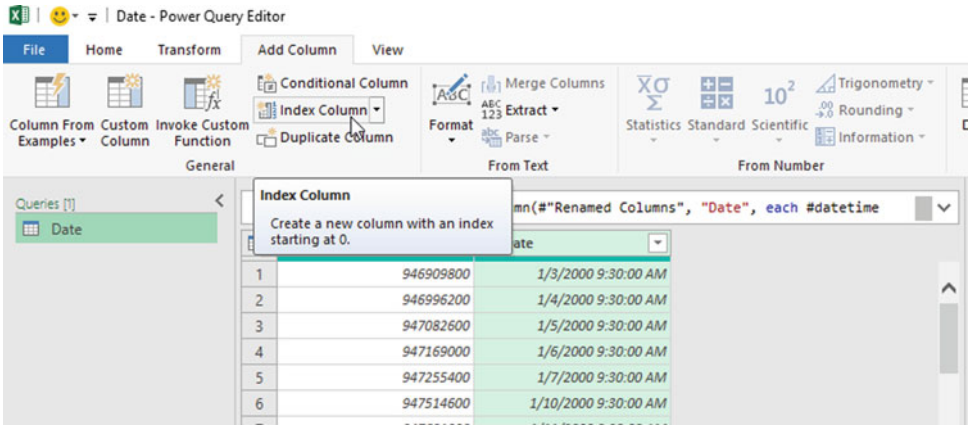
Index Column

We will now add an index column to the date query. This is necessary because, later on, we will want to merge the date query with the price query.

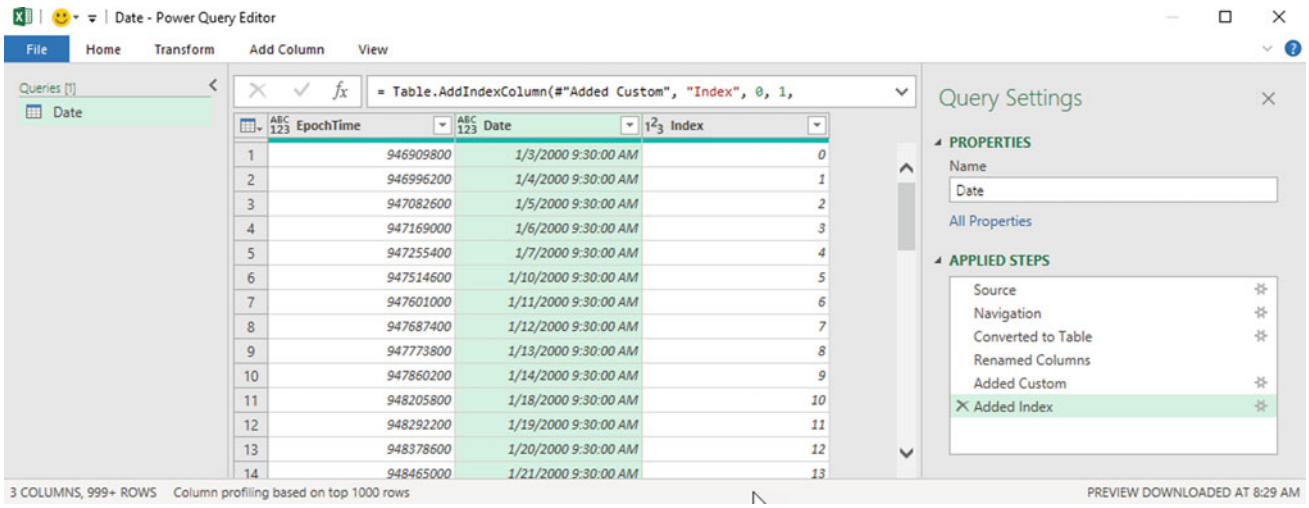
Below are points to the index column of the *timestamp* child.



Click on the *Index Column* button on the *Add Column* tab to add an index column to the date query.



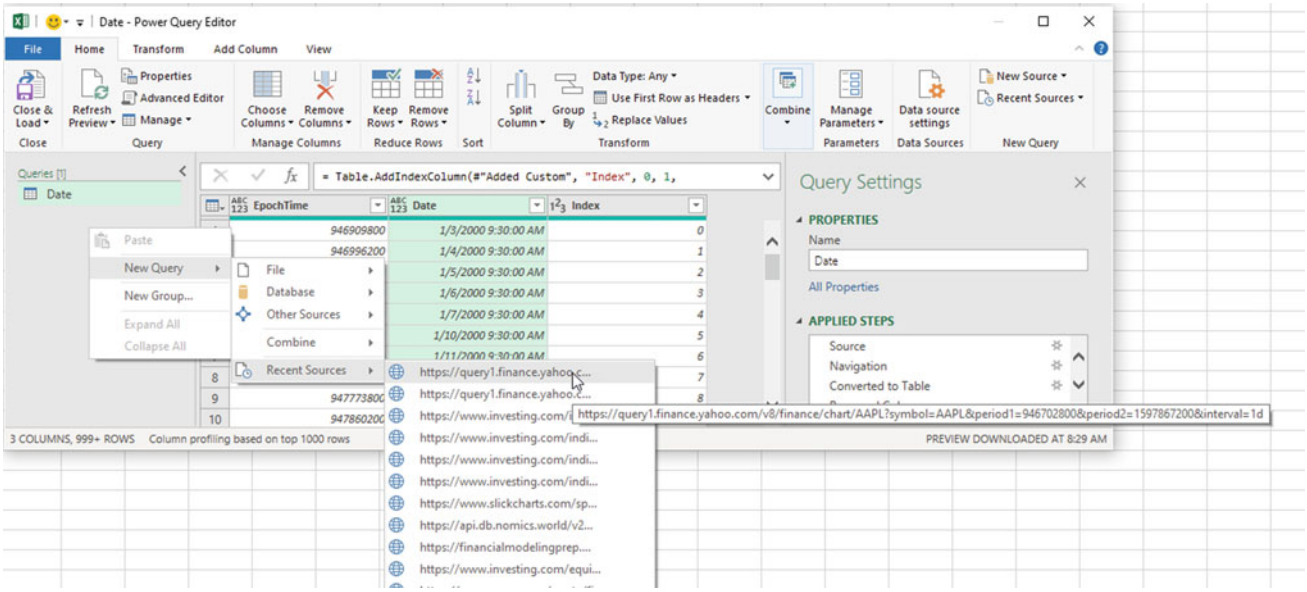
Below shows the final *Date* query.



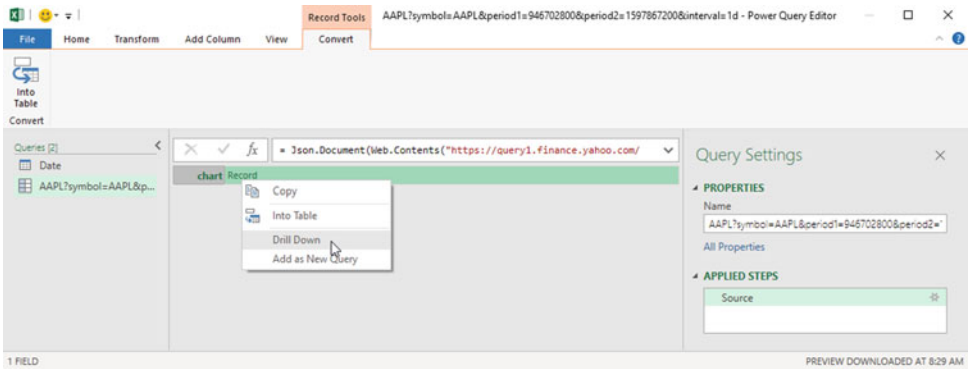
2.7.3.2 Price Query

The following are the steps on how to create a price query. The starting point is from the finished date query discussed above.

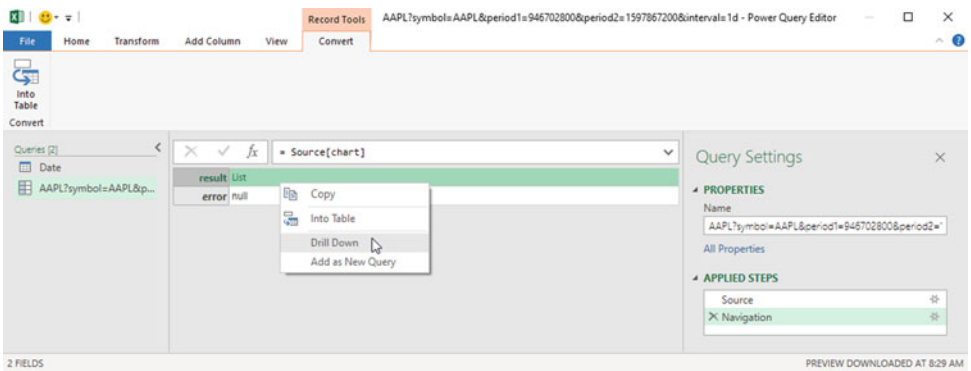
(1) Right-mouse click in the *Queries* section and choose the correct *Recent Source*.



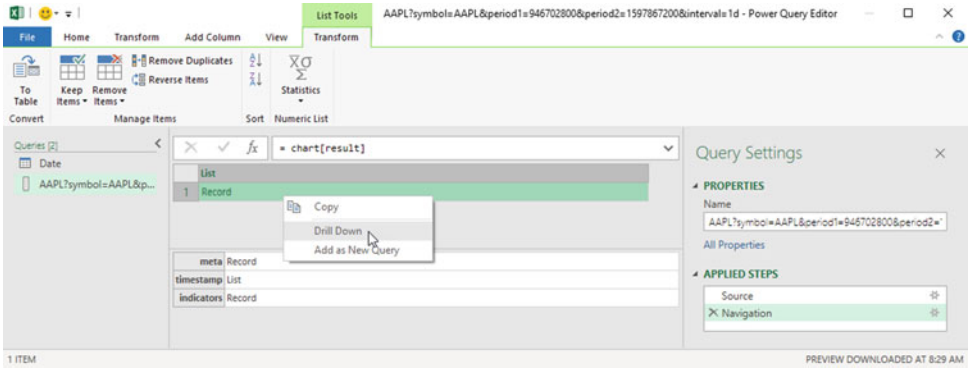
(2) Right-mouse click on *Record* and choose *Drill Down*.



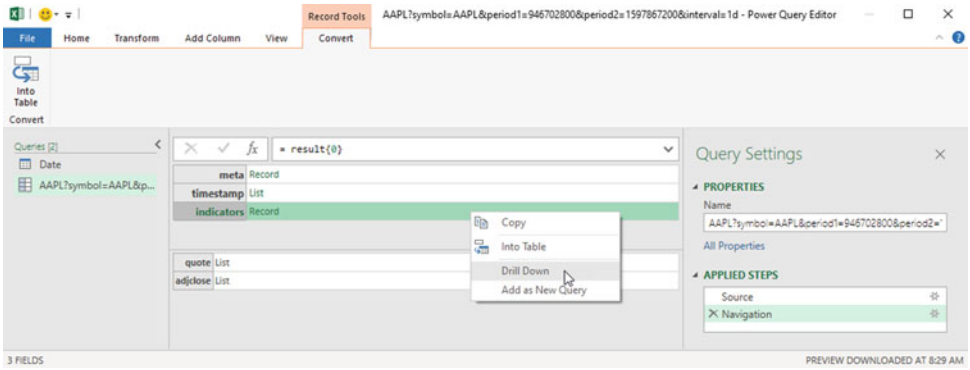
(3) Right-mouse click on *List* and click on *Drill Down*.



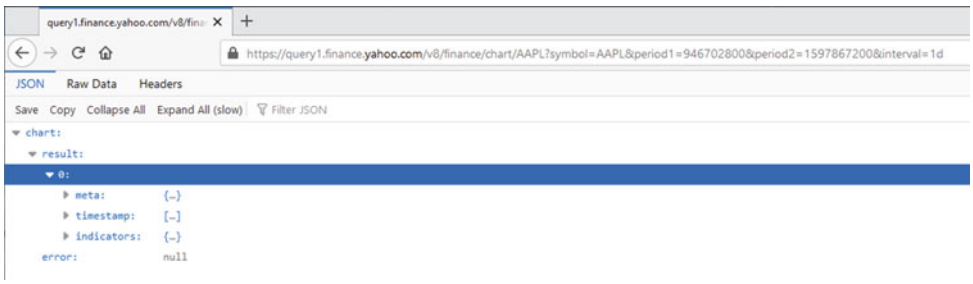
(4) Right-mouse click on *Record* and click on *Drill Down*.



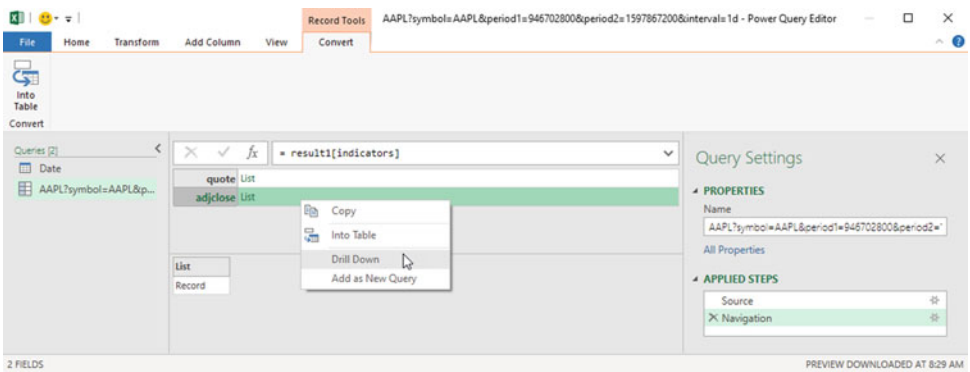
(5) Right-mouse click on *Recor* and click on *Drill Down*.



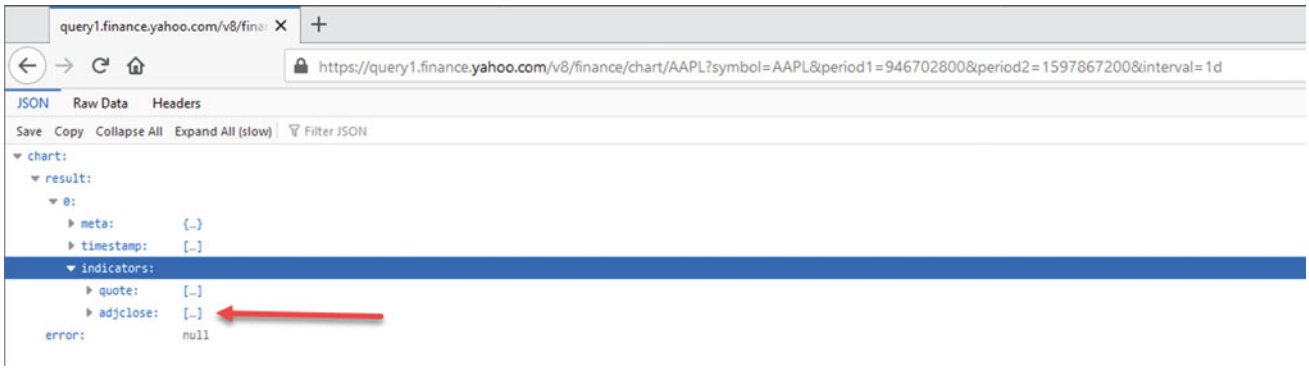
Note that *Power Query* mimics Firefox. *Power Query* has three records. Firefox has three similar elements.



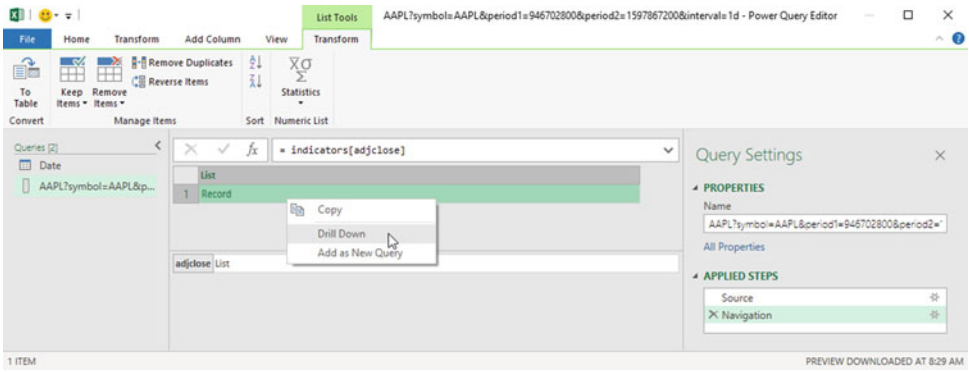
(6) Right-mouse click on *adjclose* and click on *Drill Down*.



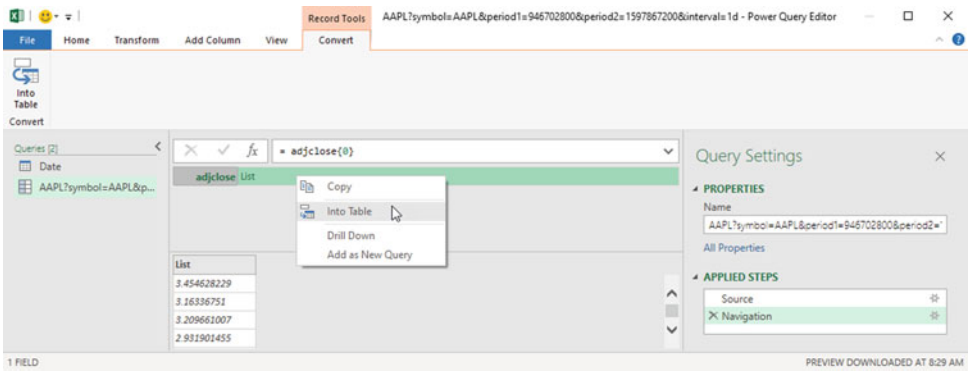
Note that *Power Query* mimics Firefox.



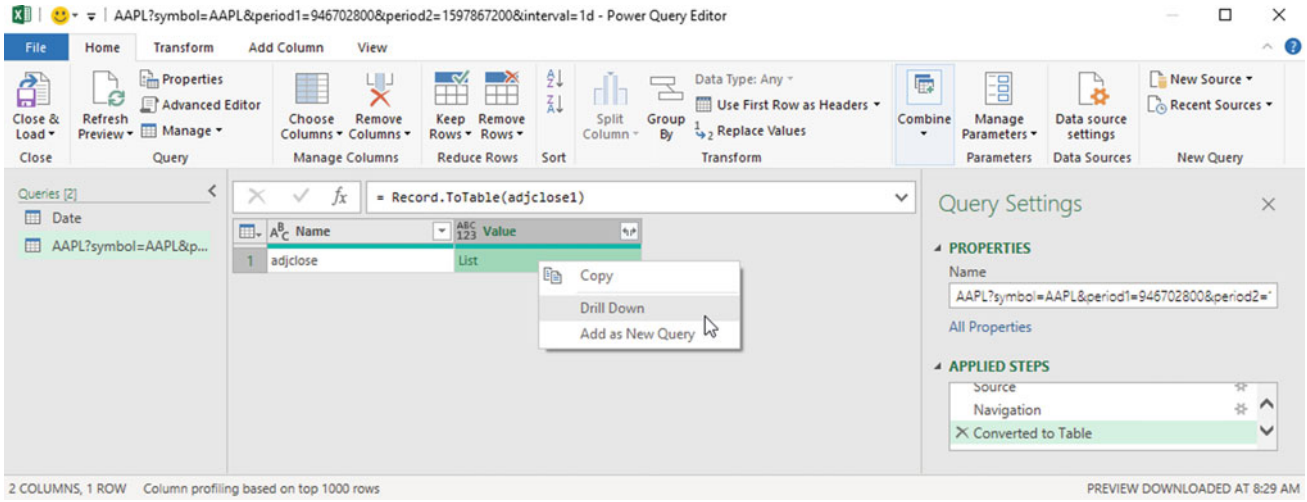
(7) Right-mouse click on *Record* and choose *Drill Down*.



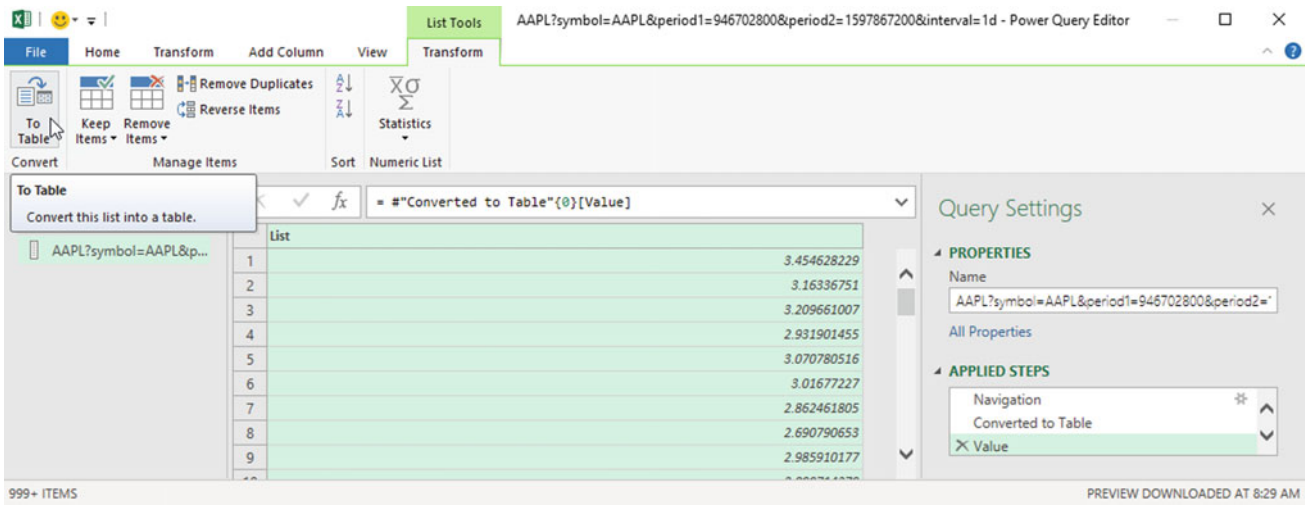
(8) Right-mouse click on *adjclose* and choose *Into Table*.



(9) Right-mouse click on *List* and choose *Drill Down*.

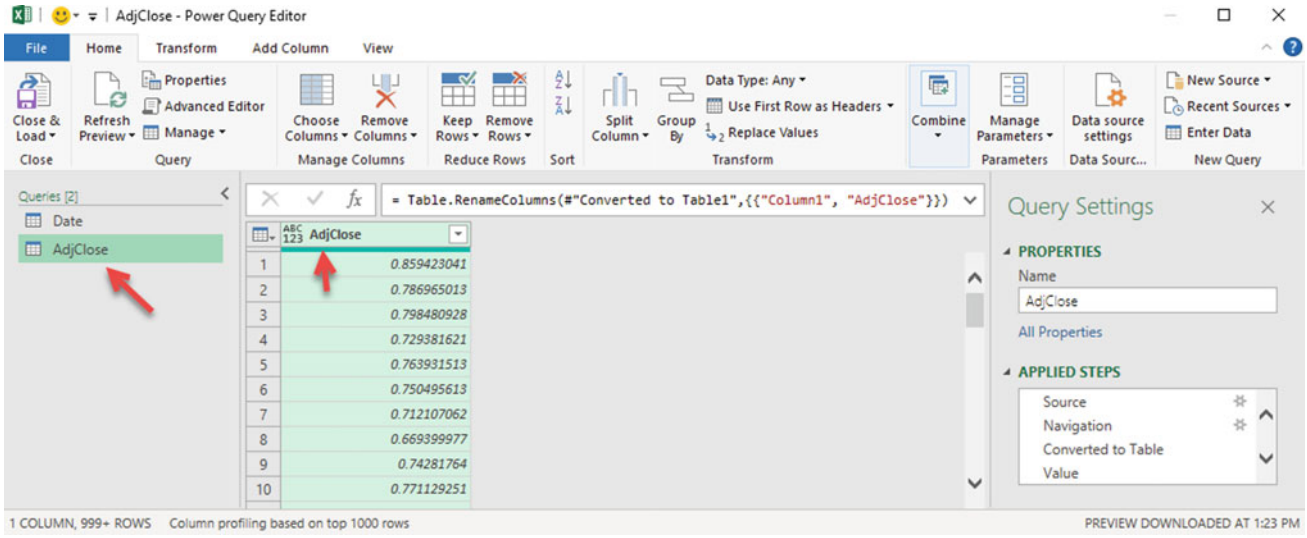


(10) Click on *To Table* in the *Transform* tab.



(11) Click on *OK*.

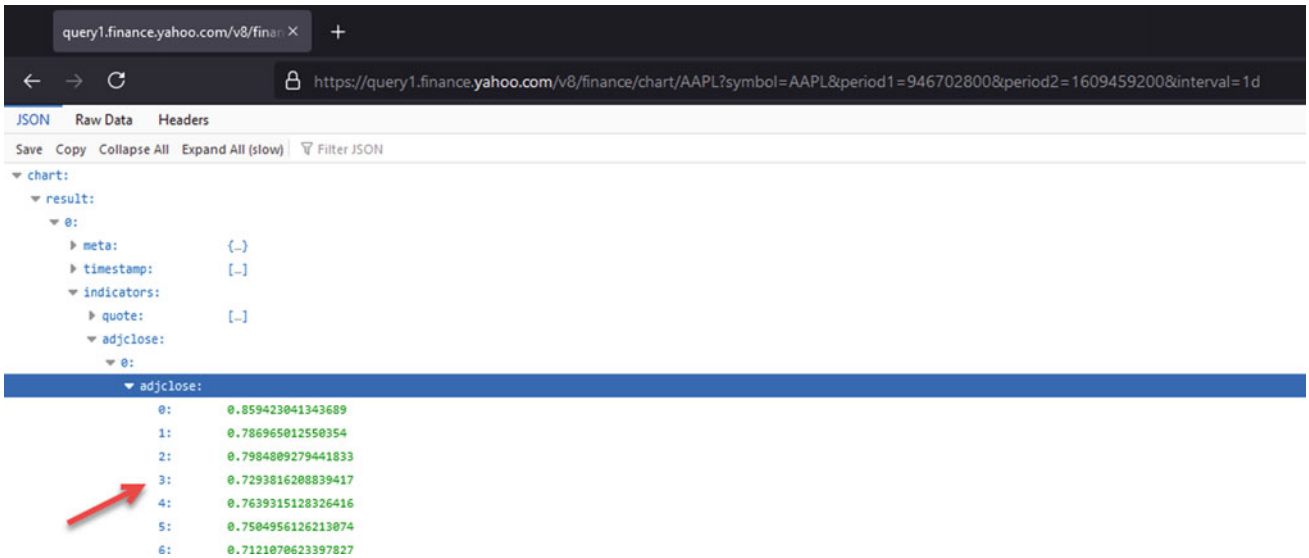
(12) Rename column and query to “AdjCose.”



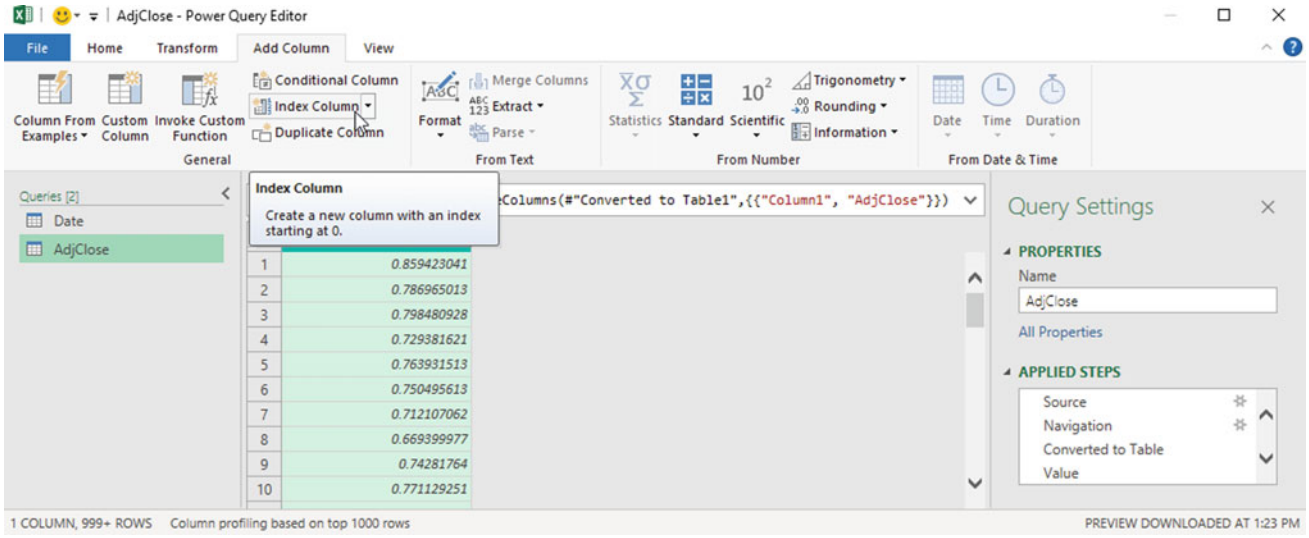
Index Column

We will now add an index column to the price query; this is necessary because, later on, we will want to merge the price query with the date query.

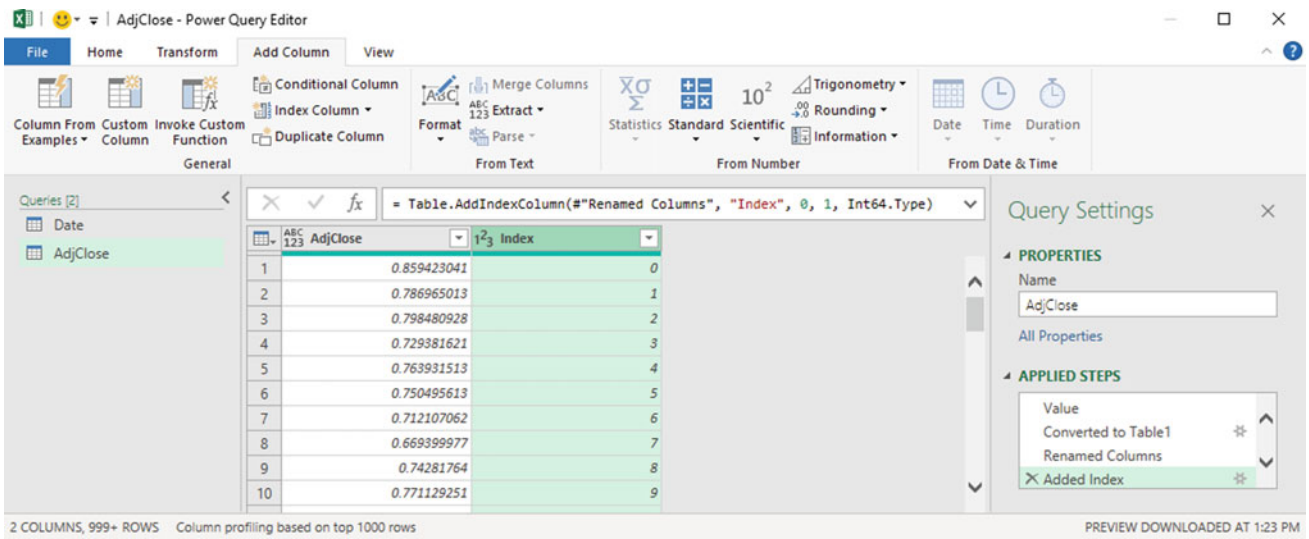
Below are points to the index column of the *adjclose* child.



Click on the *Index Column* button on the *Add Column* tab to add an index column to the date query.



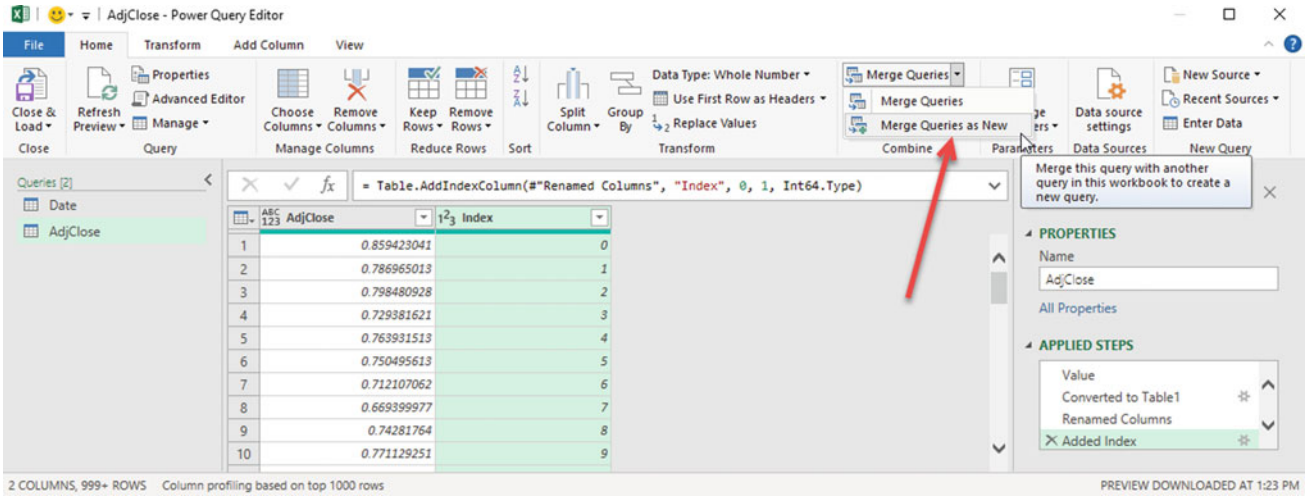
Below shows the final *AdjClose* query.



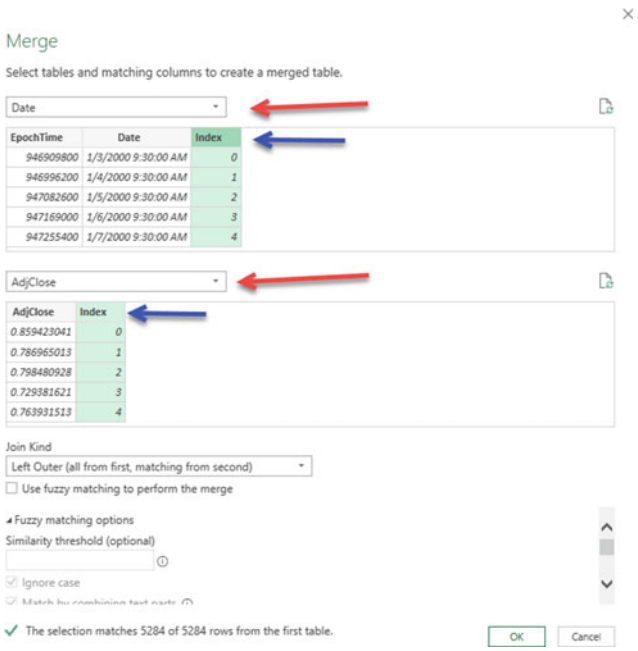
2.7.3.3 Merge Query

We will now create a new query that will have both the date and price columns. The new query will use the "date" and "price" queries.

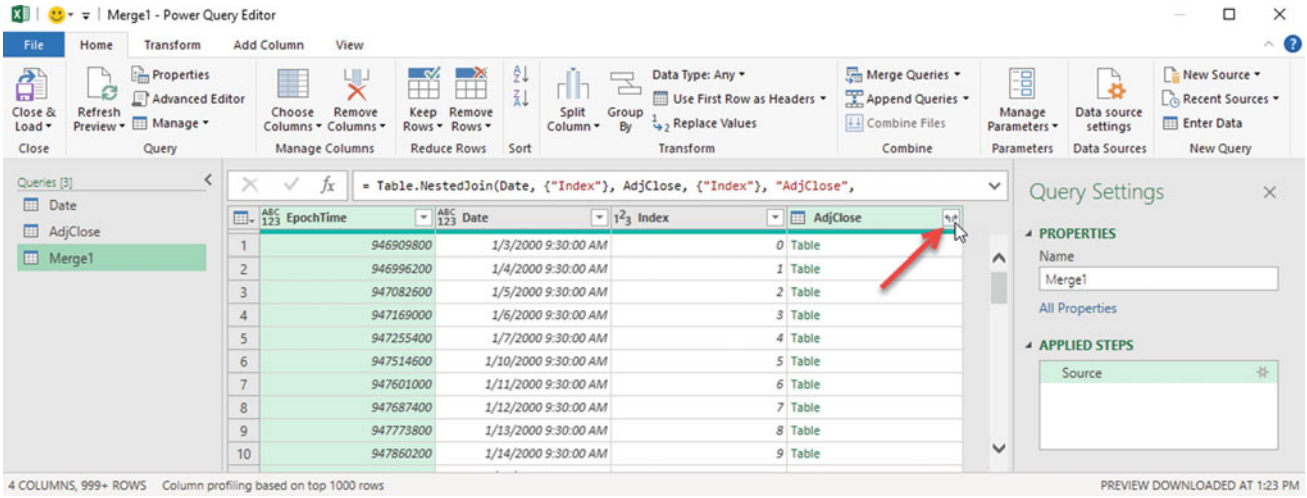
(1) From the *Home* tab, click on *Merge Queries as New*.



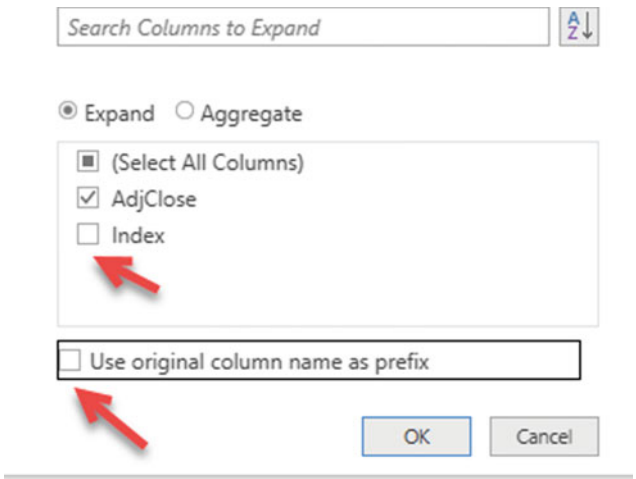
(2) Indicate that the *Date* and *AdjClose* query are to be merged. The *Date* query should be the top query. Select the *Index* column in both queries to indicate that these are the columns used to merge the two queries.



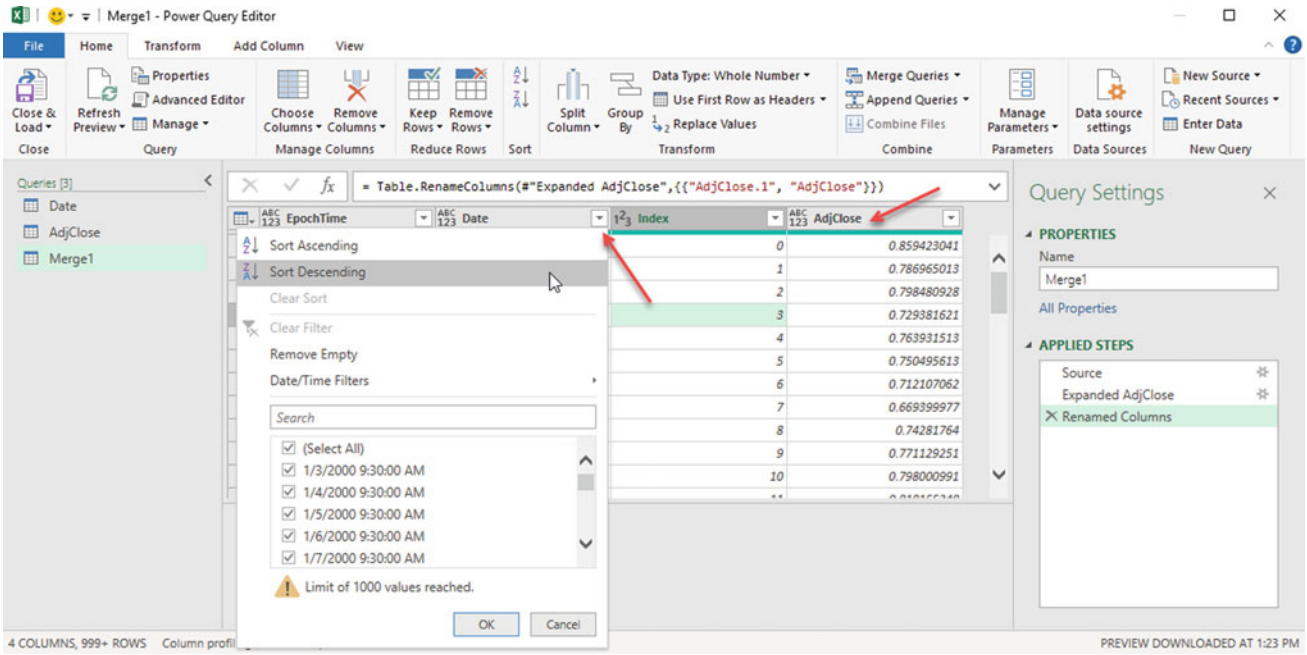
(3) Click on expand column button to expand *AdjClose* column.



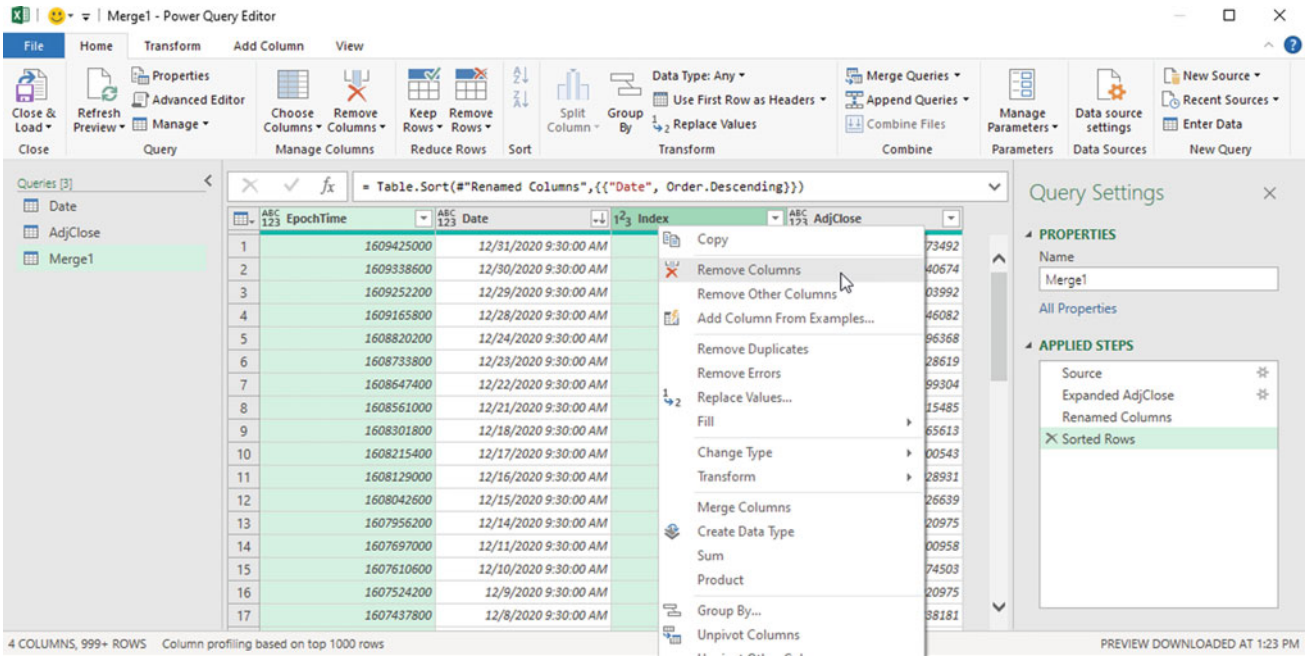
(4) Deselect the *Index* column and deselect the *Use original column name as prefix* and then press the *OK* button.



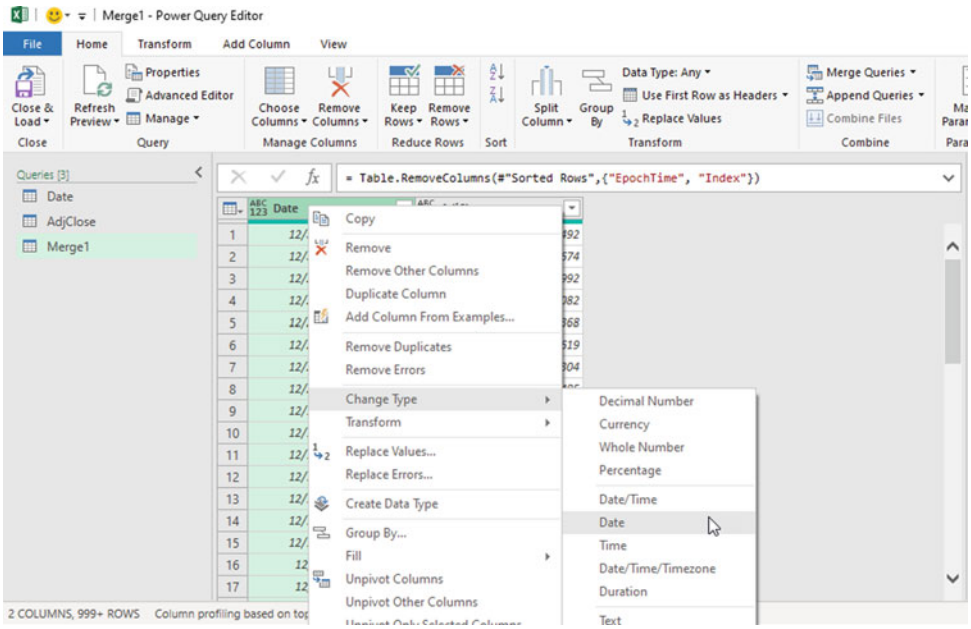
(5) Let's see the most recent date first. To do this, click on the drop-down arrow on the *Index* field and then choose *Sort Descending*. Let's also rename *AdjClose.1* to *AdjClose*.



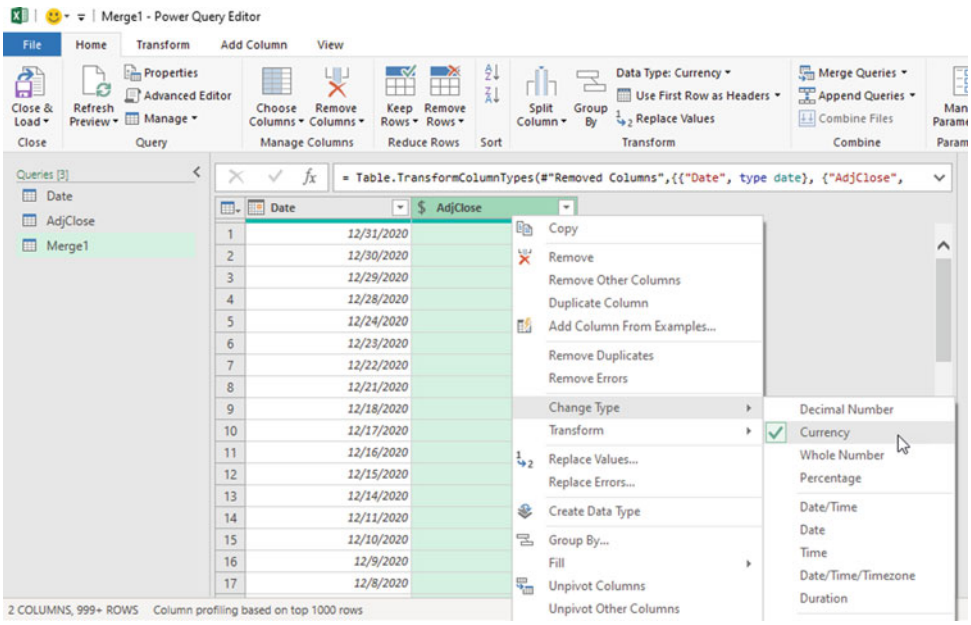
(6) Let's remove the *EpochTime* and *Index* columns by selecting the columns and then right-mouse click on the *Index column* and choose *Remove Columns*.



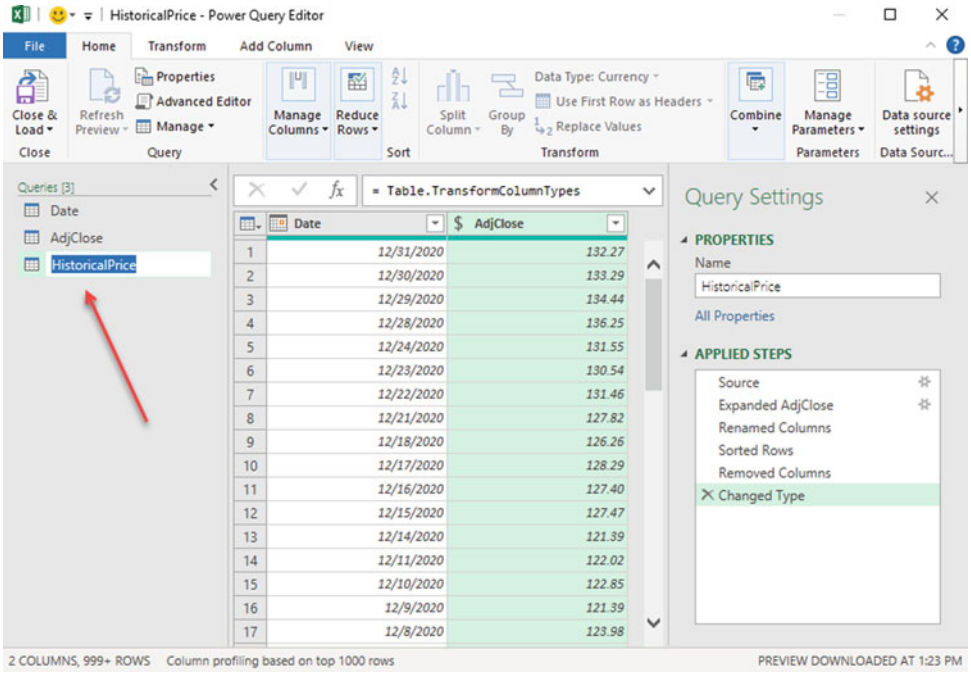
(7) Let's change the *Date* column to *Date* type by right-mouse clicking on the *Date* column and then clicking on the *Date* type as shown below.



(8) Let's change the *Price* column to *Decimal* type by right-mouse clicking on the *Price* column and then clicking on the *Decimal Number* type as shown below.

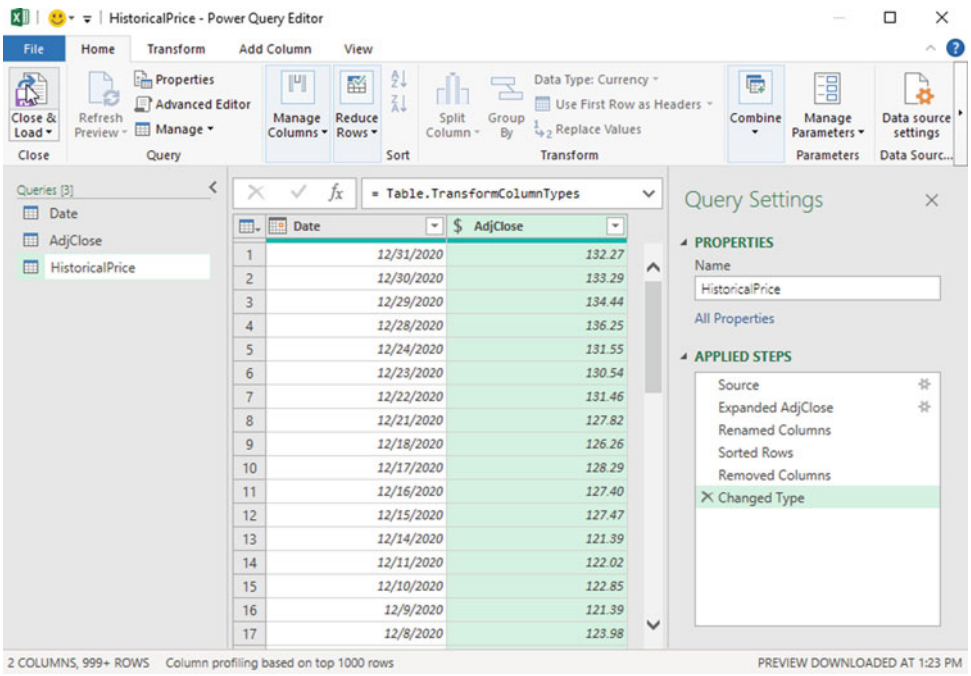


(9) Rename query to *HistoricalPrice*.



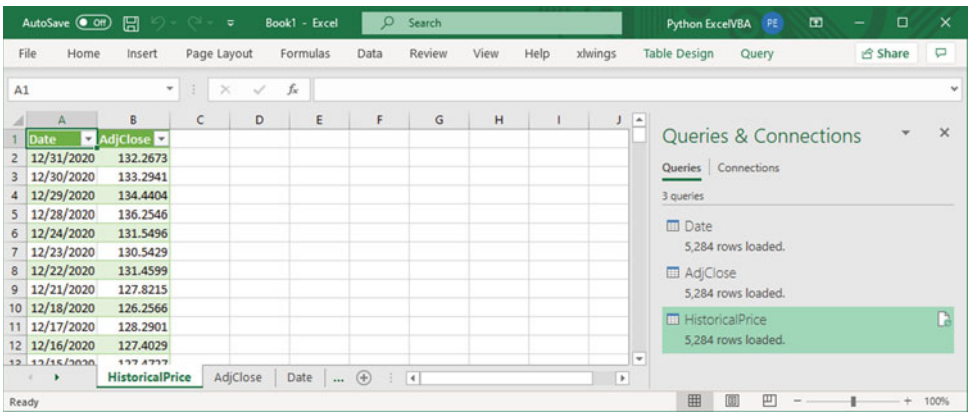
2.7.3.4 Load Data to Microsoft Excel Worksheets

Click on *Close & Load* on the *Home* tab to load data to Excel worksheets.

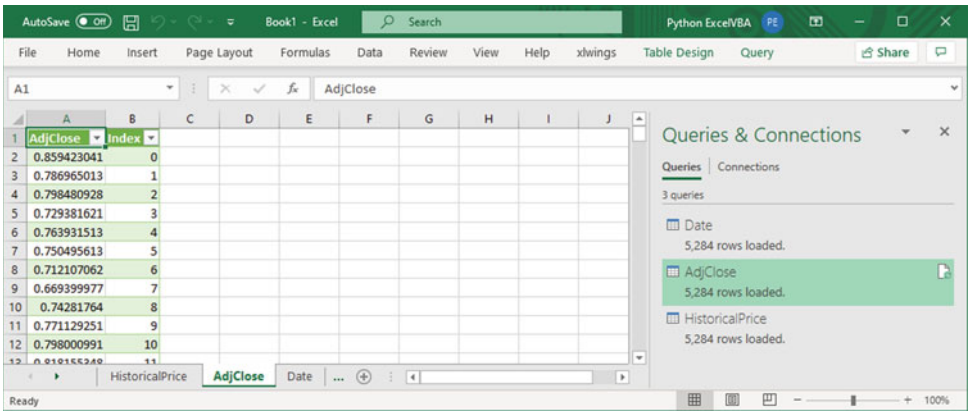


Three Excel worksheets will be populated, one for each query. Excel will first show you the worksheet that has the *HistoricalPrice* query data. The query data is populated into an Excel table. When the Excel table is selected, Excel will

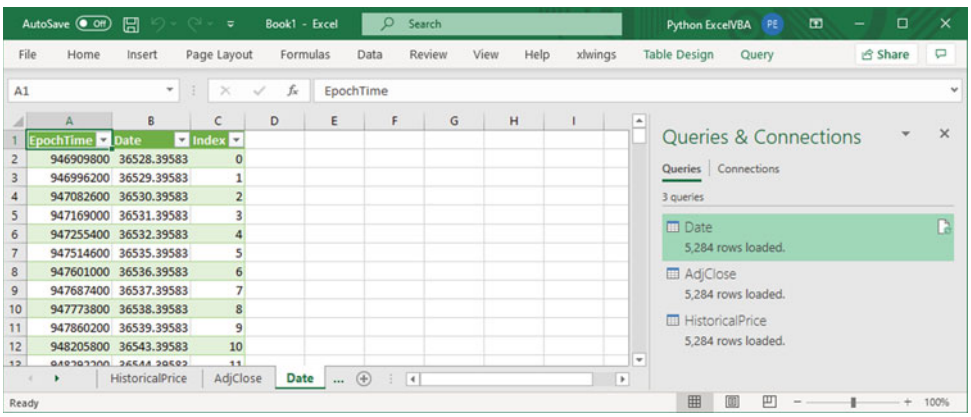
indicate which query is used for that table. Below shows that worksheet *HistoricalPrice* contains the data for the *HistoricalPrice* query.



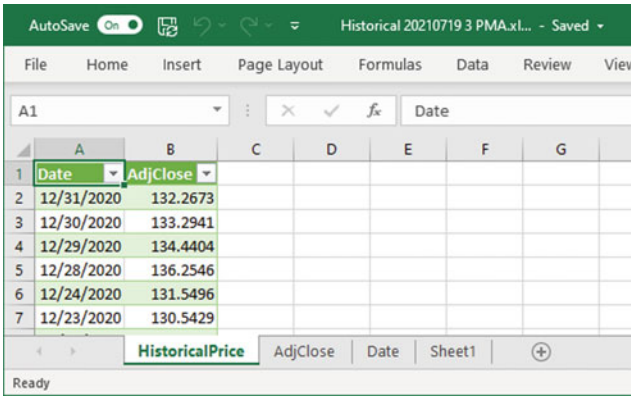
Below shows that worksheet *AdjClose* contains the data for the *Price* query.



Below shows that worksheet *Date* contains the data for the *Date* query.



We can go to the URL https://finance.yahoo.com/quote/AAPL/history to confirm that *Power Query's* price for December 23, 2020, to December 31, 2020, is correct.



As shown above, Yahoo! Finance's AAPL prices agrees with *Power Query* AAPL's price.

2.7.4 Python

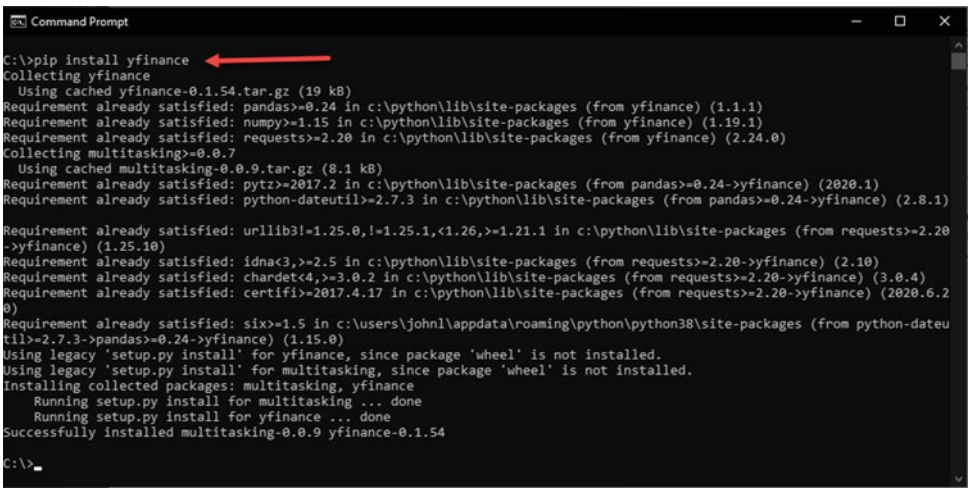
2.7.4.1 Python Libraries

Yahoo! Finance has historically been the go-to place to get financial data. We can see this because many Python libraries scrape Yahoo! Finance for market and financial data. The following are three popular Yahoo! Finance Python libraries:

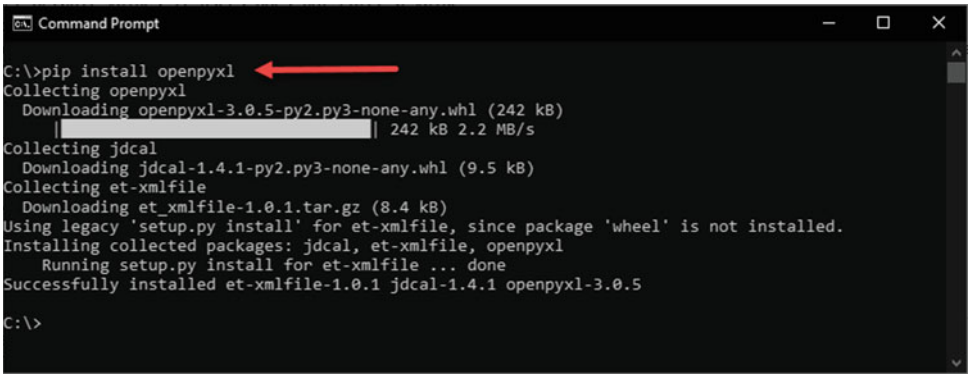
(1) Yahoo! Finance Python library, "Yahoo_Fin." The URL http://theautomatic.net/yahoo_fin-documentation/ has more details on this library.
(2) Yahoo! Finance Python library, "yfinance." The URL https://aroussi.com/post/python-yahoo-finance has more details on this library.
(3) Yahoo! Finance Python library, "yahooquery." The URL https://yahooquery.dpguthrie.com/ has more details on this library.

2.7.4.2 PIP Python Command to Install Libraries

Use the *Python pip* command to install any *Python* library on your computer. Below shows how to install the *yfinance Python* library.



The following *Python pip* command installs the *openpyxl Python* library. This library is necessary to save *Python* data to Microsoft Excel workbooks.



2.7.4.3 Python Code

The following is the *Python* code that was developed in Visual Studio Code and it uses the *yfinance* Python library to retrieve the historical prices of a ticker. The *Python* code has user input to indicate which ticker to retrieve and the date range.

HistoricalPrice_20200829_2PM_C.py

C: > Demo > Historical > HistoricalPrice_20200829_2PM_C.py > ...

```
2   #Python libraries
3   import yfinance # library to refrieve market data from Yahoo! finance
4   import openpyxl # library to save panda dataframes to Microsoft Excel files
5   import os # library used to find path
6   import datetime # library used for date manipulations
7
8   #initialize start date and end date
9   previousday = datetime.date.today()-datetime.timedelta(days=1)# get the date of previous day
10  Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
11  years_ago = datetime.date.today()-datetime.timedelta(days=20 * 365)
12  Default_start = years_ago.strftime("%Y-%m-%d")
13
14  # request user input
15  # set default values if users does not enter a value
16  Ticker = input('Please enter ticker(Default = AAPL)-->') or 'AAPL'
17  StartDate = input('Enter start date(' + 'Default = 20 Years ago, ' + Default_start + ')-->') or Default_start
18  EndDate = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
19  Time_interval = input('Enter date interval(1d,1wk,1mo Default = 1d)-->') or '1d'
20  print("")
21
22  #display long name for ticker
23  ticker = yfinance.Ticker(Ticker)
24  Ticker_info = ticker.info
25  print ("Downloading historical prices for " + Ticker_info['longName'] )
26
27  dir_path = os.path.dirname(os.path.realpath(__file__))
28  ExcelFileName = Ticker + '_' + Time_interval +'_'+ StartDate + '_to_' + EndDate + '.xlsx'
29
30  #prep historical data
31  tickerhistory=yfinance.download(Ticker,start=StartDate,end=EndDate,interval = Time_interval)
32  tickerhistory = tickerhistory.sort_index(ascending=0) # show most current date first
33  tickerhistory = tickerhistory.dropna()
34  tickerhistory["ticker"] = Ticker
35  tickerhistory.to_excel(ExcelFileName) #save historical data to Excel file
36
37  print("")
38  print(str( "{:,}".format(len(tickerhistory))) + " historical prices retrieved.")
39  print (ExcelFileName )
40  print('Saved to ' + dir_path)
41  print("")
```

2.7.4.4 Python Output

```
Python 3.8.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Demo\Historical\HistoricalPrice_20200829_2PM_C.py ========
Please enter ticker(Default = AAPL)-->AAPL
Enter start date(Default = 20 Years ago, 2000-09-03)-->2000-01-01
Enter end date(Default = Yesterday, 2020-08-28)-->2020-08-20
Enter date interval(1d,1wk,1mo Default = 1d)-->1d

Downloading historical prices for Apple Inc.
[**********************100%**********************]  1 of 1 completed

5,191 historical prices retrieved.
AAPL_1d_2000-01-01_to_2020-08-20.xlsx
Saved to C:\Demo\Historical
```

Above is the *Python* output from the *Python* code shown above. Note that to get AAPL's August 19, 2020, price, we had to enter August 20, 2020, as the end date. The historical files are saved as a Microsoft Excel file in the same folder as the *Python* code.

Opening the Microsoft Excel file *AAPL_1d_2000-01-01_to_2020-08-19.xlsx* we can see that the historical price for AAPL for August 19, 2020, and August 18, 2020.

AutoSave Off — AAPL_1d_2000-01-01_to_2020-08-20.xlsx

File Home Insert Draw Page Layout Formulas Data Review View Developer

F2 462.829986572265

	A	B	C	D	E	F	G	H	I
1	Date	Open	High	Low	Close	Adj Close	Volume	ticker	
2	2020-08-19 00:00:00	463.93	468.65	462.44	462.83	462.83	36384500	AAPL	
3	2020-08-18 00:00:00	457.41	464	456.03	462.25	462.25	26408400	AAPL	
4	2020-08-17 00:00:00	464.25	464.35	455.85	458.43	458.43	29890400	AAPL	
5	2020-08-14 00:00:00	459.32	460	452.18	459.63	459.63	41391300	AAPL	
6	2020-08-13 00:00:00	457.72	464.17	455.71	460.04	460.04	52520500	AAPL	
7	2020-08-12 00:00:00	441.99	453.1	441.19	452.04	452.04	41486200	AAPL	
8	2020-08-11 00:00:00	447.88	449.93	436.43	437.5	437.5	46975600	AAPL	

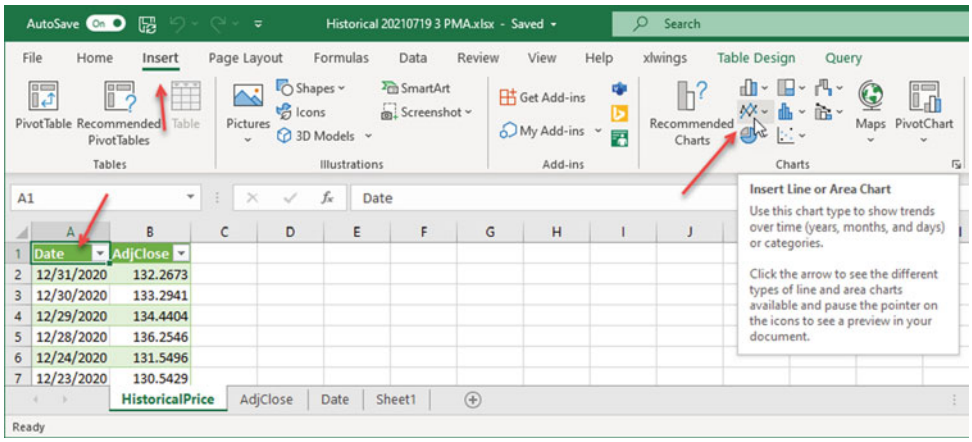
Sheet1

2.8 Charting Historical Data

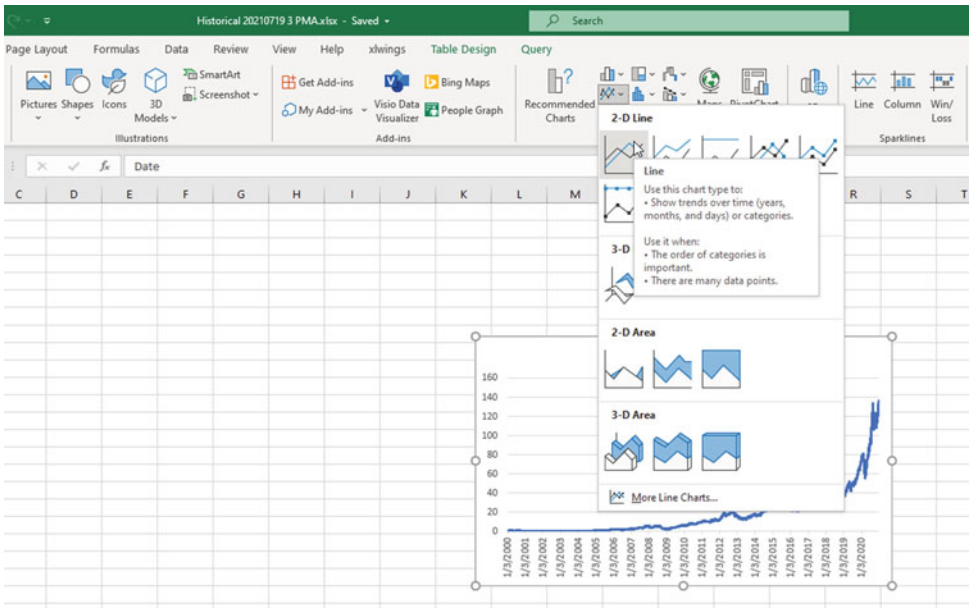
2.8.1 Microsoft Excel 365 Chart Wizard

We will graph AAPL's stock historical stock prices by using Excel's *chart wizard*. The *chart wizard* will allow us to create a time-series graph of AAPL's historical stocks quickly. We will use AAPL's historical data set shown above.

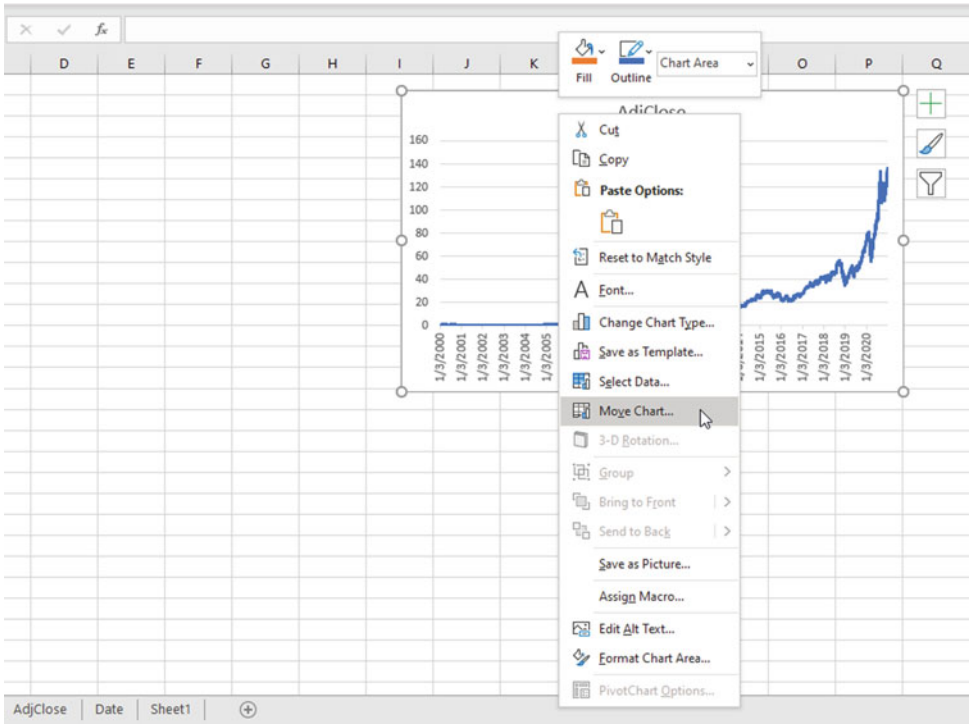
(1) While cell "a1" is selected, click on the *Insert Line or Area Chart* in the *Insert Tab*.



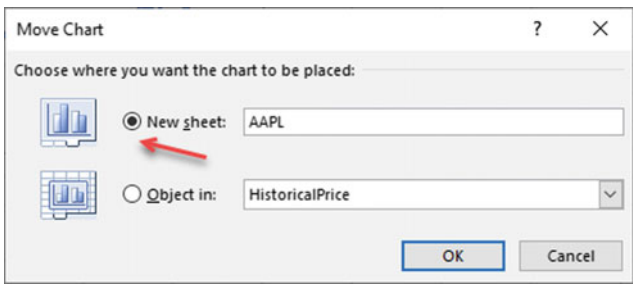
(2) Click on the first *2-D* line option. Excel will automatically create a line graph.



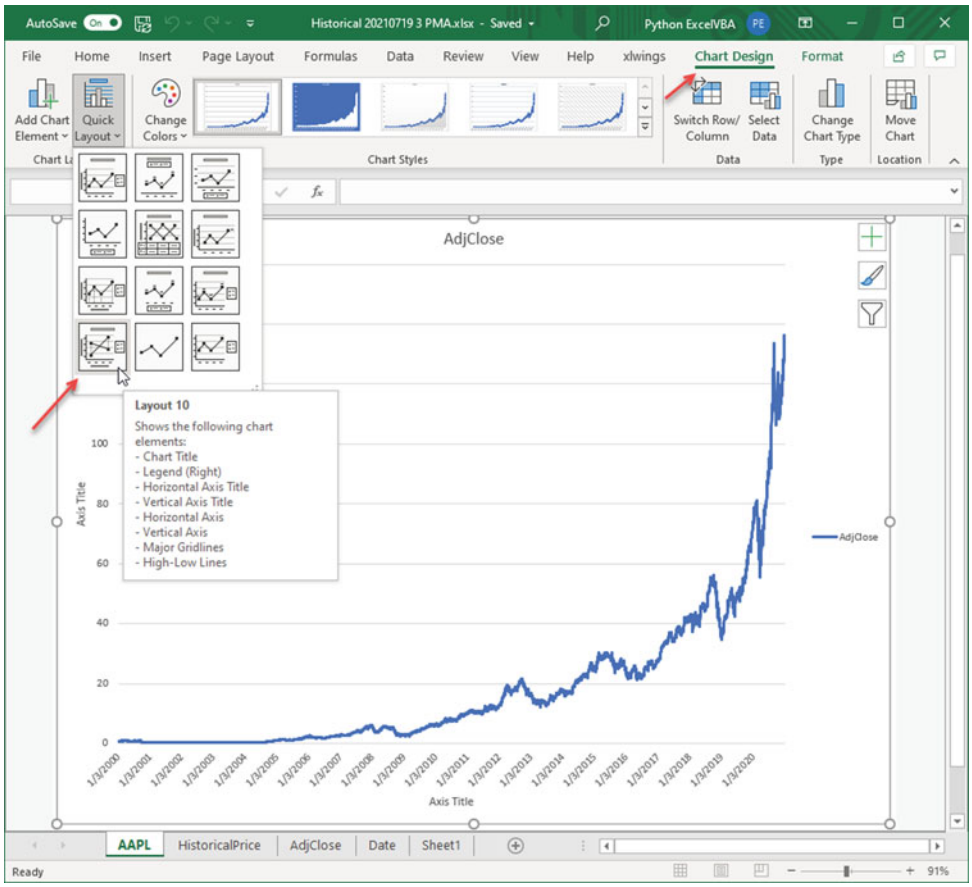
(3) Right-mouse click on the chart and click on *Move Chart*.

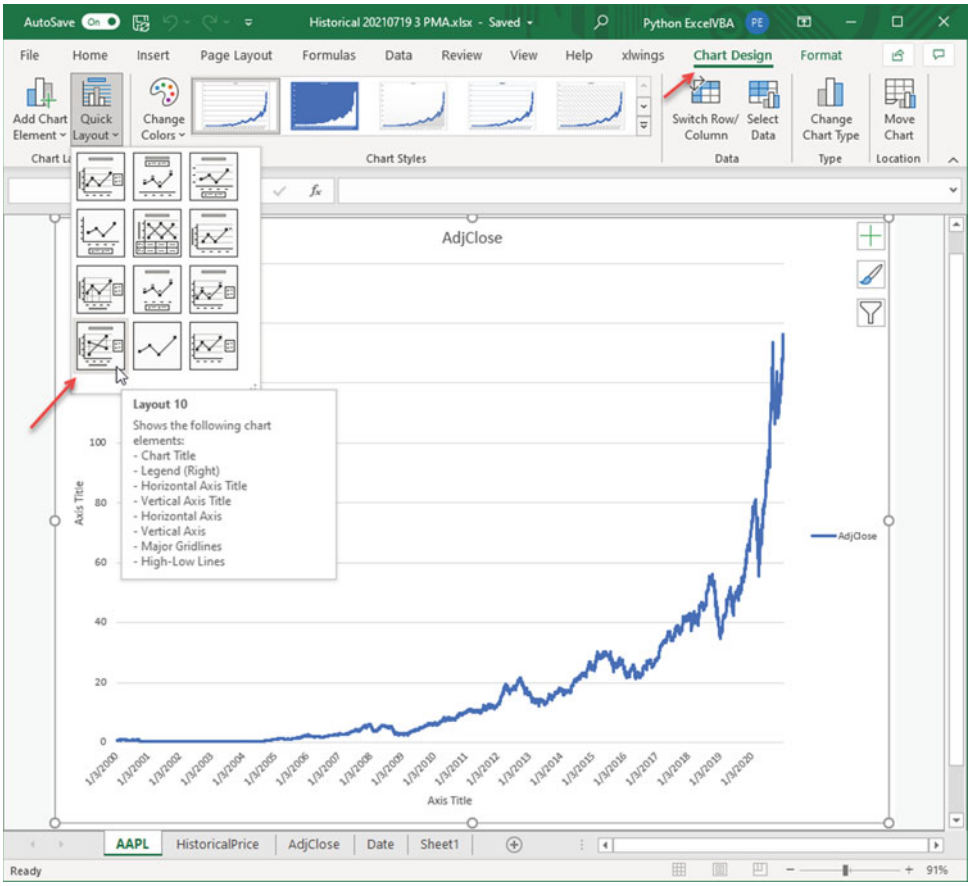


(4) Choose *New sheet* and type in AAPL.

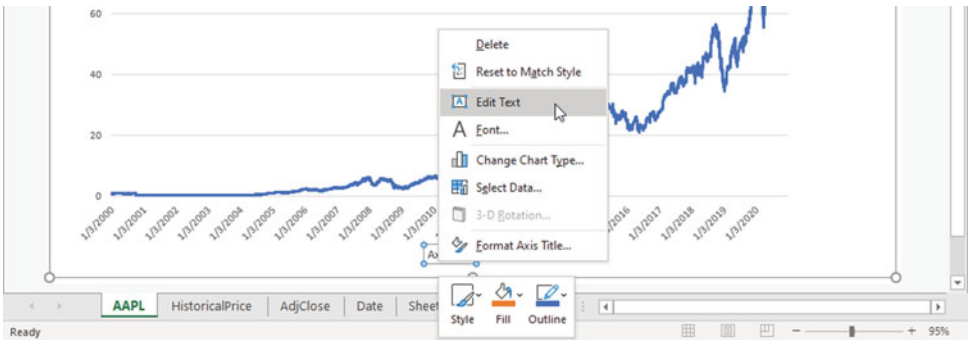


(5) From the *Chart Design* tab choose *Layout 10*.

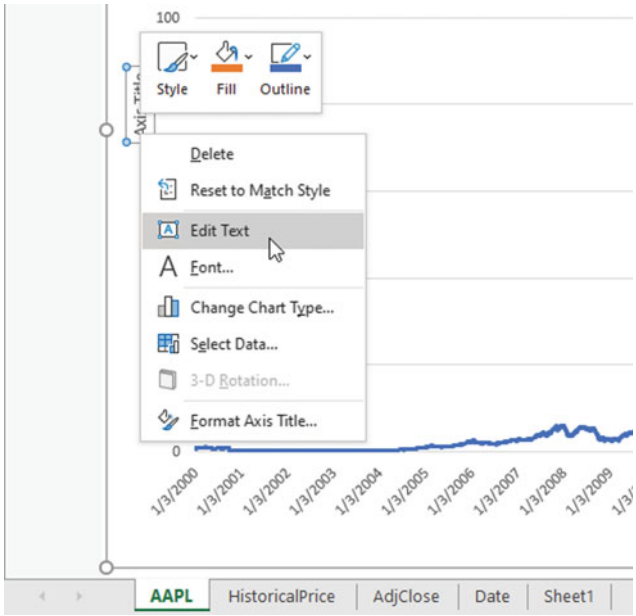




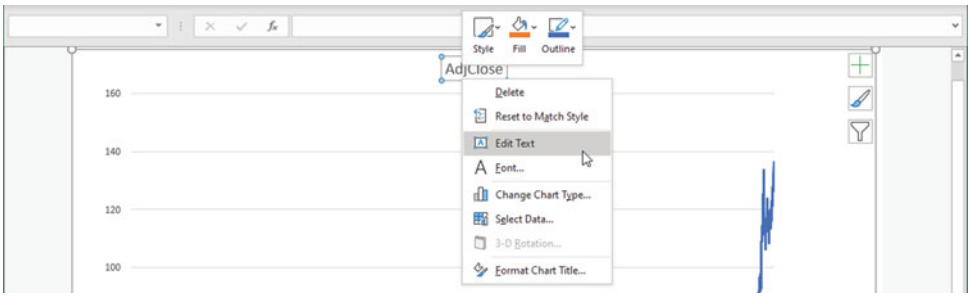
(6) Choose *Edit Text.* Change text to "Date."



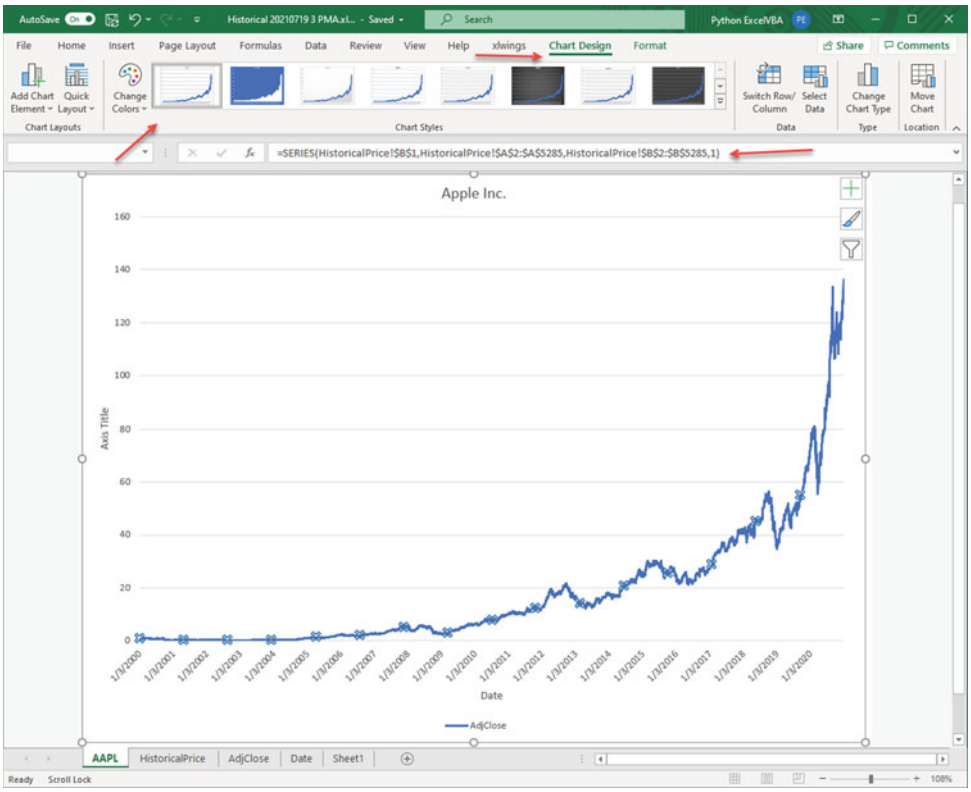
(7) Choose *Edit Text*. Change text to "Price."



(8) Choose *Edit Text*. Change text to "Apple Inc."



Below is the time-series graph of *Apple Inc.*

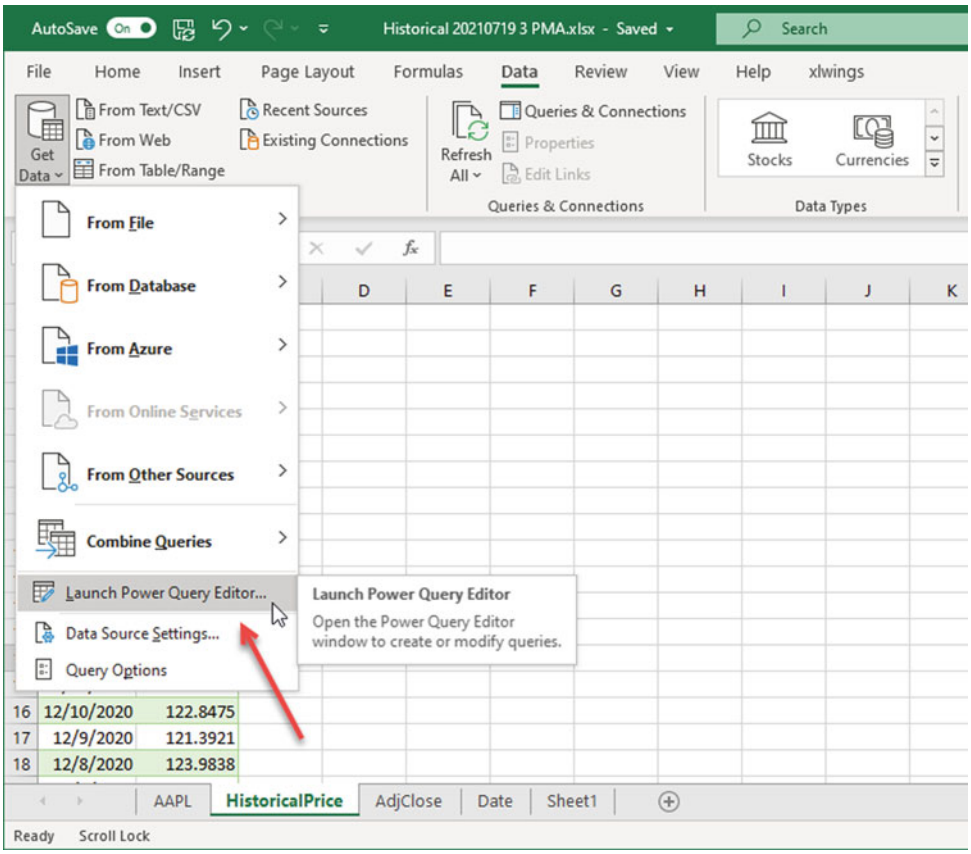


Notice that the formula bar indicates that the data source of the graph is from the *HistoricalPrice* worksheet.

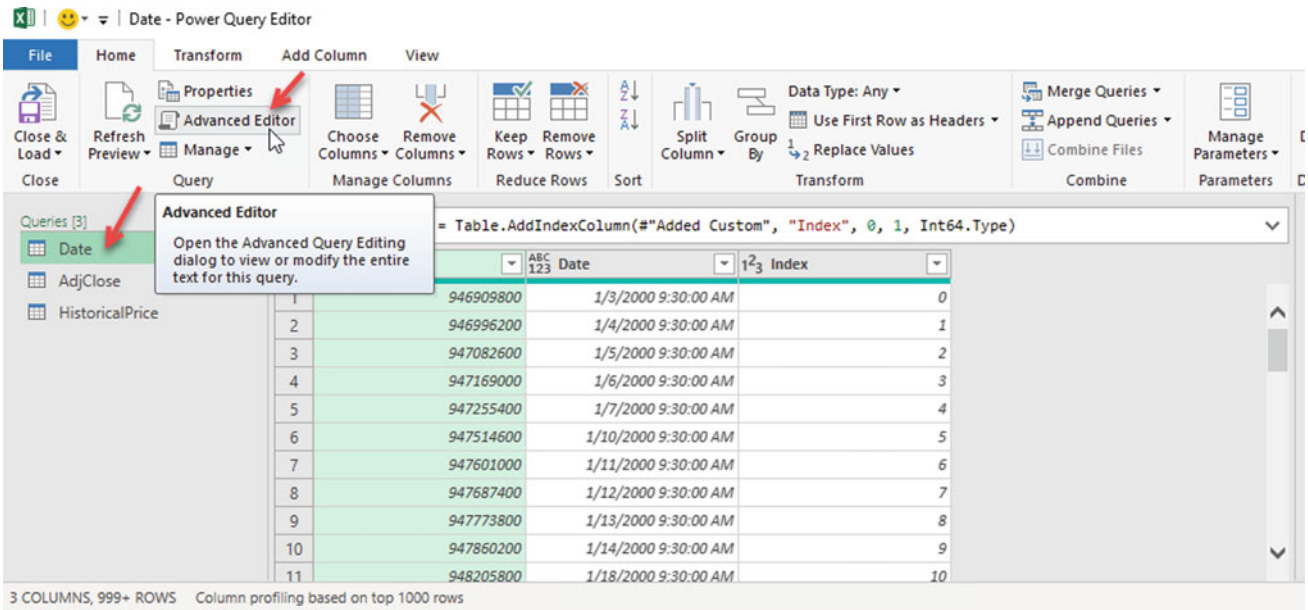
2.8.2 Power Query M Code

When we performed the *Power Query* manual steps above, *Power Query* recorded those steps in a language called *M Code*. In this section, we will adjust the *M Code* to retrieve Amazon.com, Inc's historical stock prices. Amazon.com, Inc's ticker symbol is AMZN.

(1) From the *Data* tab select Get Data → Launch Power Query Editor.

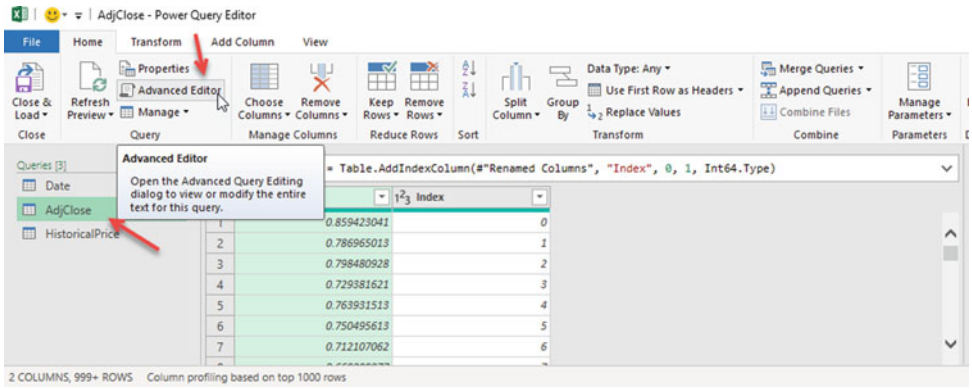


(2) To edit the *M Code* for the *Date* query, select the *Date* query and then click on *Advanced Editor*.



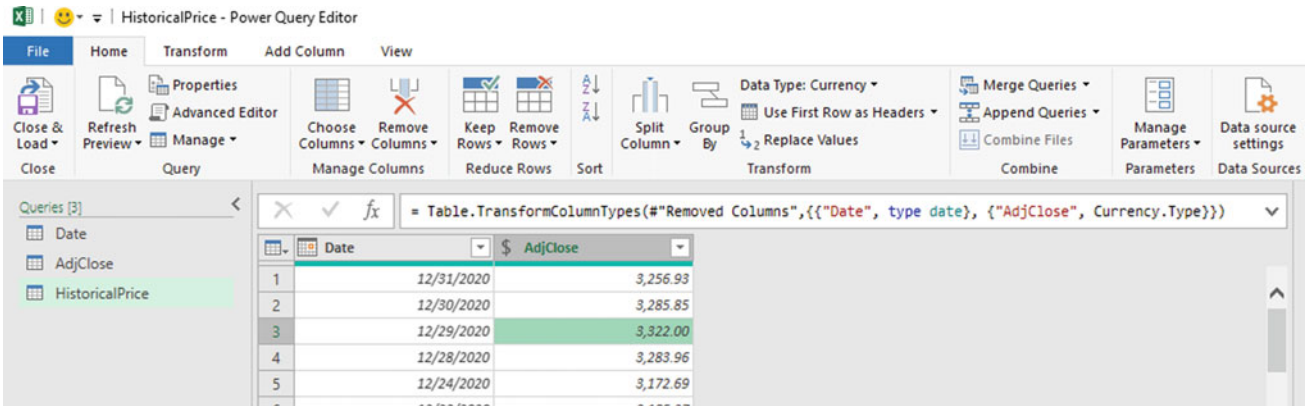
(3) In line 2, change the *AAPL* ticker to *TSLA*.

(4) To edit the *M Code* for the *Date* query, select the *AdjClose* query and click on *Advanced Editor*.



(5) In line 2, change the *AAPL* ticker to *TSLA*.

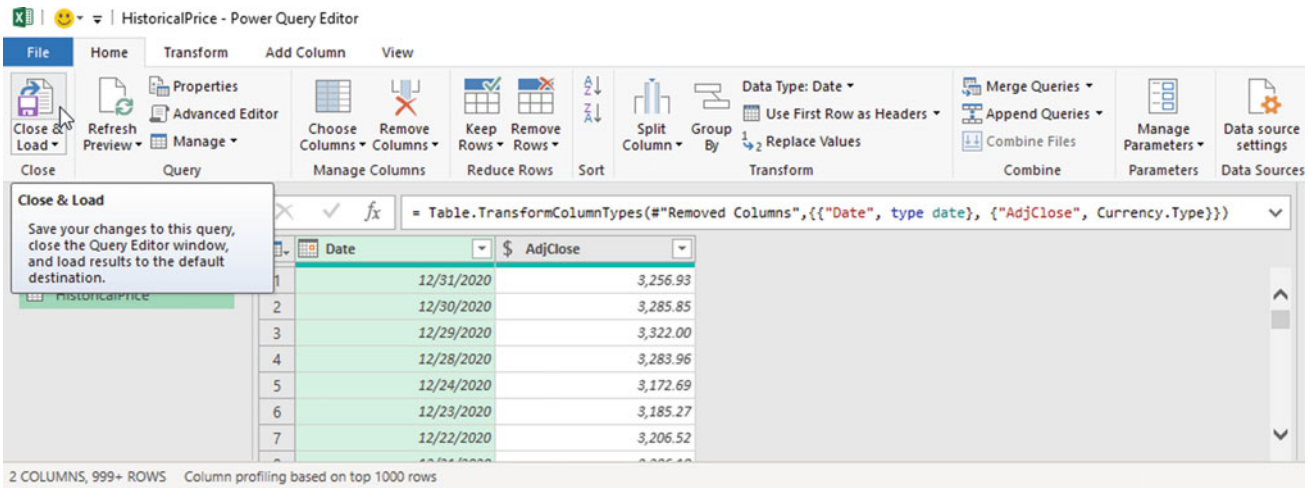
(6) Select the *HistoricalPrice* query. Notice that AMZN's price is automatically updated. It was unnecessary to update the *HistoricalPrice* query because this query relies on the *Date and AdjClose* queries.



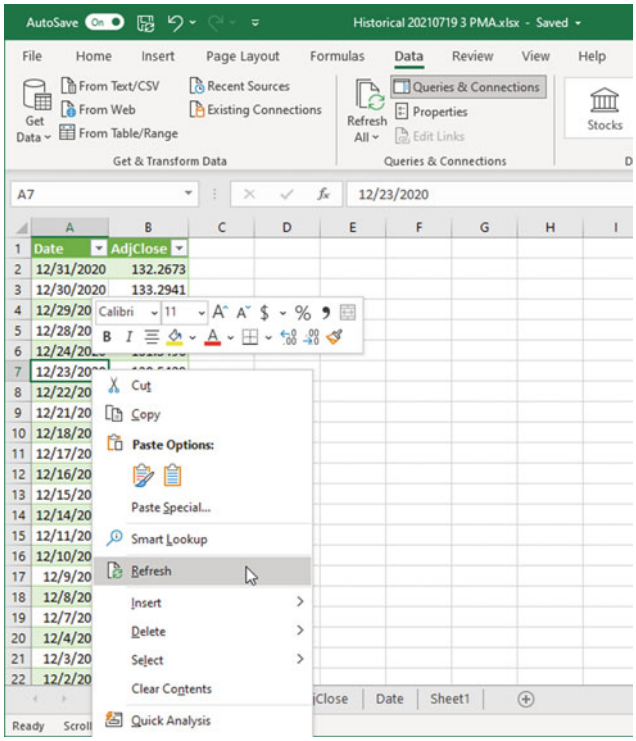
We can confirm that TSLA prices in this *HistoricalPrice* are correct by going to https://finance.yahoo.com/quote/AMZN/history.

This example demonstrates that we can use AAPL's *Power Query* setup as a template to retrieve historical prices for other tickers.

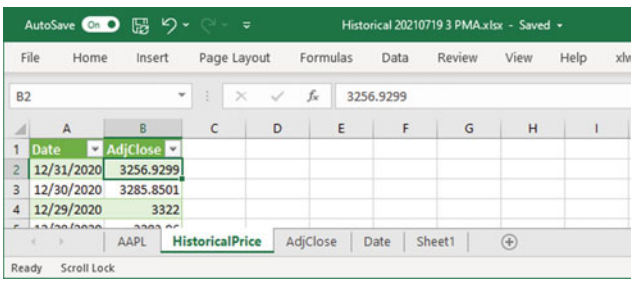
(7) Click on Load.



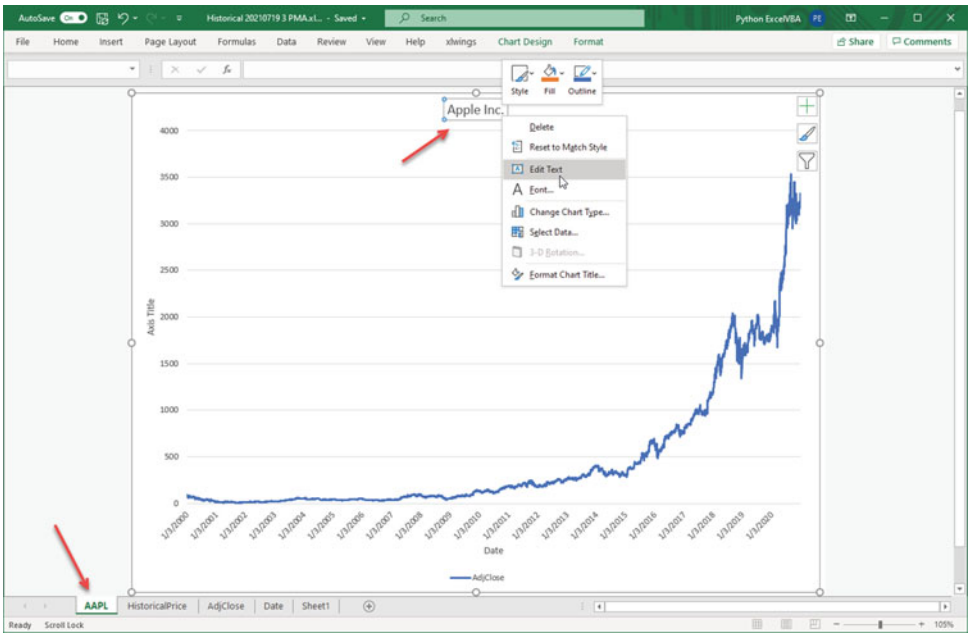
(8) Select cell "A7" in *HistorialPrice* and then right-mouse click on click on *Refresh*. Noticed that AAPL's December 2020 price is still showing before refreshing.



After the refresh, cell "B2" is showing AMZN price for December 31, 2020.



(9) Going to the *AAPL* chart, we can see that it is now graphing *AMZN* historical prices. Let's change the chart title by right-mouse clicking on the chart title and choose *Edit Text*. Enter Amazon.com Inc for the chart title. Change the chart tab at the bottom to AMZN.



2.9 Using Python to Graph Johnson & Johnson's Historical Prices

The following Python code, developed in Google Colabroartory, will graph John & Johnson's 5-year historical prices:

```
[23] pip install yfinance
```

```
[24] pip install requests
```

```
[25] pip install bs4
```

```
[26] #Last Update:Apr-10-2021
     import yfinance as yf
     import matplotlib.pyplot as plt
     import numpy as np
     import datetime as dt#get current date
     import pandas as pd
     import requests
     from bs4 import BeautifulSoup
     from google.colab import files
```

```
[27] #url: url that beautifulsoup will analyze
     #beautifulsoup html will not look like soure
     #review beautifulsoup html to use find and findall
     def html_beautifulsoup(url):
         response = requests.get(url)
         html = response.content
         soup = BeautifulSoup(html,"lxml")
         return soup

     #soup_html: soup html is not same as source html
     def yahoo_description(soup_html):
         #element that has ticker description
         element = soup_html.find('h1',class_='D(ib) Fz(18px)')
         return element.text
```

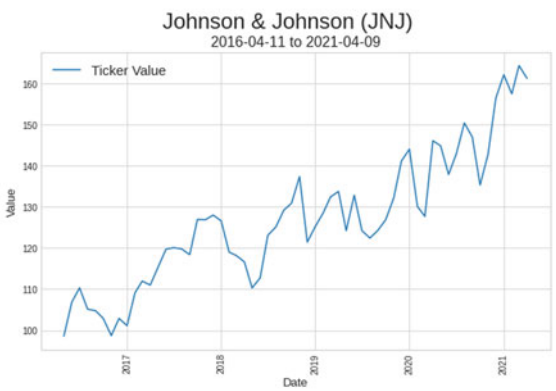
```
[41] #graph historical values of ticker
     def graph_ticker(Ticker, time_interval, start_t, end_t):
       Data=yf.download(Ticker,start=start_t,end=end_t,interval = time_interval)
       Adjclose=Data['Adj Close']
       Dr = Adjclose.dropna()
       soup_html = html_beautifulsoup("https://finance.yahoo.com/quote/"  + Ticker)

      #graph ticker
       plt.figure(figsize=(10,6))
       plt.style.use('seaborn-whitegrid')#set the background to be grid
       plt.xticks(rotation=90)# rotate the ylim so that it does not look too tight
       plt.title(start_t+' to '+end_t,fontsize=16)
       plt.suptitle(yahoo_description(soup_html),fontsize=24)

       plt.plot(Dr)

       plt.legend(labels =['Ticker Value'],fontsize=15,  loc=0)
       plt.ylabel('Value',fontsize=13)
       plt.xlabel('Date',fontsize = 13)

       print("")
       print("")

       plt.show()

       #print data set
       Dr = pd.DataFrame(Dr)
       Dr = Dr.sort_values(by='Date', ascending=False)
       pd.set_option('display.max_rows', None)
       Dr = Dr['Adj Close'].round(decimals=2) # rounding results in a series
       Dr = pd.DataFrame(Dr) # convert to dataframe to print header
       print(Dr)

       #Download data as CSV
       Dr.to_csv('flowchart.csv')
       files.download('flowchart.csv')
```

```
[42] #initialize dates
     previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
     Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
     years_ago = dt.date.today()-dt.timedelta(days=5 * 365)
     Default_start = years_ago.strftime("%Y-%m-%d")

     # request user input
     # set default values if users does not enter a value
     Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
     Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
     End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
     T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'
```

```
Please enter ticker(Default = JNJ)-->JNJ
Enter start date(Default = 5 Years ago, 2016-04-11)-->2016-04-11
Enter end date(Default = Yesterday, 2021-04-09)-->2021-04-09
Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo
```

```
print("")
graph_ticker(Ticker, T_interval,Start,End)
print("")
```

```
[**********100%**********]  1 of 1 completed
```



Date	Adj Close
2021-04-01	161.25
2021-03-01	164.35
2021-02-01	157.48
2021-01-01	162.12
2020-12-01	156.40
2020-11-01	142.79
2020-10-01	135.32
2020-09-01	146.94
2020-08-01	150.41
2020-07-01	142.91
2020-06-01	137.88
2020-05-01	144.83
2020-04-01	146.09
2020-03-01	127.68
2020-02-01	130.11
2020-01-01	144.03
2019-12-01	141.13
2019-11-01	132.11
2019-10-01	126.87
2019-09-01	124.31
2019-08-01	122.42
2019-07-01	124.19
2019-06-01	132.83
2019-05-01	124.23
2019-04-01	133.75
2019-03-01	132.41
2019-02-01	128.57
2019-01-01	125.22
2018-12-01	121.43
2018-11-01	137.35
2018-10-01	130.89
2018-09-01	129.19
2018-08-01	125.10
2018-07-01	123.09
2018-06-01	112.70
2018-05-01	110.29
2018-04-01	116.62
2018-03-01	118.15
2018-02-01	118.99
2018-01-01	126.60
2017-12-01	128.00
2017-11-01	126.87
2017-10-01	126.94
2017-09-01	118.38
2017-08-01	119.77
2017-07-01	120.08
2017-06-01	119.70
2017-05-01	115.28
2017-04-01	110.98
2017-03-01	111.95
2017-02-01	109.12
2017-01-01	101.12
2016-12-01	102.87
2016-11-01	98.70
2016-10-01	102.86
2016-09-01	104.76
2016-08-01	105.13
2016-07-01	110.32
2016-06-01	106.86

2.10 Summary

Statistics is mainly associated with data and numbers. A complete statistical analysis of a data set consists of both numbers and graphs. In this chapter, we showed how to graph a data set in both Excel and Python.

One aspect of statistical analysis not often discussed is the difficult process of retrieving data for statistical analysis. Excel, Python, and the Internet have made the process of retrieving data for statistical data analysis easier. We discussed these techniques and concepts on how to automate the retrieval of market data. In Excel, discussed on to use *Power Query* to retrieve market data from Yahoo! Finance and Quandl. In Python, we demonstrated how to retrieve market data using Python libraries.

This chapter published and demonstrated the Python code on retrieving and graphing the 5-year history from Yahoo! Finance.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O’Reilly Media, 2017.

3 Histograms, Rate of Returns, and Financial Statements

3.1 Introduction

In this chapter, we are interested in the dispersion and the general location of ticker's monthly returns. We will use *histograms* to analyze the dispersion and location of a ticker's monthly stock price returns. We will also look at how to automate the retrieval of ticker's financial statements.

In Sect. 3.2 of this chapter, we discuss rate of return. We then talk about histograms in Sect. 3.3. In Sect. 3.4, we use Python to create Johnson & Johnson's rate of return histogram. In Sect. 3.5, we look at financial statements, and finally, in Sect. 3.6, we summarize the chapter.

3.2 Rate of Return

Suppose P1 is the price of a stock on the last business day of the month year and that P2 is the price of the stock on the last business day of the current month. Then, the monthly rate of return will be calculated as follows:

$$\text{Monthly rate of return} = (\text{P2} - \text{P1})/\text{P1}.$$

The above monthly rate of returns calculation ignores the effects of dividends.

3.2.1 Power Query

This section will demonstrate how *Power Query* can calculate the monthly rate of return for AAPL. Later on, we will extend it to any stock ticker. Our starting point will be the historical *Power Query* that we created in Chap. 2.

We will use the concepts described in the following *ExcelisFun* video by Mike Girvin, https://www.youtube.com/watch?v=sOsw_d1-y8w&t=24s. This video talks about how to calculate the price change of a stock from the prior period. In this section, we extend it by discussing how to calculate the rate of return of a stock.

In the previous chapter, we retrieved Apple Inc.'s historical daily prices from January 01, 2000 to January 01, 2021. In this chapter, instead of retrieving the daily prices, we will retrieve the monthly prices. The URL to retrieve the monthly prices is the same as the daily prices; the only difference is we will change the interval from "1d" to "1mo." The following is the URL to retrieve the monthly prices:

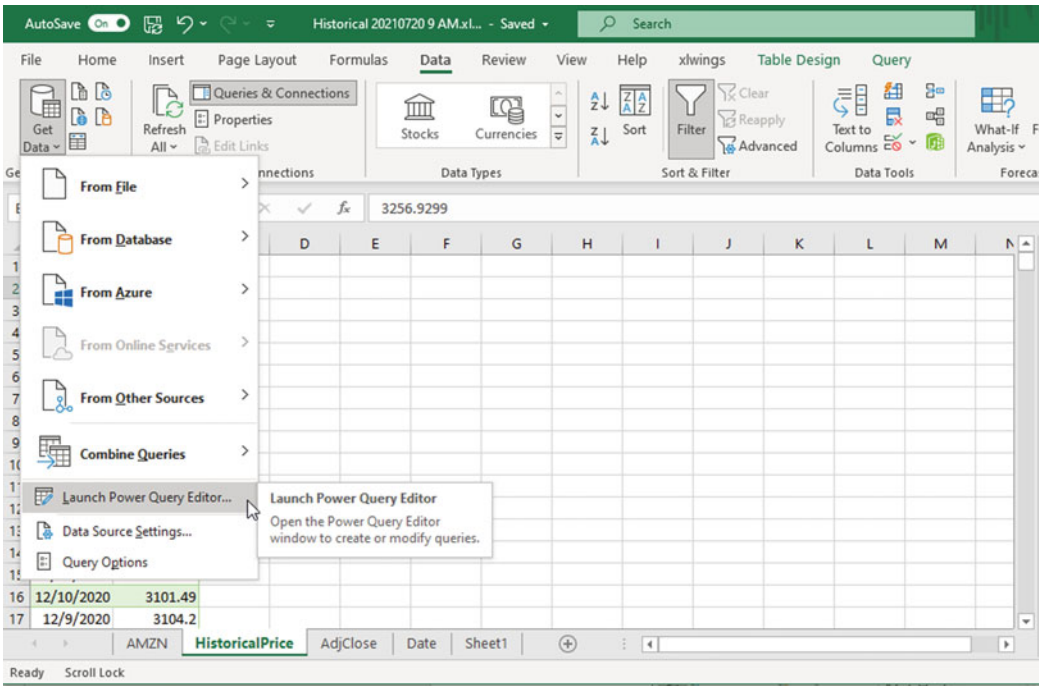
https://query1.finance.yahoo.com/v8/finance/chart/AAPL?symbol=AAPL&period1=946702800&period2=1609459200&interval=1mo.

3.2.1.1 Monthly Historical Prices

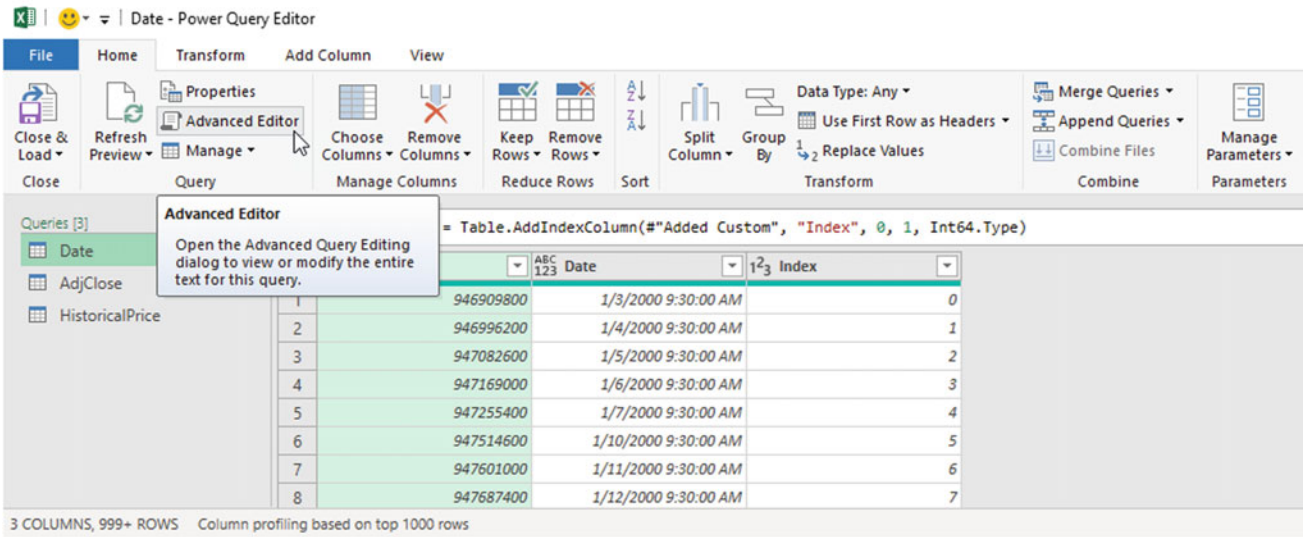
The following are the steps to modify AAPL's daily historical prices in Chap. 2 to monthly historical prices.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_3

(1) From the *Data* tab, click on *Get Data* → *Launch Power Query Editor*.

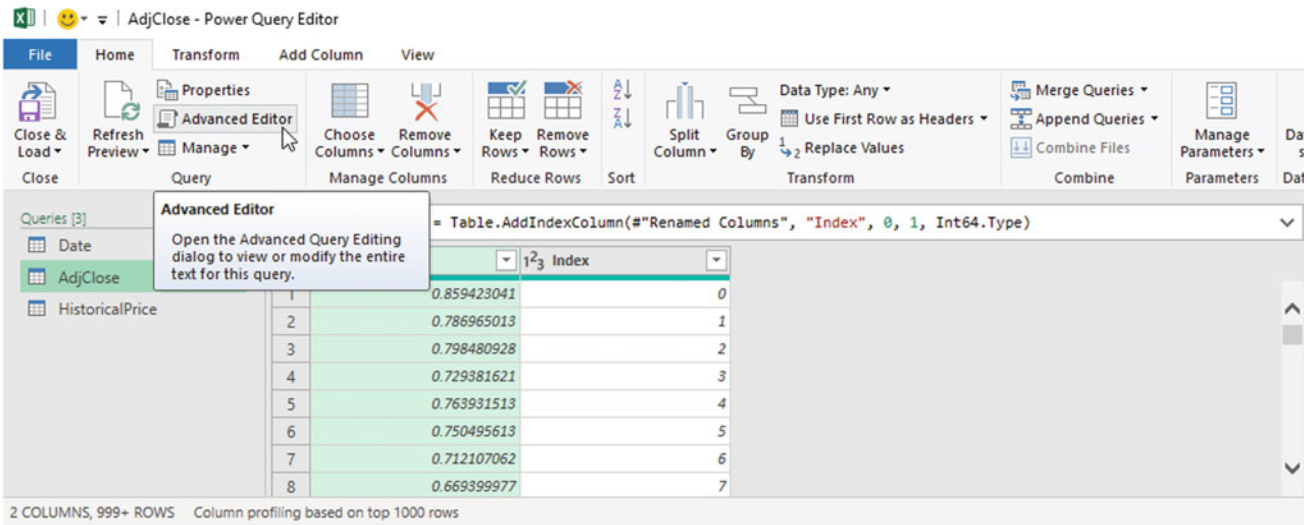


(2) With the *Date* query selected, click on *Advanced Editor*.



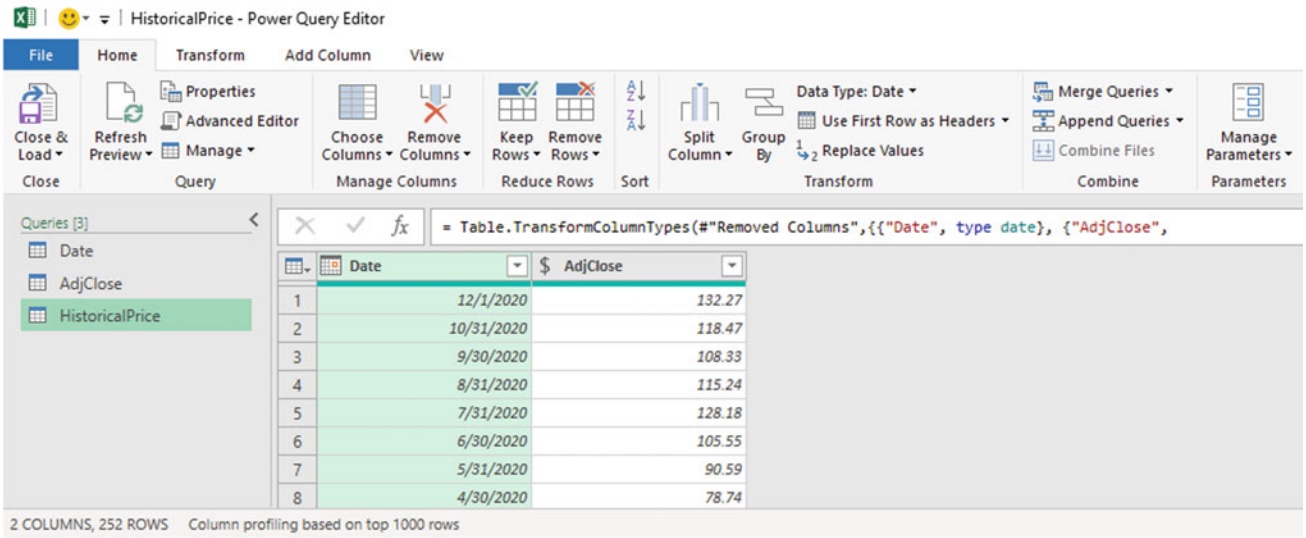
(3) In line 2, change the data interval from "1d" to "1mo."

(4) With the *AdjClose* query selected, click on *Advanced Editor.*



(5) In line 2, change the data interval from "1d" to "1mo."

(6) Selecting the *HistoricalPrice* query shows that the historical data is now monthly.

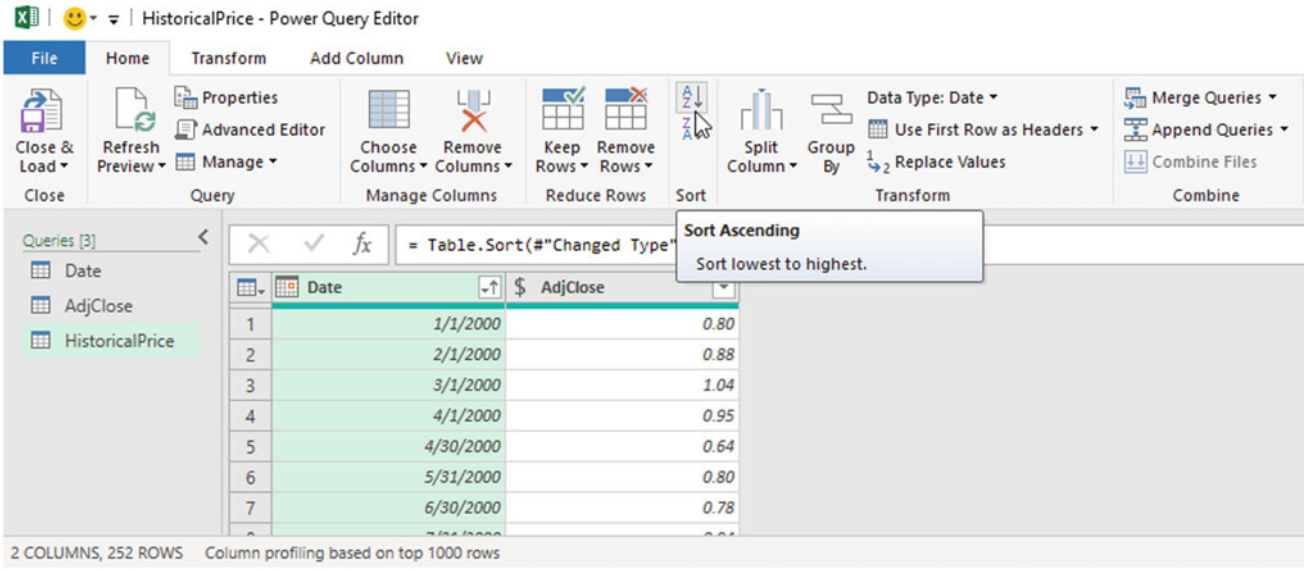


3.2.1.2 Ascending Date Order

The AAPL's historical *Power Query* that we created in Chap. 2 had the dates in descending order. To create the rate of return calculations, we need the dates to be in ascending order.

The following are the steps to sort the date data in ascending order:

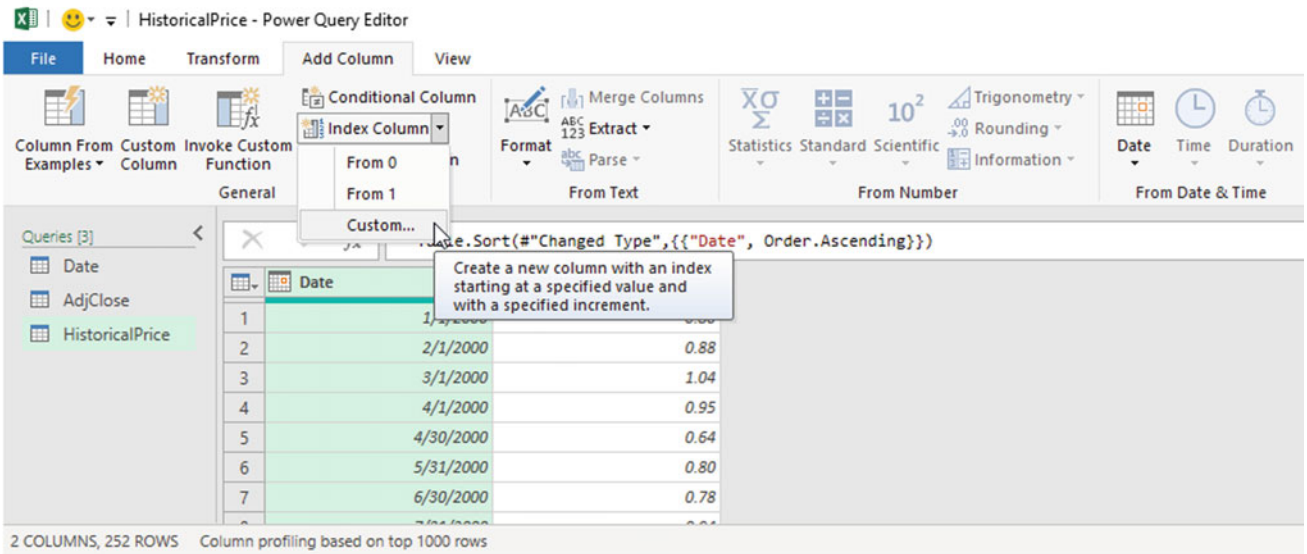
(1) Select the *Date* field and click on the *Sort Ascending* icon.



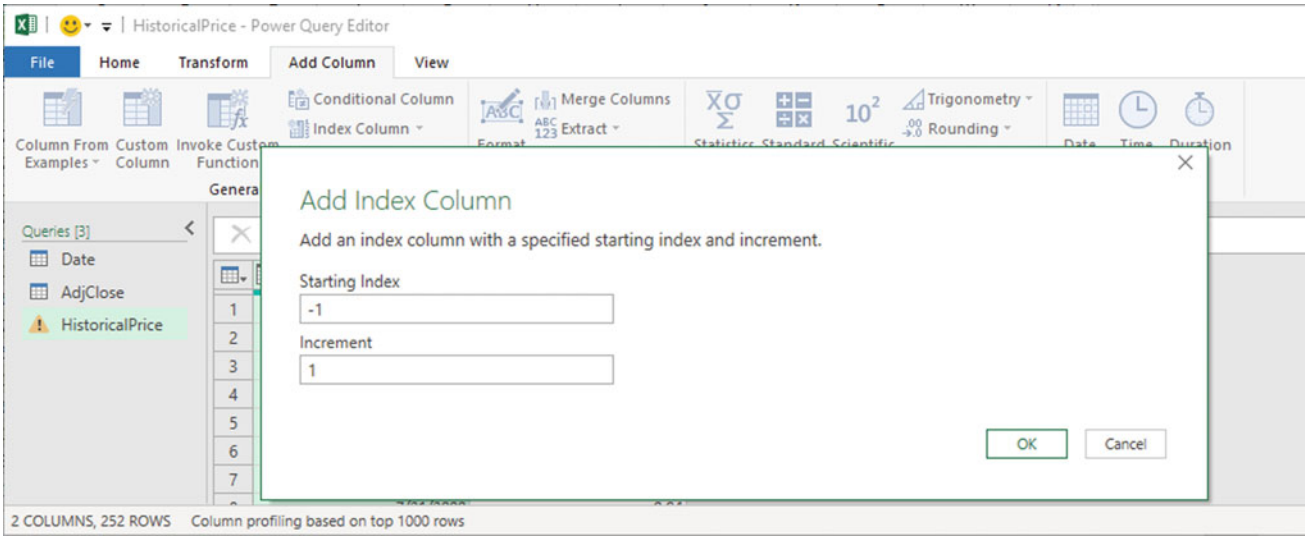
3.2.1.3 Adding Index Columns

The following demonstrates how to add an index column:

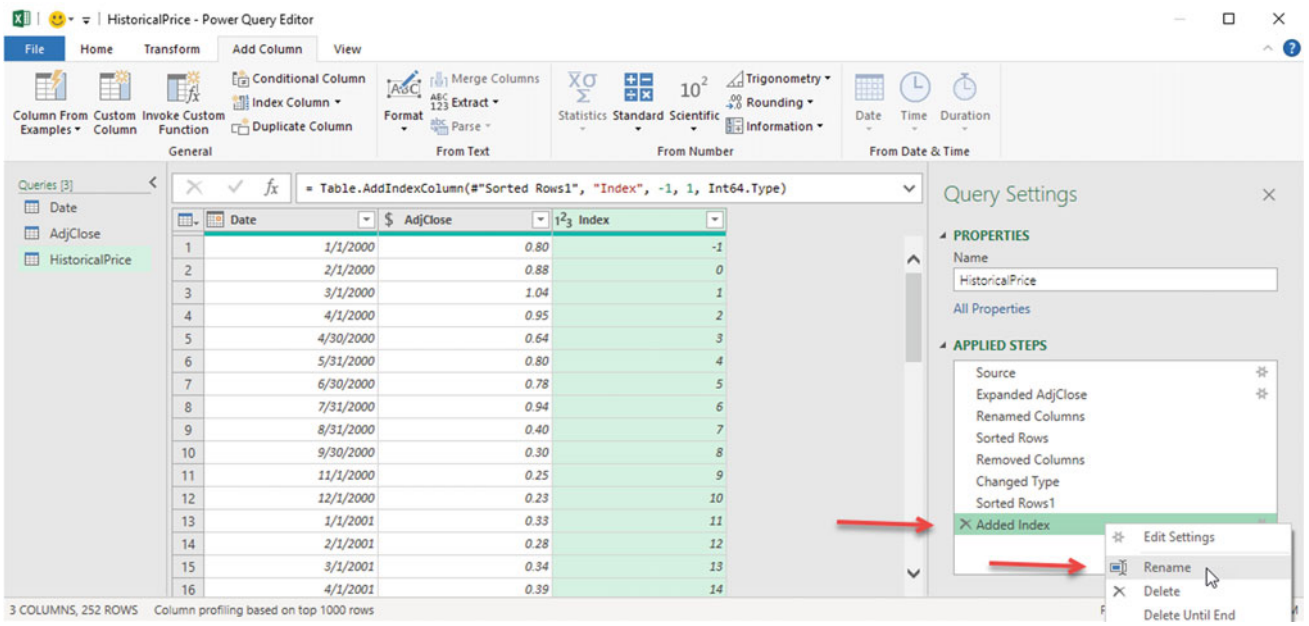
(1) First, select the *historicalPrice* query. Next, in the *Add Column* tab, select *Custom* from the *Index Column.*



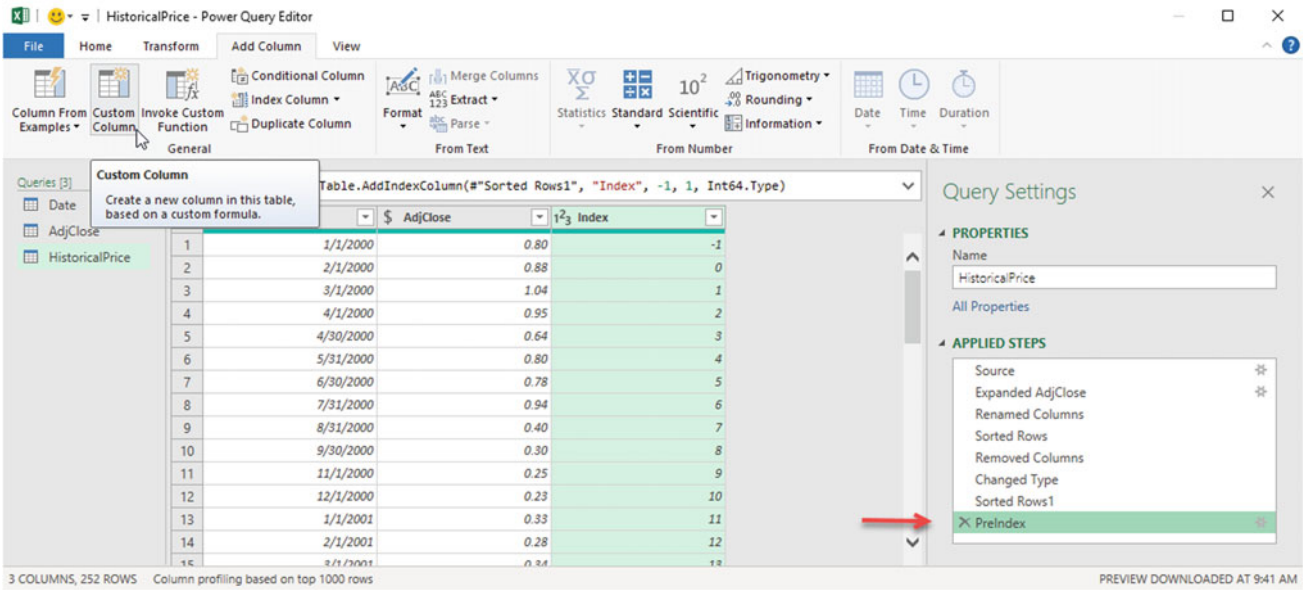
(2) Let's create an index column that starts with −1 and increments by 1. *It is important to note that Power Query is 0 based.*



(3) In the *APPLIED STEPS* section, the name of the new table is a called *Added Index*. Let's rename the table to *PreIndex* by selecting it and right-mouse click and choose *Rename*.



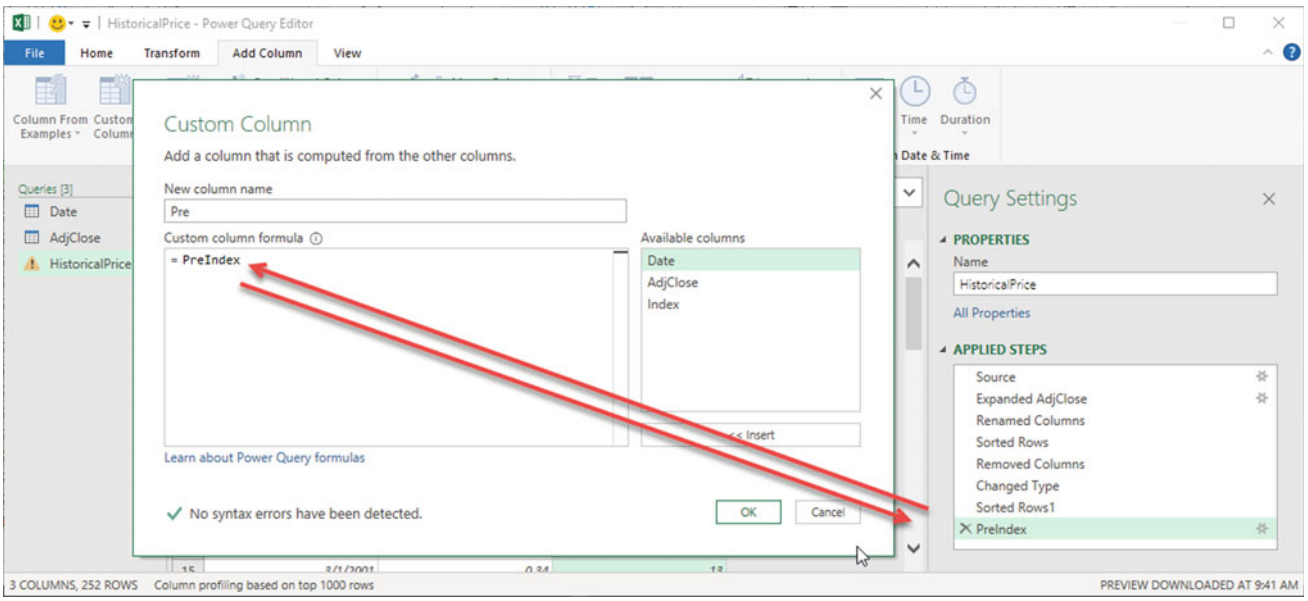
(4) Click on *Custom Column*. Notice that the last applied step is "PreIndex," which is a table.



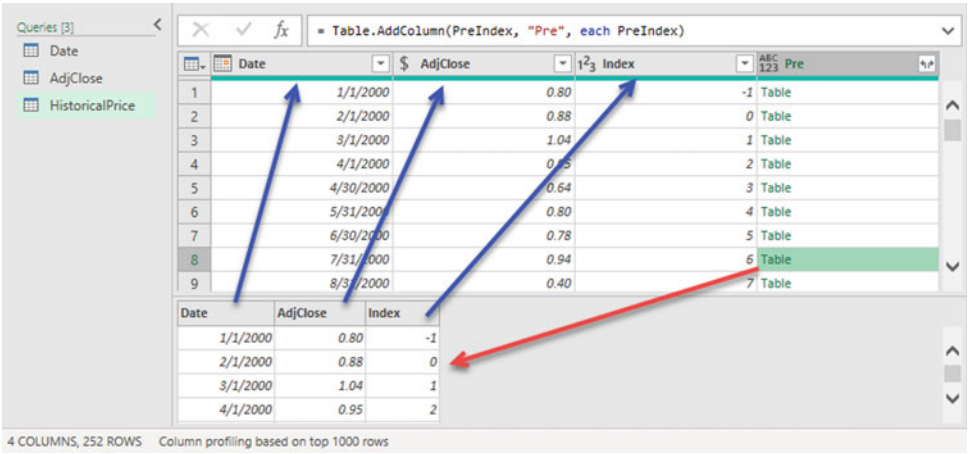
(5) Do the following:

a. Name the new column *Pre*.

b. Make the custom formula as = *PreIndex*.



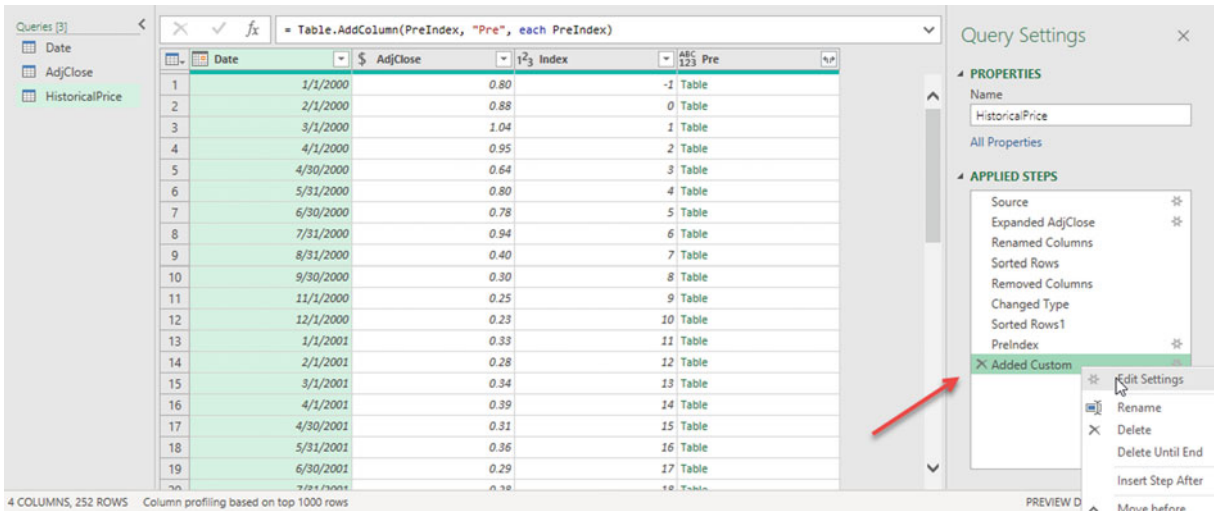
(6) The *Power Query* table has *table* elements for the column *Pre*.



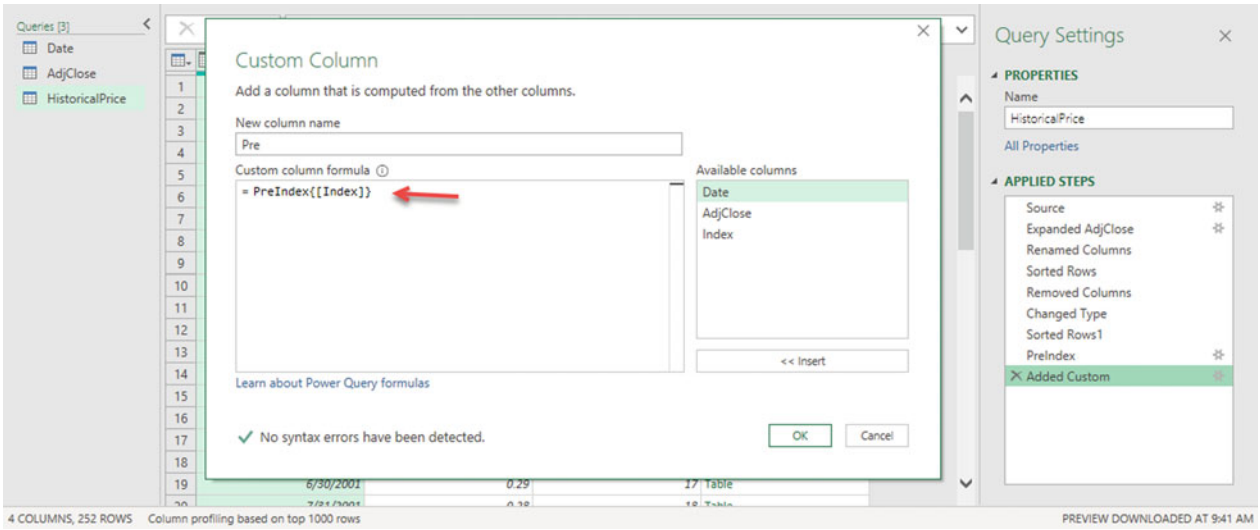
3.2.1.4 Index Column Records

Instead of returning tables, we will modify *Power Query* to return the correct record for each row.

(1) Do the following:
   a. Right-mouse click on *Added Custom* in the *APPLIED STEPS* section.
   b. Select *Edit Settings*

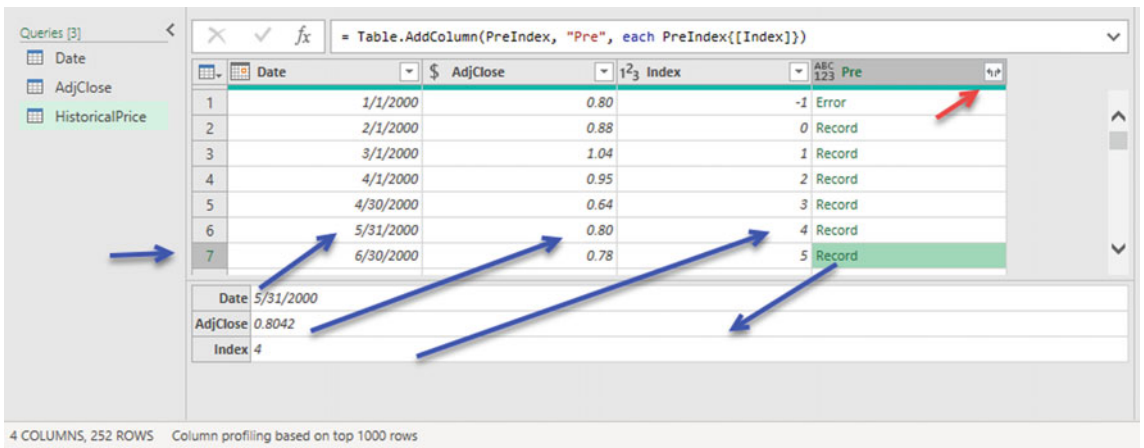


(2) Enter the *Custom column formula* as
   a. PreIndex{[Index]}.



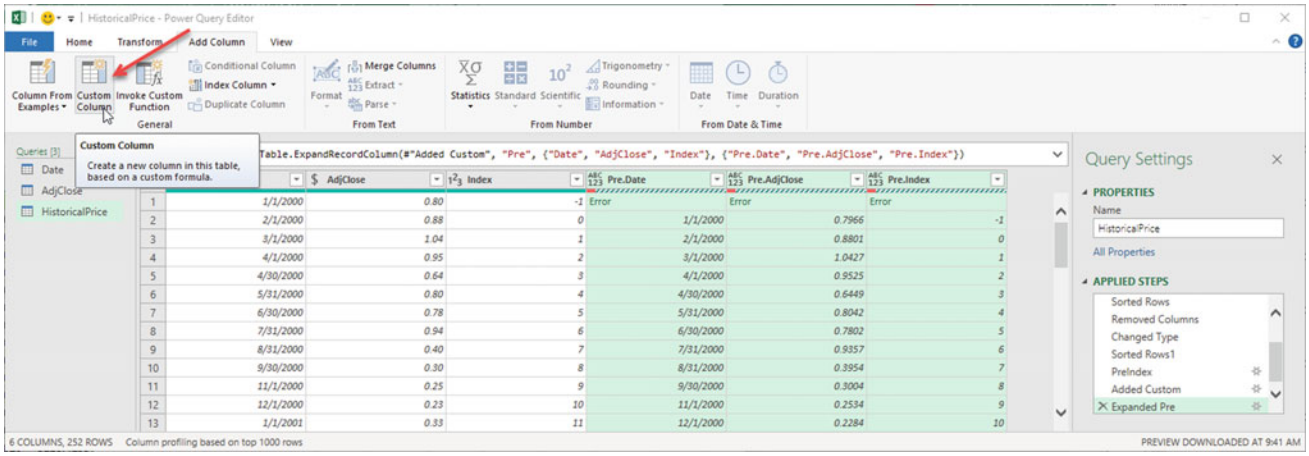
This formula tells *Power Query* to return the record for each row based on the *Index* field.

(3) Click on the expand button in the *Pre*field. Notice that the record for each row is the previous row. The first row has the value *Error* because no previous row is available.



3.2.1.5 Rate of Return

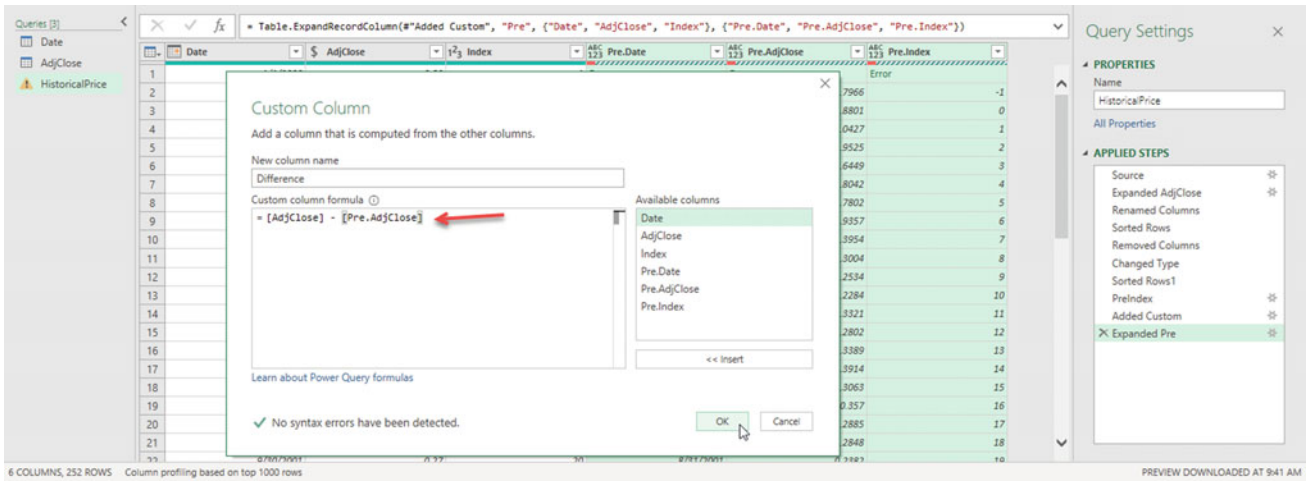
(1) Click on *Custom Column* from the *Add Column* tab. We now will add a column that calculates the difference in price from the previous period.



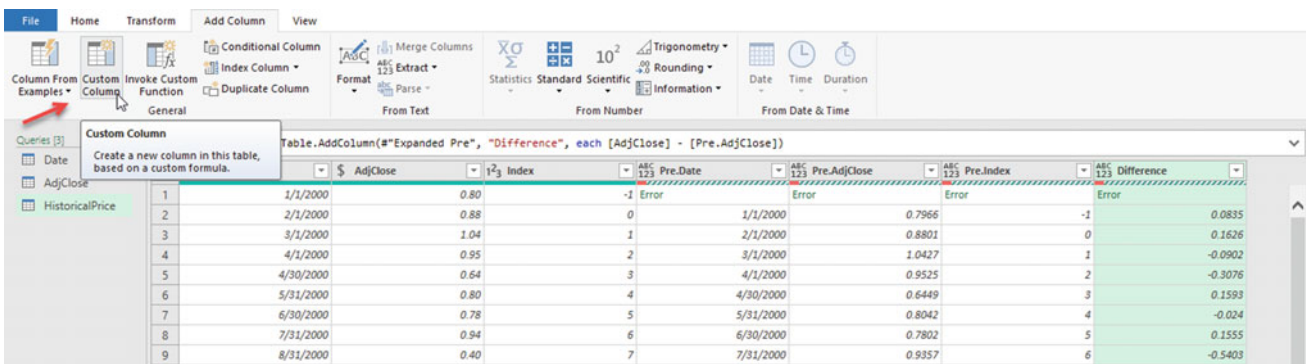
(2) Do the following:

a. *New column name* could be "Difference."

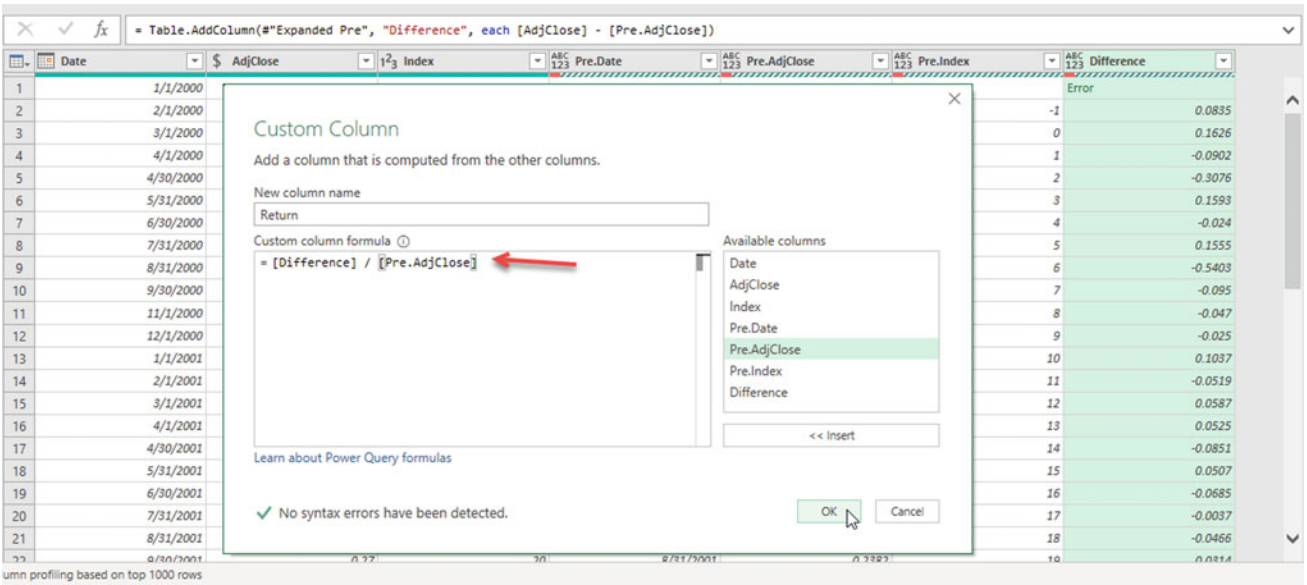
b. *Custom column formula* should be "= [AdjClose] − [Pre.AdjClose]."



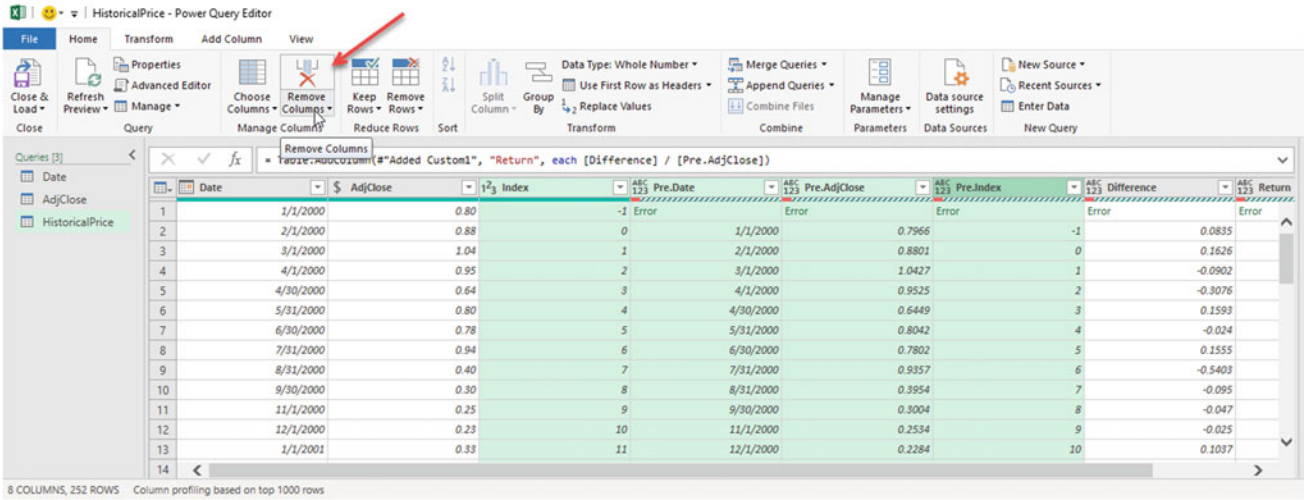
(3) Click on *Custom Column* to add the "return" column.



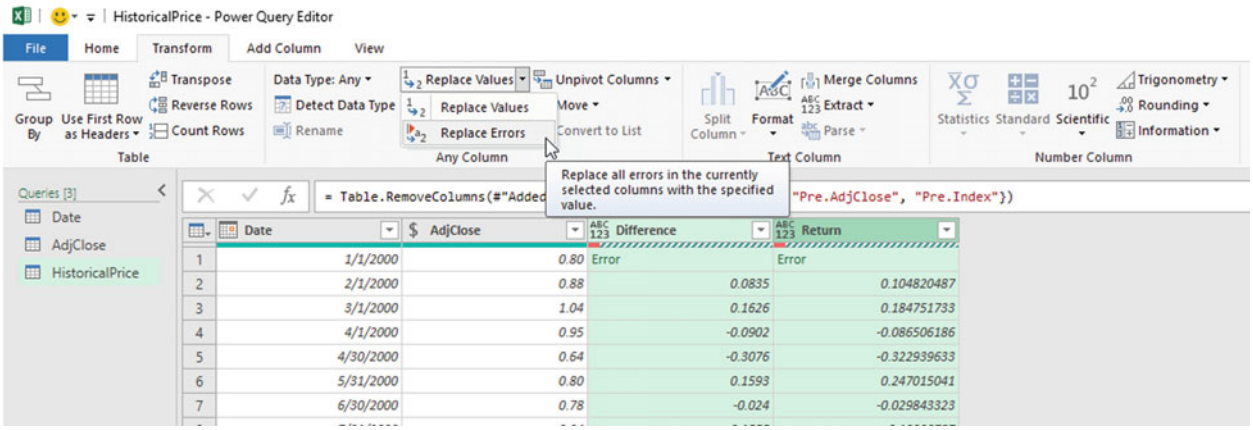
(4) Do the following:
   a. *New column name* should be "Return."
   b. *Custom column formula* should be "= [Difference]/[Pre.AdjClose]."



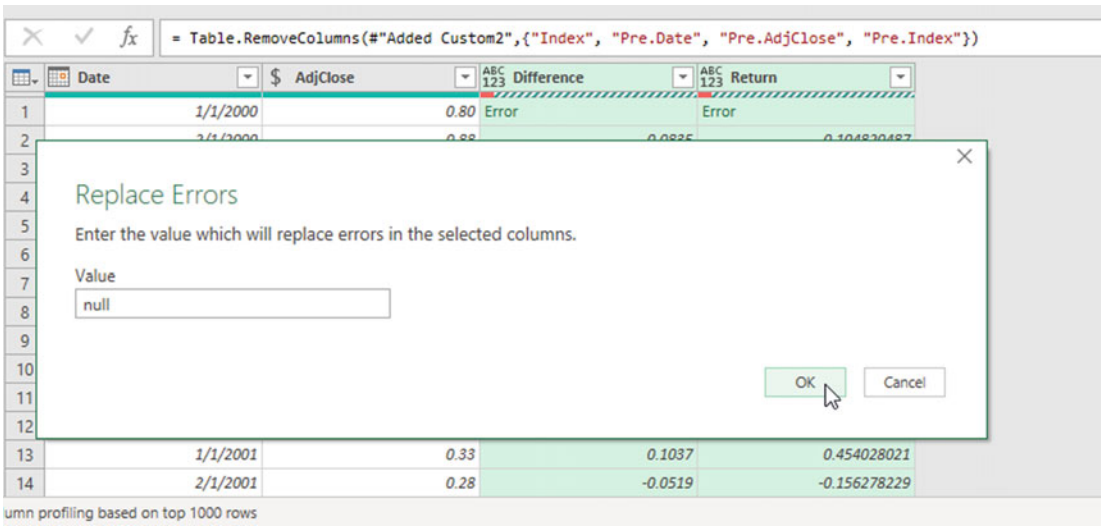
(5) Do the following:
   a. Select the columns *Index, Pre.Date, Pre.AdjClose, Pre.Index.*
   b. On the *Home* tab, click *Remove Columns.*



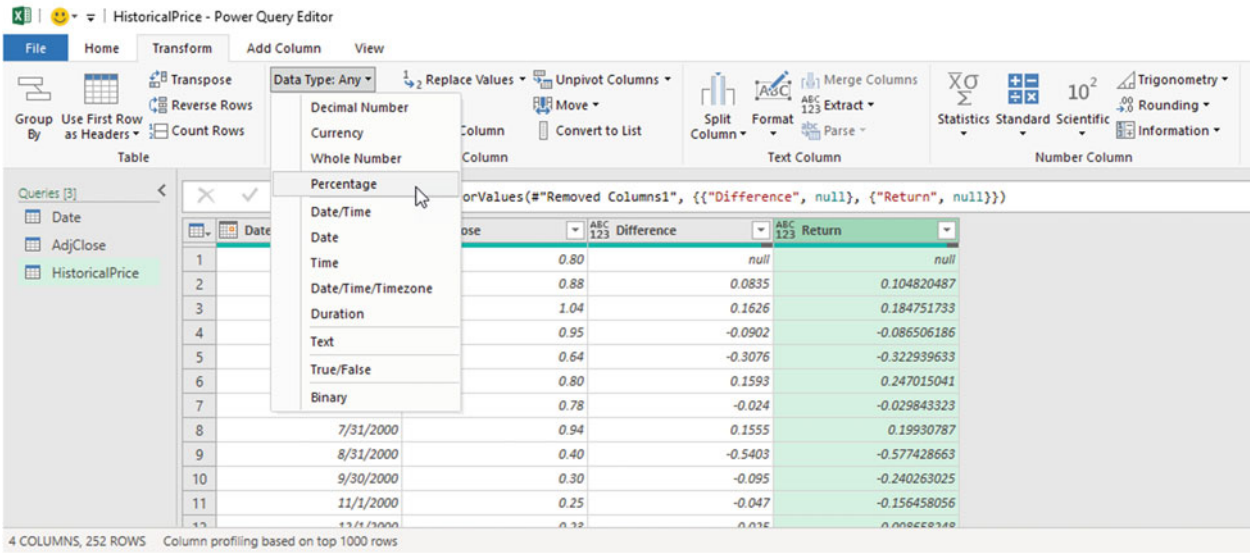
(6) Do the following:
   a. Select the columns *Difference, Return.*
   b. Click on *Replace Errors* from the *Transform* tab.



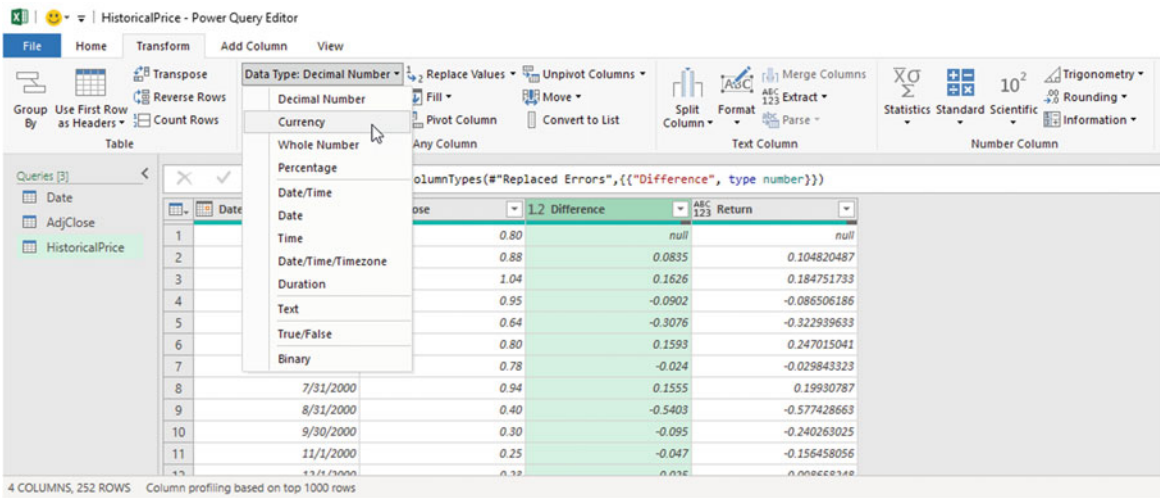
(7) Enter *null.*

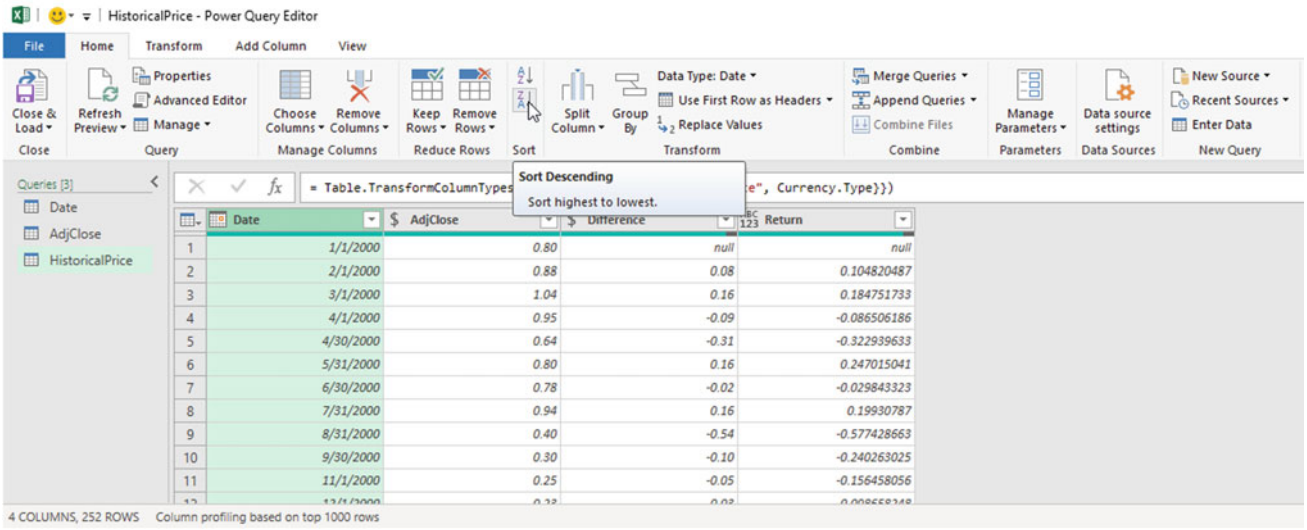


(8) Do the following:
   a. Select the *Return* column.
   b. Click on *Percentage* from the *Transform* tab.



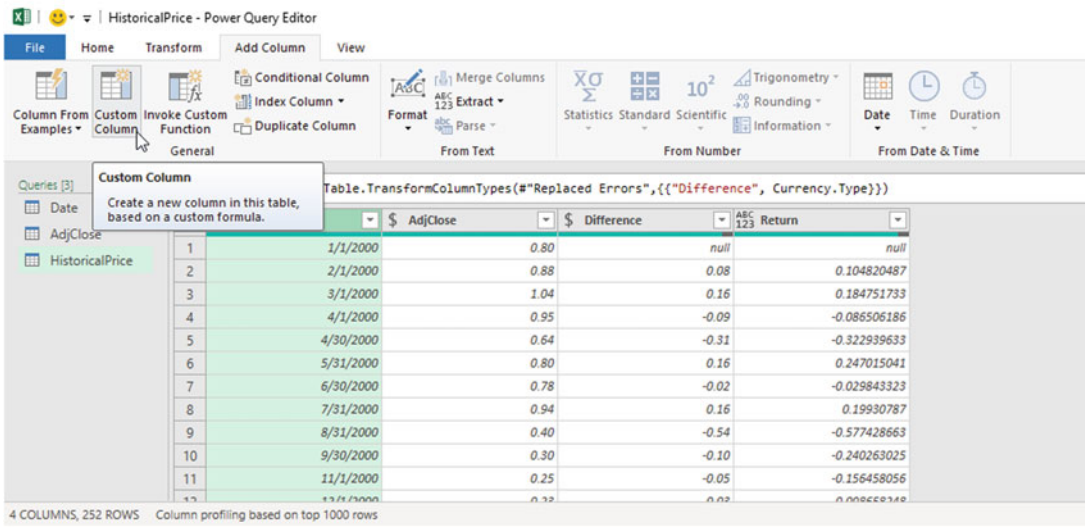
(1) Do the following:
   a. Select the *Difference* column.
   b. Click on *Currency* from the *Transform* tab.



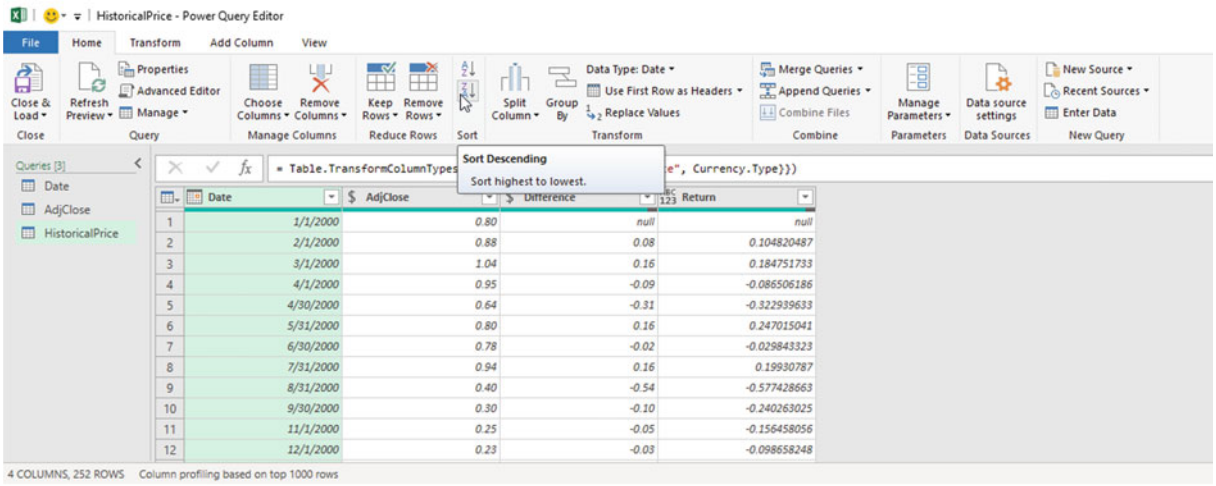


(2) Let's add a column to indicate the ticker we are calculating for the rate of return.

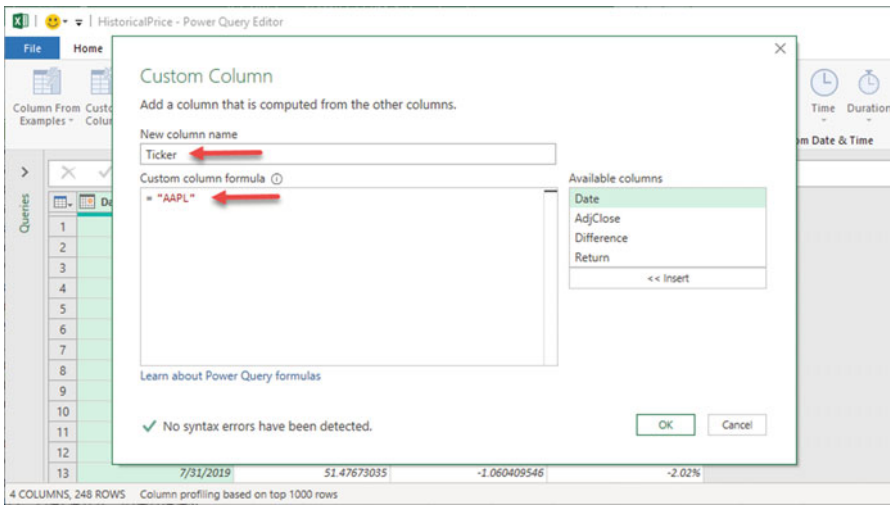
a. Click on *Custom Column*.



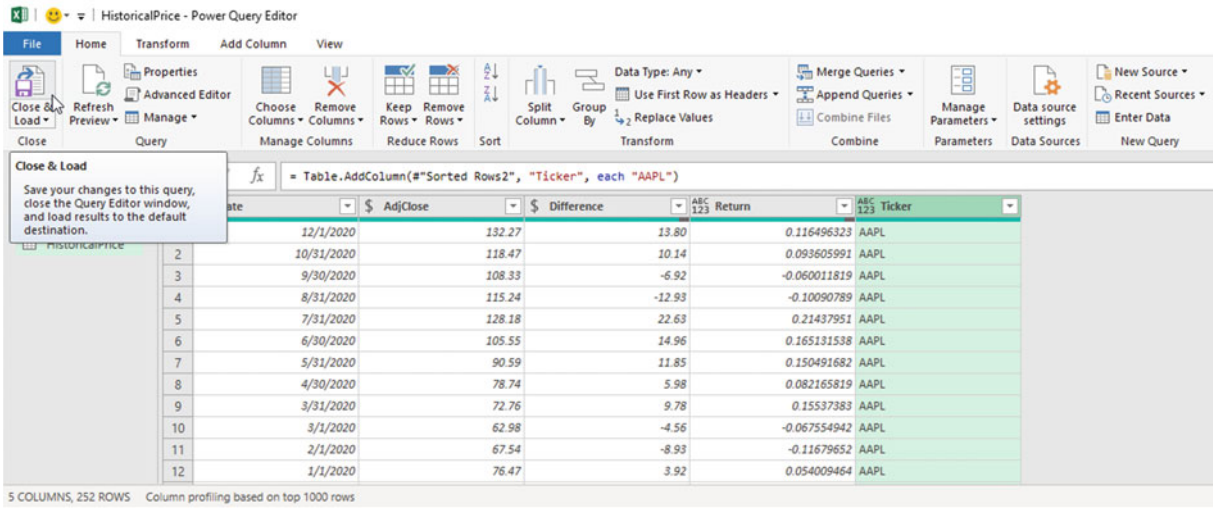
(3) Do the following:
   a. Select the *Date* column.
   b. Click on the *Sort Descending* icon.



(4) Do the following:
   a. *New column name* should be "Ticker."
   b. *Custom column formula* should be "= AAPL."



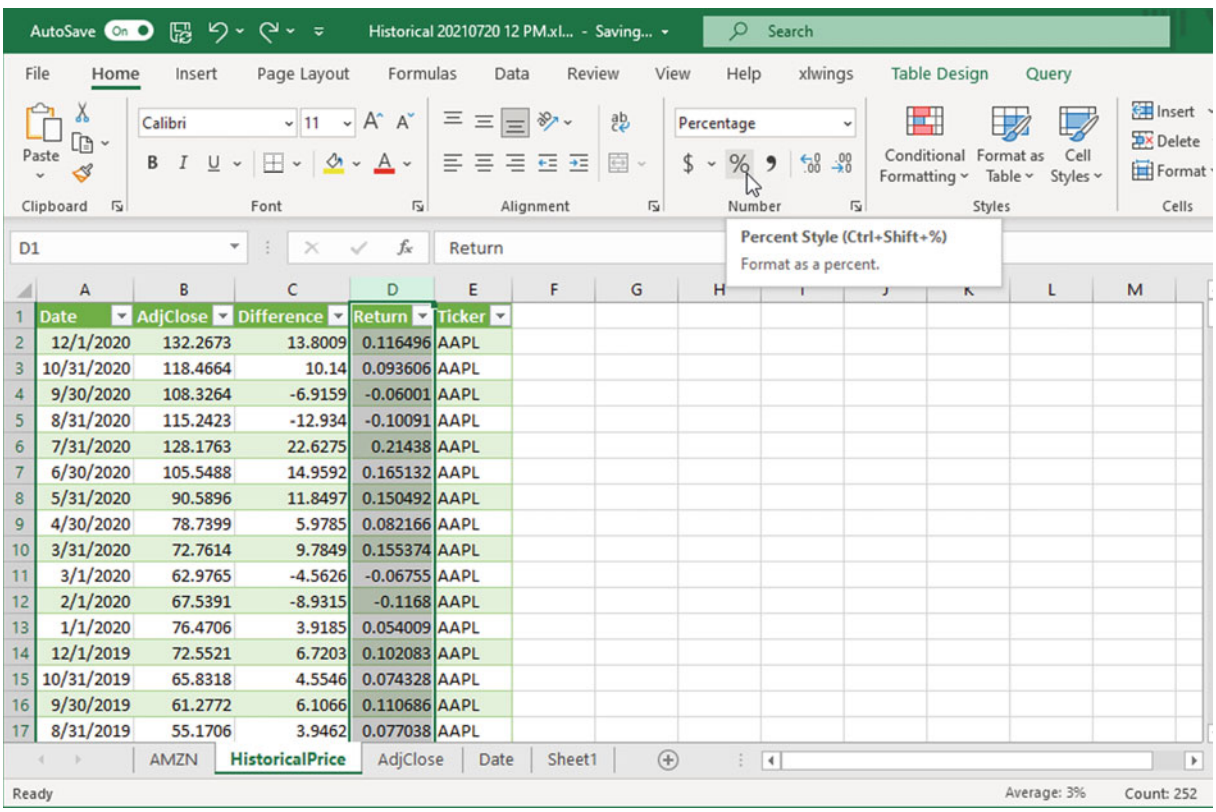
(5) Click on *Close & Load.*



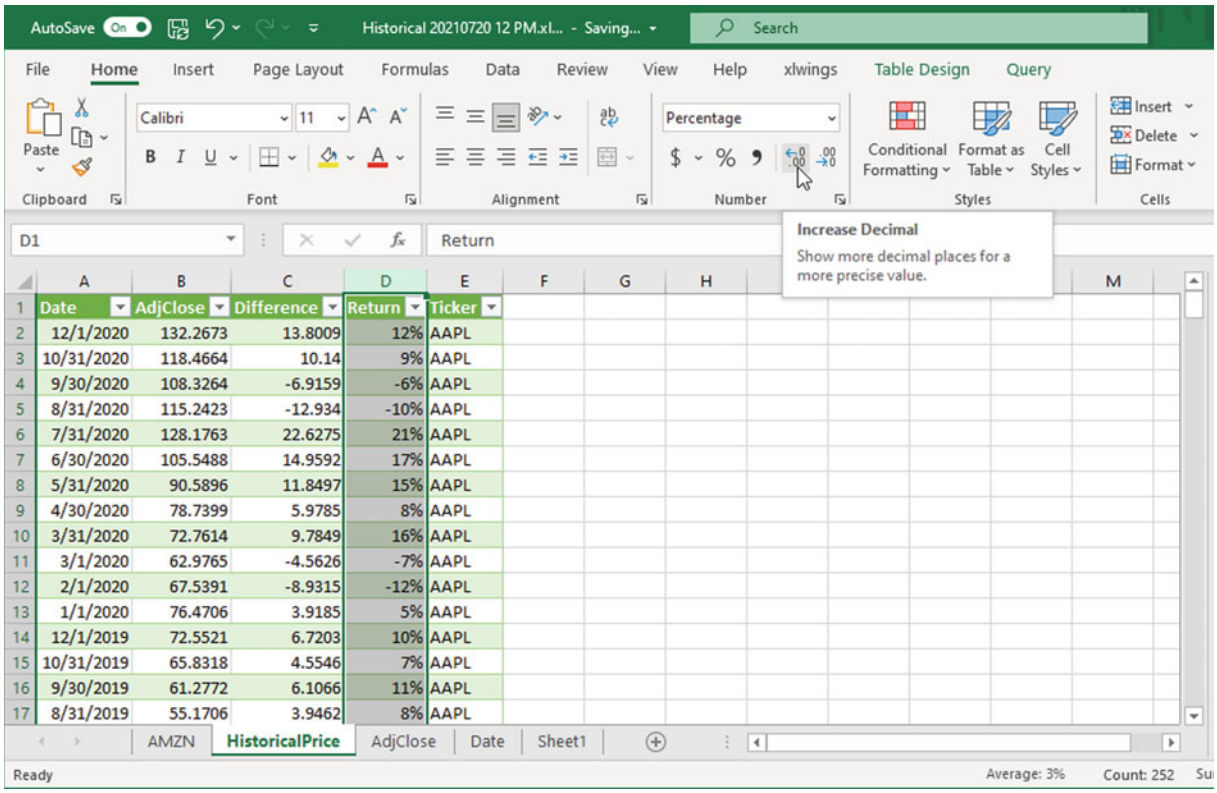
(6) Do the following:

a. Select column D.

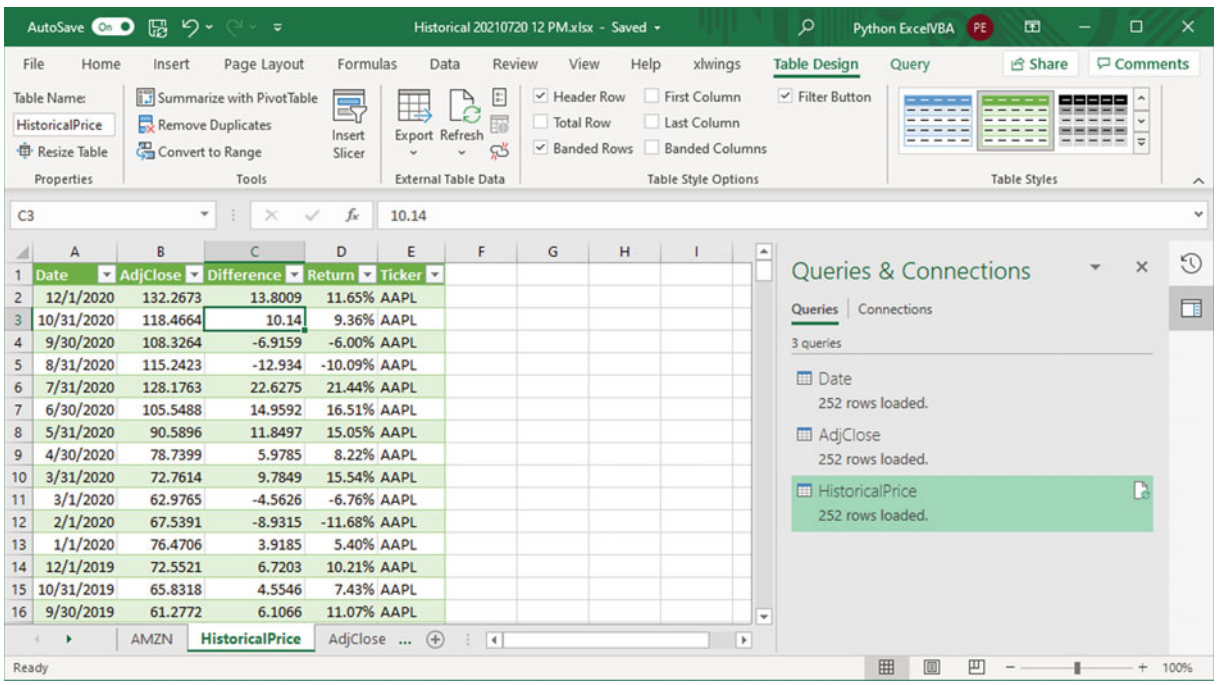
b. Click on the *Percentage Style* icon.



(7) Click on *Increase Decimal* twice.



Below shows the rate of return for AAPL.

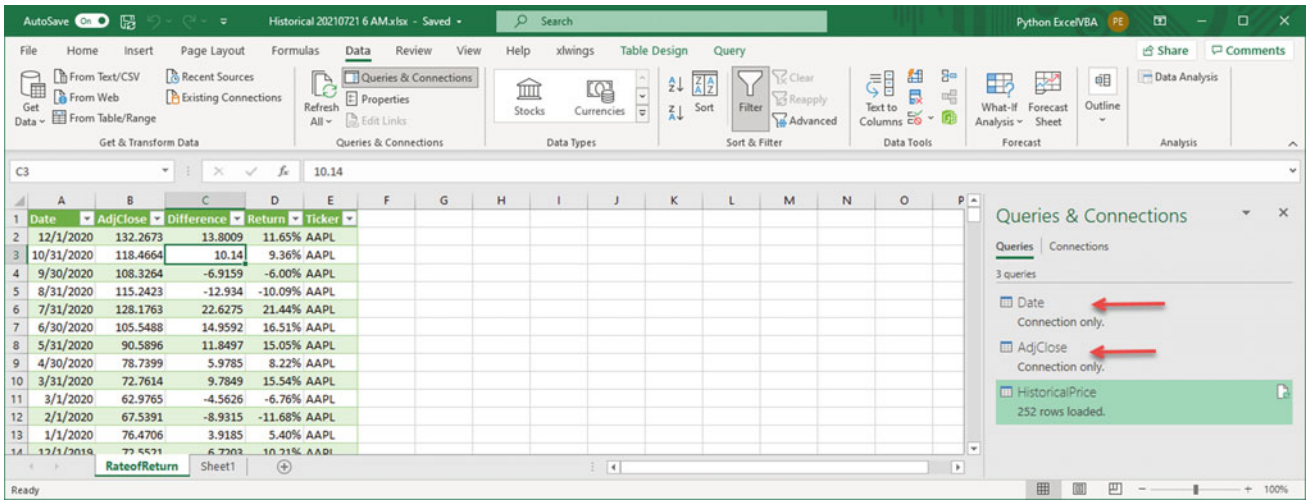


3.2.2 Dynamic Power Query

We spent a lot of time and effort setting up Power Query to retrieve and calculate the return rate for the AAPL ticker. It turns out that the same steps are required for other tickers like MSFT, WMT, and AMZN. Instead of spending the time and effort for each ticker, we will set it up so we can indicate the ticker, start time, and end time and then click on the *Refresh All* in the *Data* tab to calculate the *rate of return* for the ticker that we want; this means that all our time and effort is spent on the first ticker and almost no time on the following tickers.

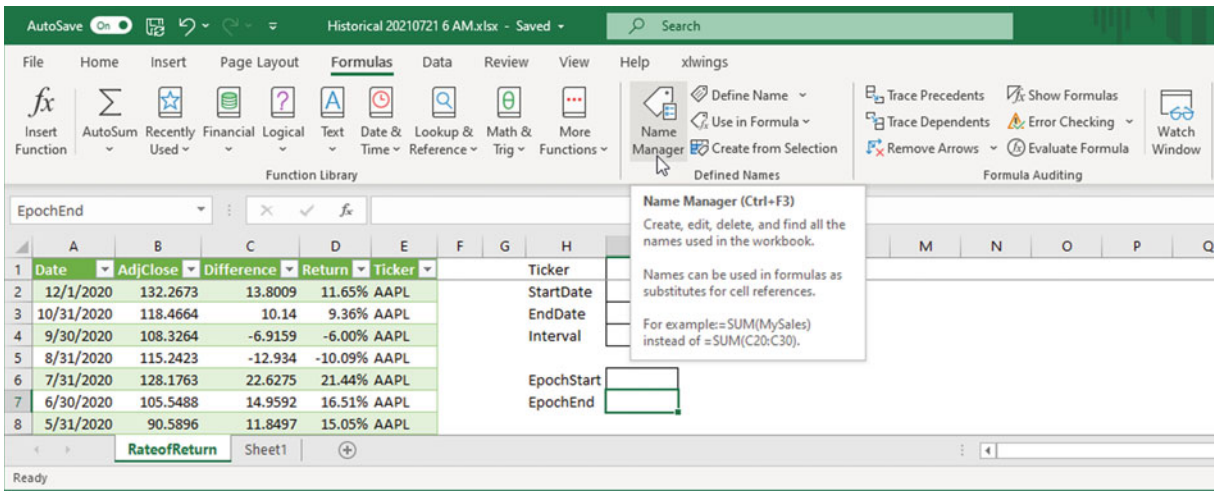
Our *Power Query* starting point will be the *Power Query* from the last section. From that *Power Query,* let's delete sheets *AMZN*, *AdClose*, and *Date* and rename *HistoricalPrice* to *RateOfReturn*.

If we click on *Queries & Connections* in the *Data* tab, we can see that the *Date* query and the *AdjClose* are "Connection only"; this means that the dataset from these queries is not loaded to a worksheet. The query *HistoricalPrice* is loaded to the worksheet *RateofReturn*.

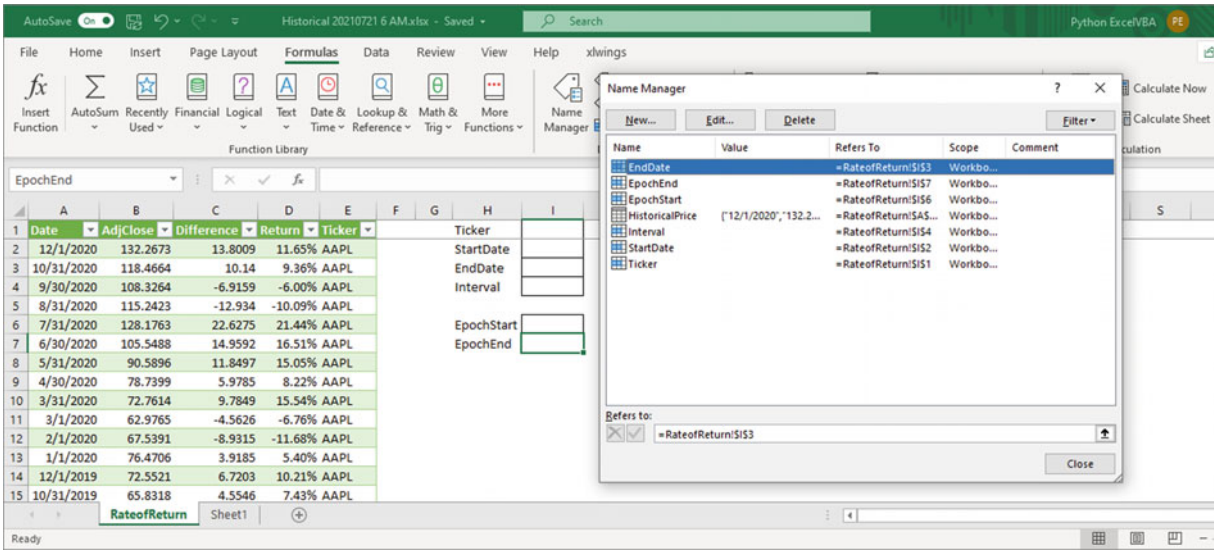


Do the following steps to create the *Microsoft Excel Rate of Return* template:

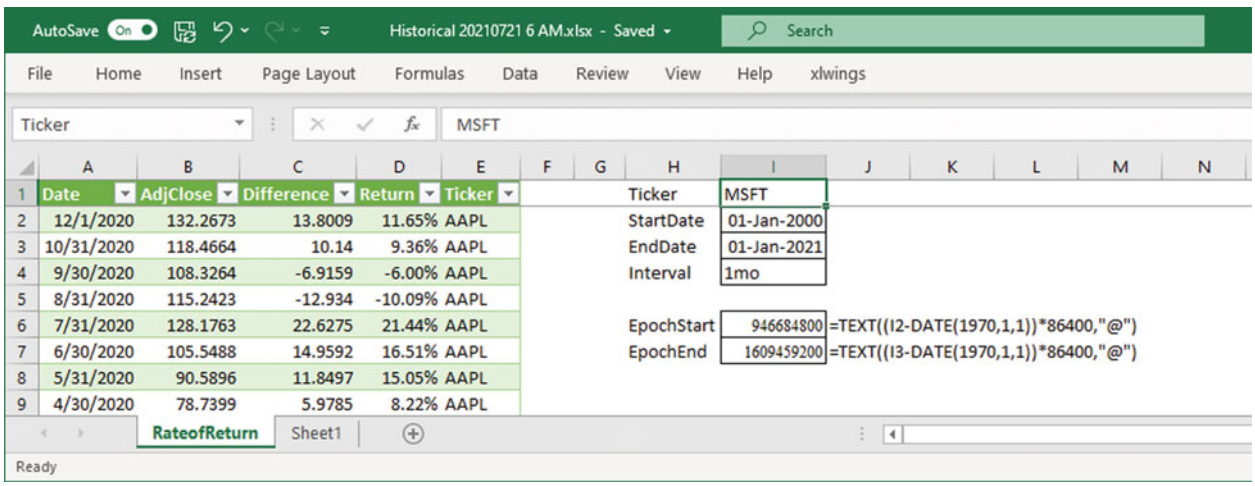
(1) Click on *Name Manager* in the *Formulas* tab to create range ranges.



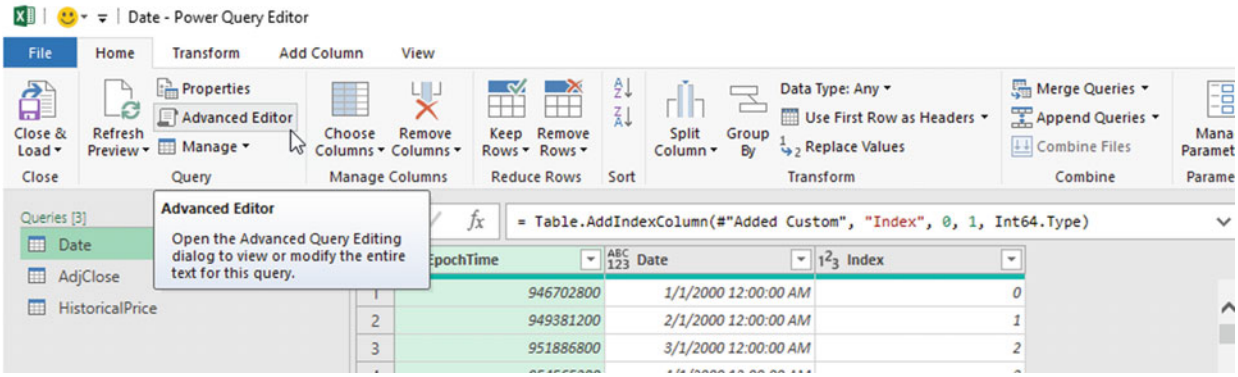
(2) Use the *Name Manger* to create range names for Ticker, StartDate, EndDate, Interval, EpochStart, EpochEnd. The cell these ranges names *Refers To* are shown in the *Name Manger* below.



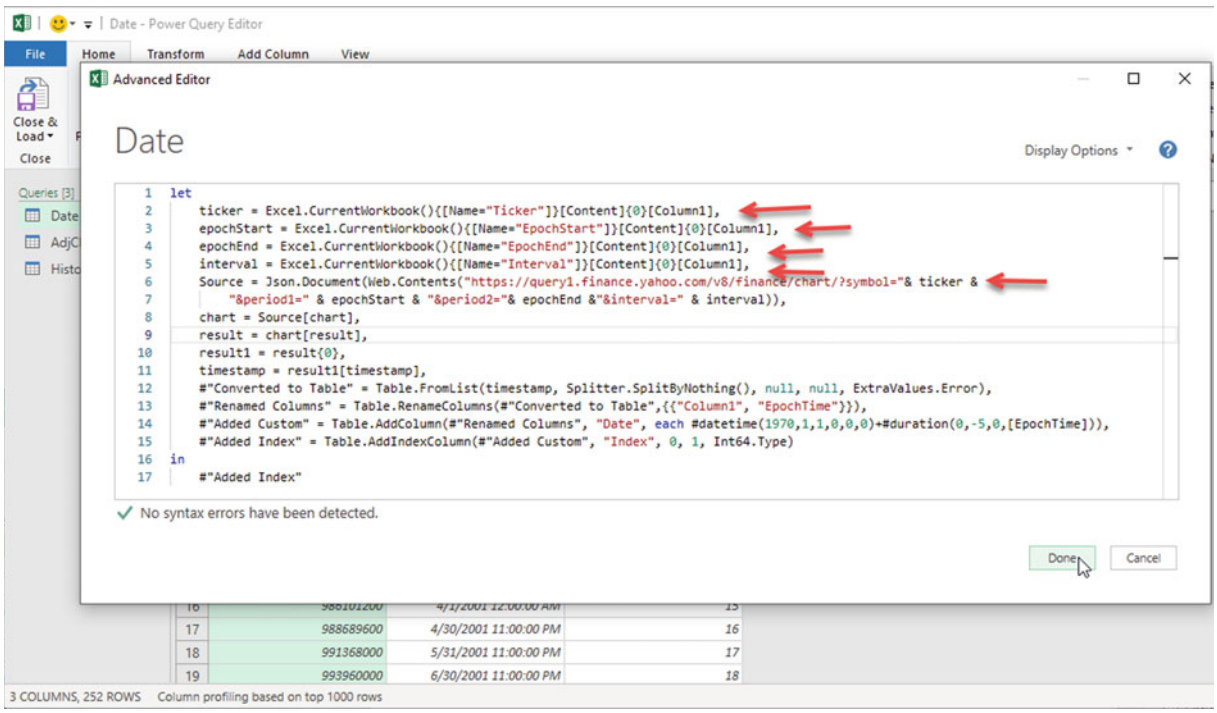
(3) Enter cells I1 to I7, as shown below.



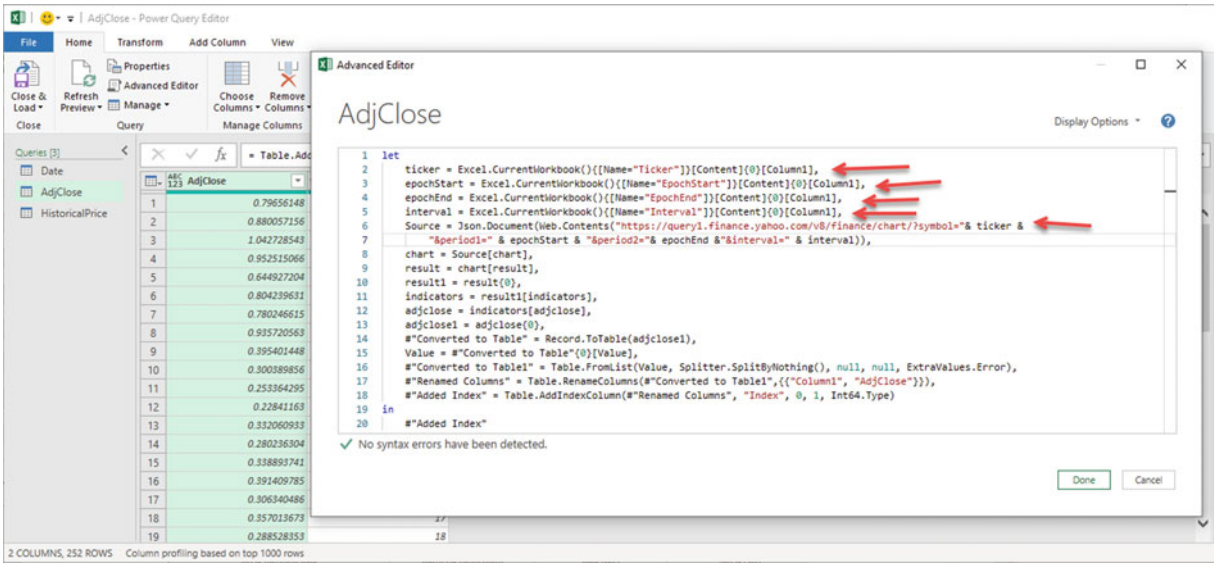
(4) Do the following:
   a. Go to *Power Query Editor*.
   b. Select the *Date* query.
   c. Click on the *Advanced Editor*.



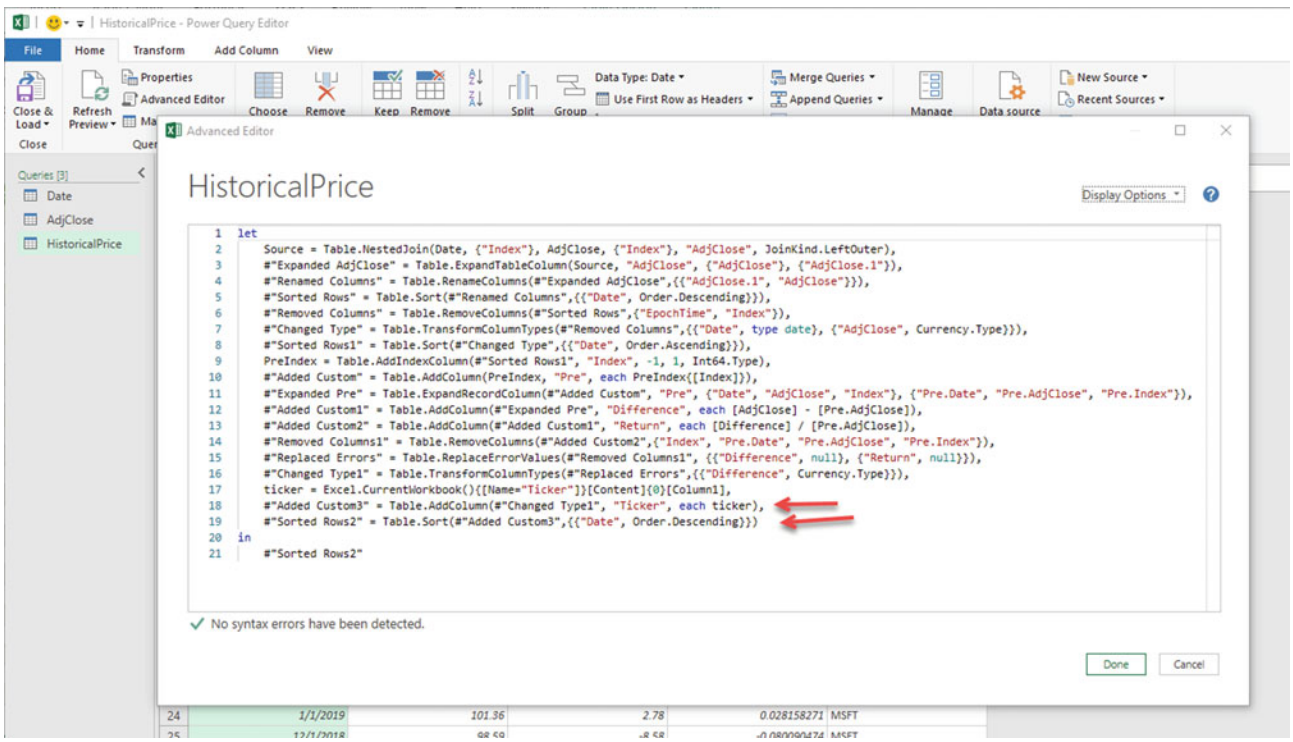
(5) Insert lines 2–5 and update line 6 as shown below. Lines 2–5 make *Power Query* dynamic. Lines 2–5 retrieve values from the Excel range names indicated.



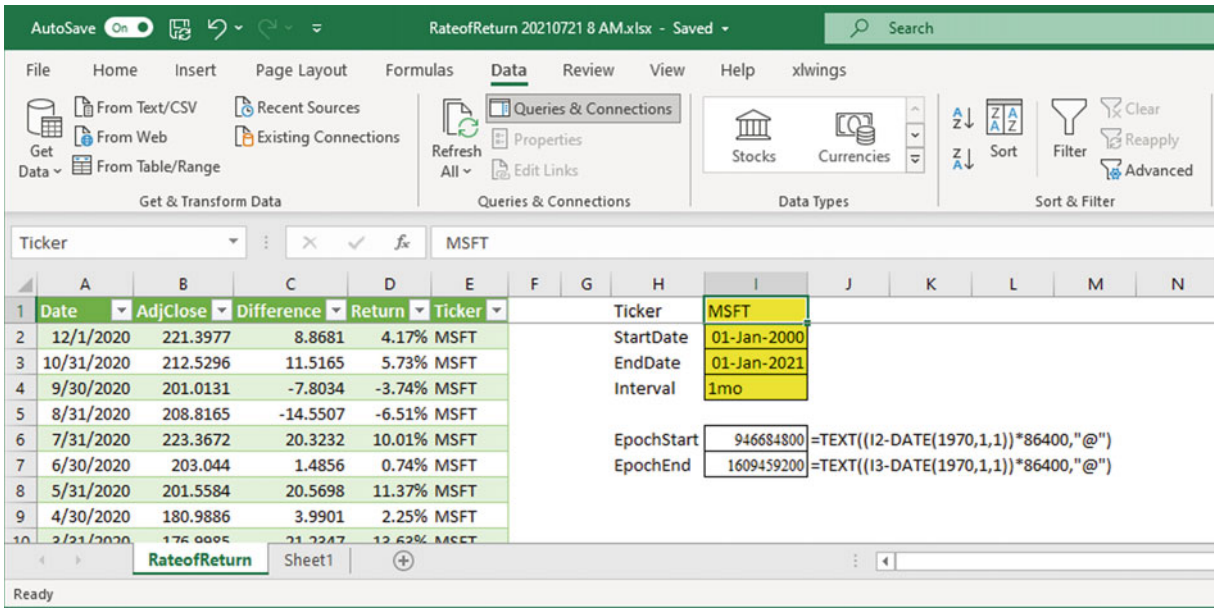
(6) Do the same thing for lines 2–6 for the *AdjClose* query.



(7) The last two lines in the *HistoricalPrice* display the ticker imputed in the Excel range *ticker*.



The Microsoft Excel Rate of Return template is now complete. You can now update the yellow cells in the worksheet *RateOfReturn* and then click on *Refresh All* in the *Data* tab to retrieve the historical data and calculate the rate of return for the ticker indicated and for the date range indicated.



It is important to note that we accomplished a lot without doing any VBA program.

3.2.3 Python

In this section, we will use the *Python* library *yahoo_fin* to retrieve the historical data of a ticker. The URL for the documentation of *yahoo_fin* is http://theautomatic.net/yahoo_fin-documentation/. Another thing to note is that the *Python* code uses the Yahoo! Finance API to retrieve the long name of a ticker. The Yahoo! Finance API is used in line 7 of the code.

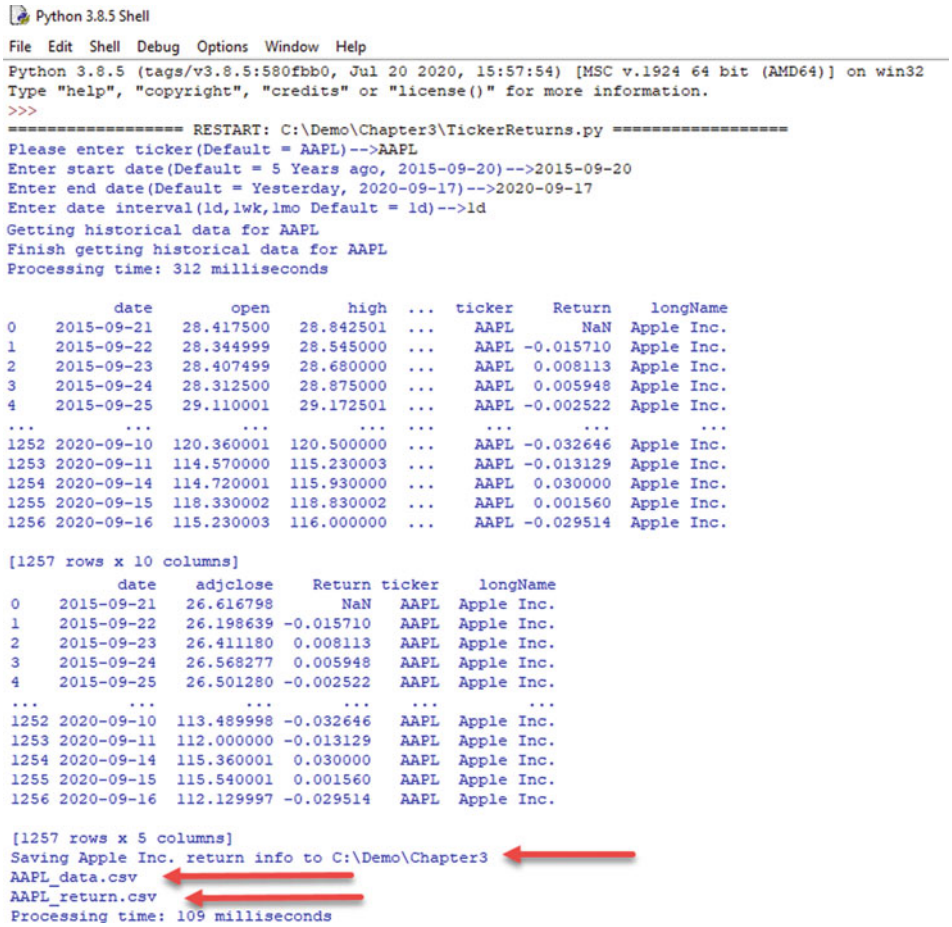
3.2.3.1 Code

TickerReturns.py

C: > Demo > Chapter3 > TickerReturns.py > ...

```
#Sep-18-2020 11 AM
import yahoo_fin.stock_info as yahoo_Fin # http://theautomatic.net/yahoo_fin-documentation/
import requests,json, os,time, pandas as pd,datetime as dt

#https://query1.finance.yahoo.com/v7/finance/quote?symbols=AAPL sample Yahoo! Finance API
def yahoo_TickerInfo(Ticker,Attribute):
    response = requests.get("https://query1.finance.yahoo.com/v7/finance/quote?symbols=" + Ticker)
    responseText = json.loads(response.text)
    quoteResponse = responseText["quoteResponse"]
    return(quoteResponse["result"][0][Attribute])

#initialize dates
previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
years_ago = dt.date.today()-dt.timedelta(days=5 * 365)
Default_start = years_ago.strftime("%Y-%m-%d")

# request user input
# set default values if users does not enter a value
ticker = input('Please enter ticker(Default = AAPL)-->') or 'AAPL'
startDate = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
endDate = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
ticker_interval = input('Enter date interval(1d,1wk,1mo Default = 1d)-->') or '1d'

print("")
print("Getting historical data for " + ticker)
starttime = int(round(time.time() * 1000))
#get ticker history
tickerData = yahoo_Fin.get_data(ticker , start_date = startDate , end_date = endDate,index_as_date= False)
print("Finish getting historical data for " + ticker)
print( "Processing time: " + str(int(round(time.time() * 1000)) - starttime) + " milliseconds")

#pull out 'adclose' to a series
adjclose=tickerData['adjclose']
adjclose = adjclose.pct_change() #calculates rate of return with prior period
adjclose= adjclose.dropna()

tickerData = tickerData.dropna(how = 'all')
tickerData['Return']=adjclose #append return info back to dataframe
tickerData['longName'] = yahoo_TickerInfo(ticker,"longName")
#only get subset of tickerData
tickerReturn = tickerData[['date','adjclose','Return','ticker','longName']]

#show sample datat and return data
print("")
print(tickerData)
print(tickerReturn)

print('Saving ' + yahoo_TickerInfo(ticker,"longName") + ' return info to ' + os.getcwd())
starttime = int(round(time.time() * 1000))
tickerData.to_csv(ticker + '_data.csv')
tickerReturn.to_csv(ticker + '_return.csv')
print(ticker + '_data.csv')
print(ticker + '_return.csv')
print( "Processing time: " + str(int(round(time.time() * 1000)) - starttime) + " milliseconds")
print("")

```

3.2.3.2 Output



Note that the output indicates the folder where the complete dataset and return dataset is saved.

3.3 Histograms

In this section, we will create histograms to understand how the returns of a ticker are distributed and anchored.

3.3.1 Sturge's Rule

Histograms contain intervals. We will use *Sturge's Rule* to determine the number of intervals for a histogram. The formula is

$$\text{number of intervals} = 1 + 3.3\log(n).$$

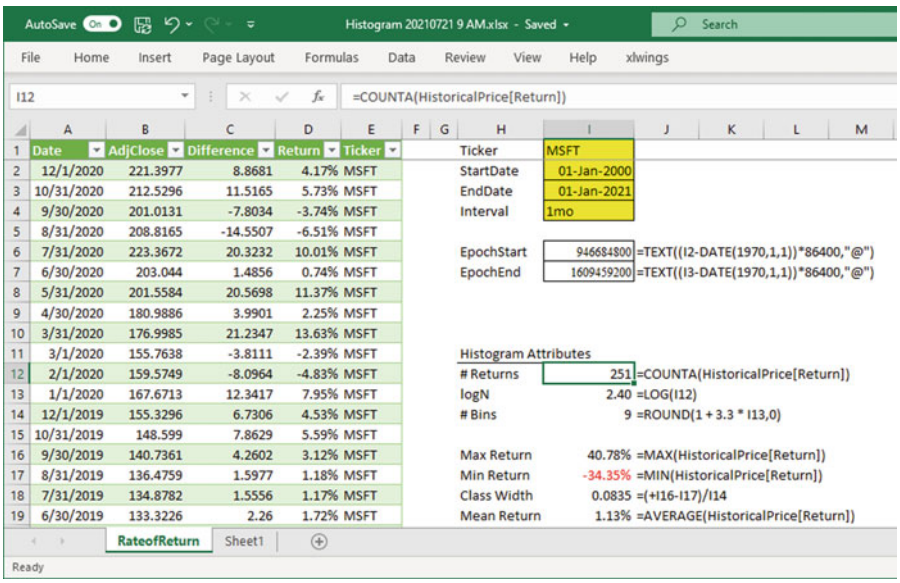
The following is the formula to determine the class width:

$$\text{class width} = (\text{largest number} - \text{smallest number})/\text{number of intervals}.$$

3.3.2 Microsoft Excel

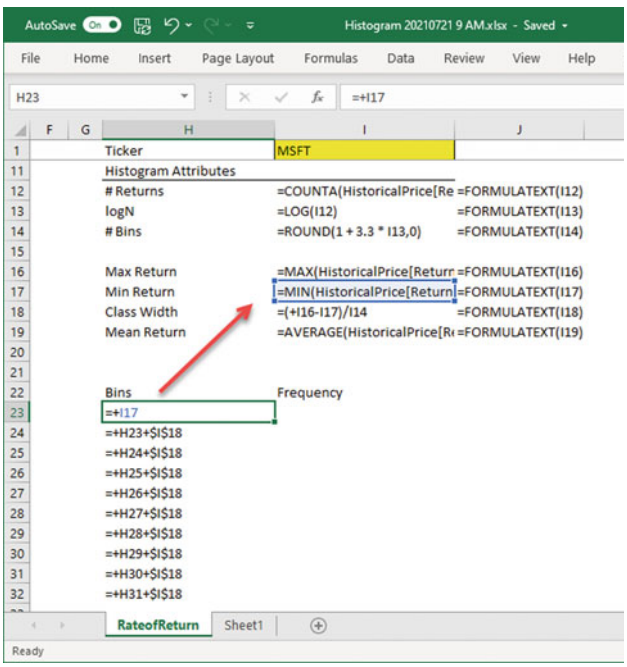
3.3.2.1 Frequency Function

This section will use Excel's *frequency* function to create MSFT's monthly rate of return histogram from January 1, 2000 to January 1, 2021.

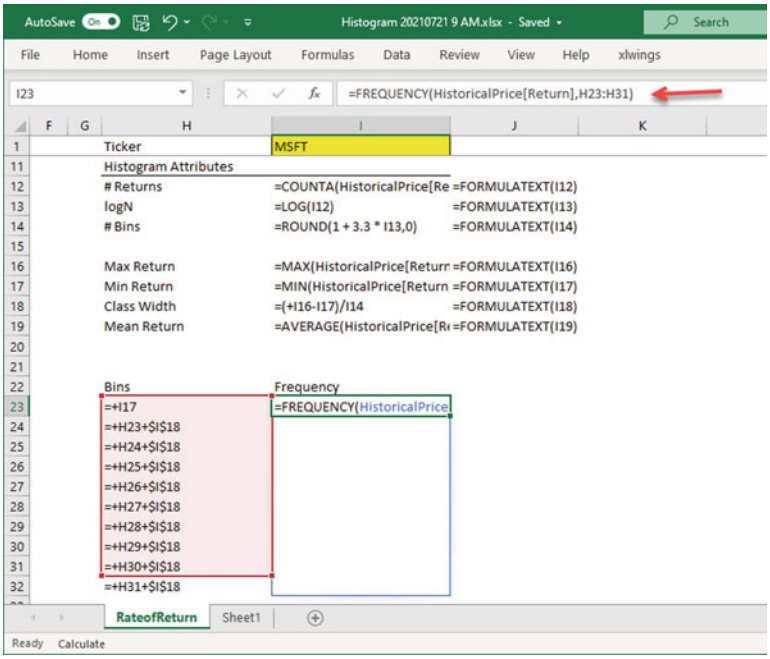


Cells I12 to I18 were created to create the histogram intervals based on *Sturge's Rule*.

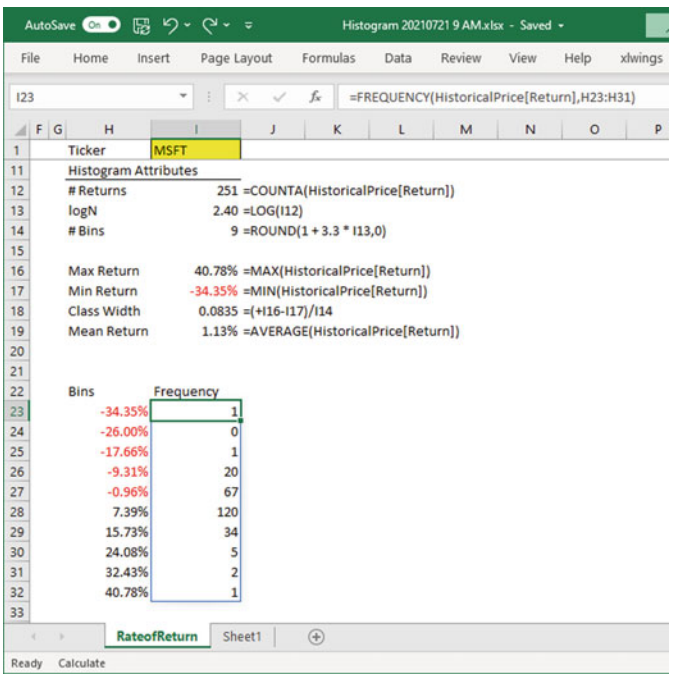
The bins for the histograms are from cells H23 to H32. Note that the first bin references the cell for the minimum return.



Enter the formula = *FREQUENCY(HistoricalPrice[Return],H23:H31)* in cell I23. Notice that the formula does not include cell H32.

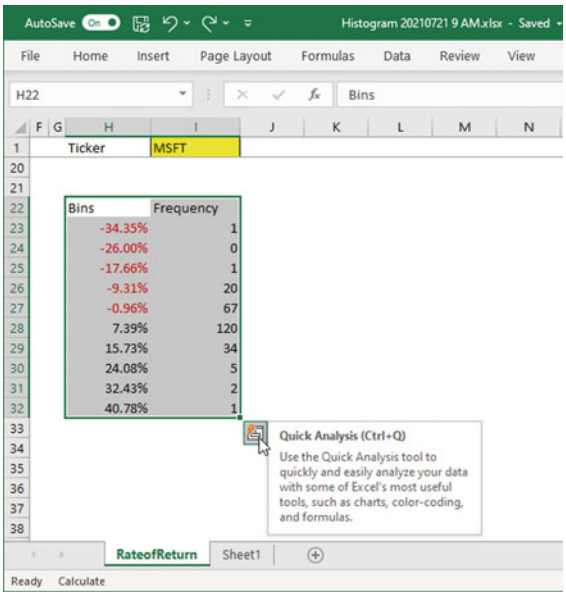


After pressing enter, Excel automatically fills in the cells I24 to I32 with the same formula. The *frequency* function is called a dynamic array formula. Go to the URL https://exceljet.net/dynamic-array-formulas-in-excel for more information about dynamic array formulas.



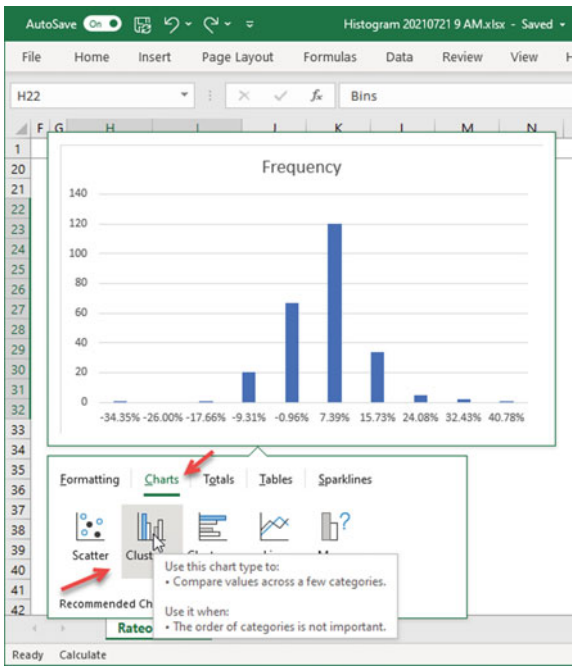
The first bin indicates the number of returns less than -34.35%. The last bin indicates the number of returns equal to or less than 40.78% and greater than 43.43%.

3.3.2.2 Quick Analysis

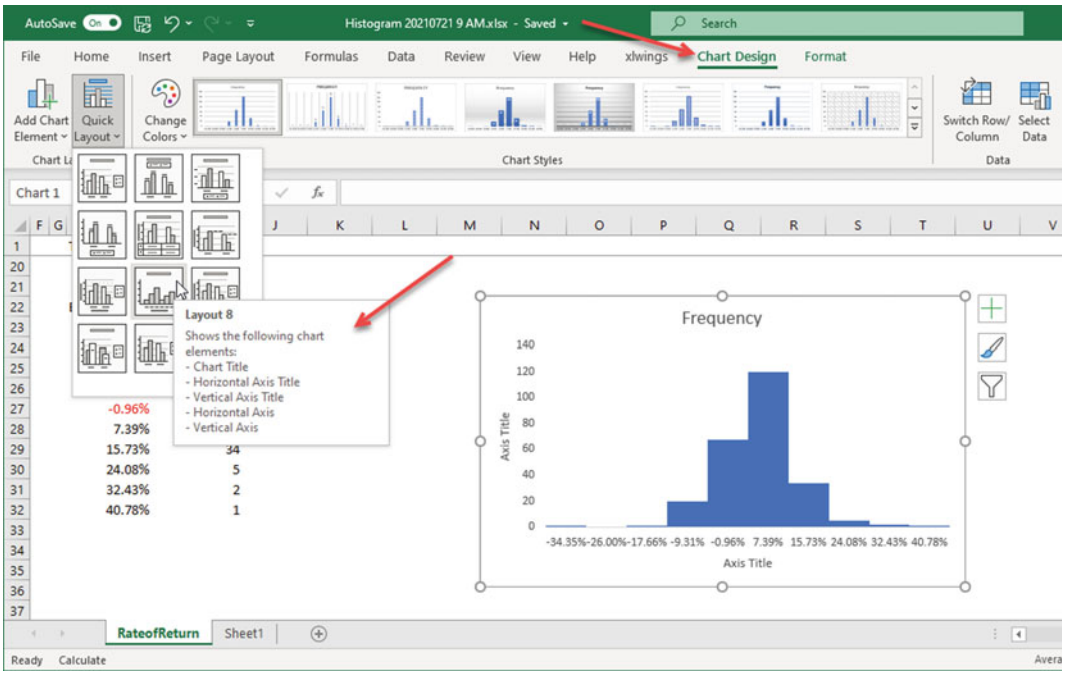


In this section, we will use the *Quick Analysis* feature to create a histogram feature.

(1) Select cells H22 to I32 and click on *Quick Analysis.*
(2) Click on *Cluster* in the *Charts* tab.



(3) In the *Chart Design* tab, choose *Layout 8*.

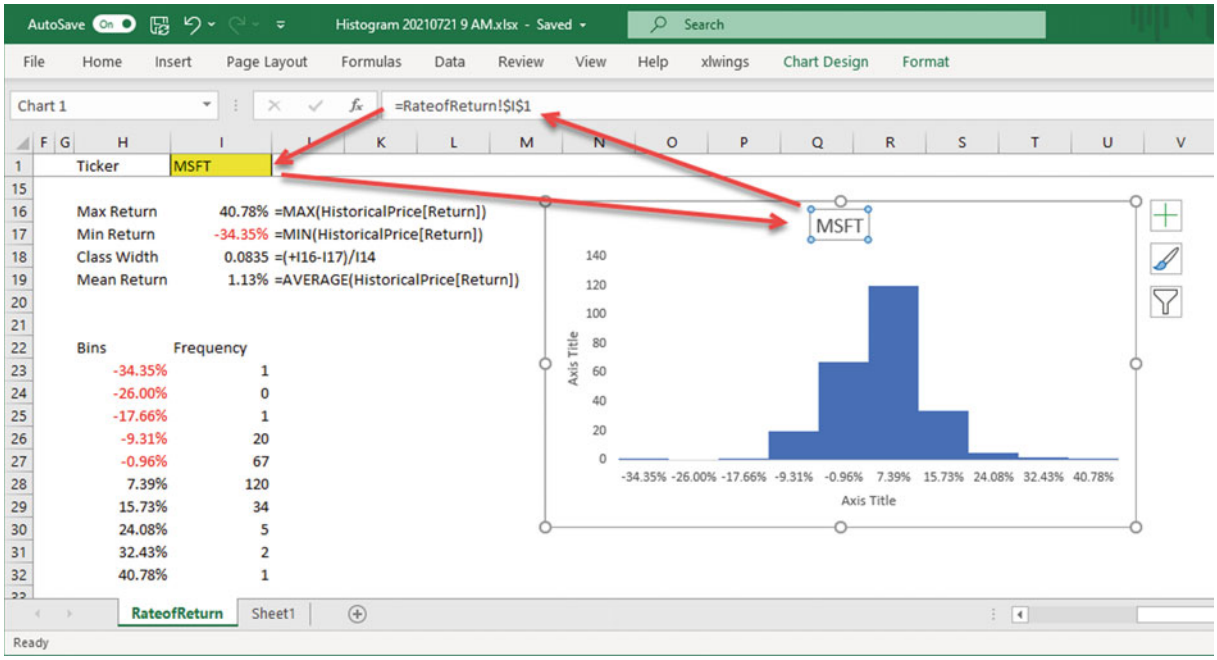


Formatting Histogram

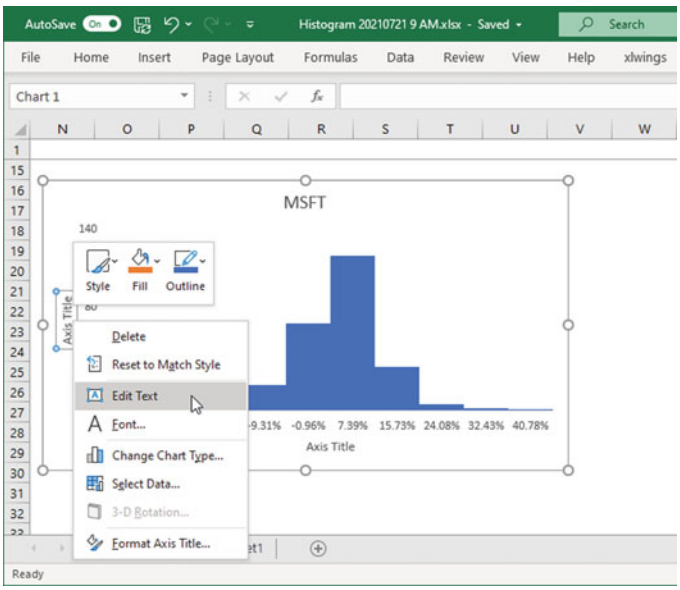
Dynamic Chart Title

In this section, we will create a dynamic chart title for the histogram. Every time the ticker cell in cell I1 is updated, the histogram's chart title will be updated.

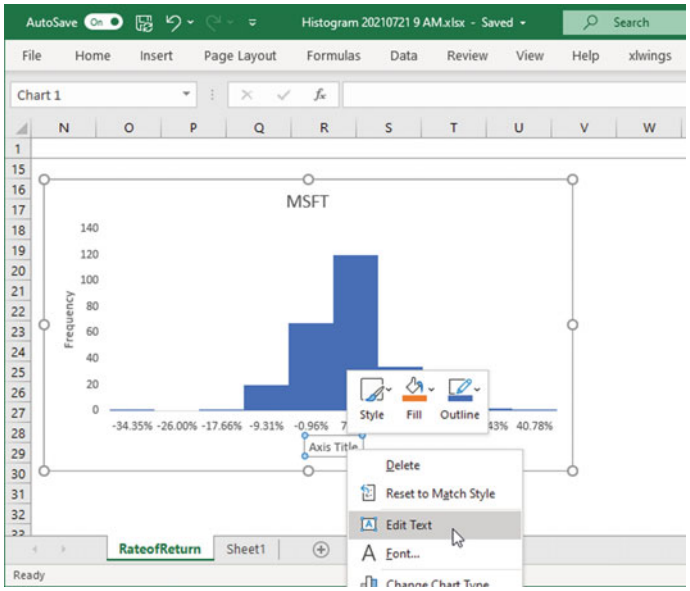
(1) Select the chart title.
(2) In the formula bar, enter the formula = *RateofReturn!$I$1*.



Axis



(1) Right-mouse click on the y-*Axis Title* label and select *Edit Text*.
(2) Change the label to *Frequency*.

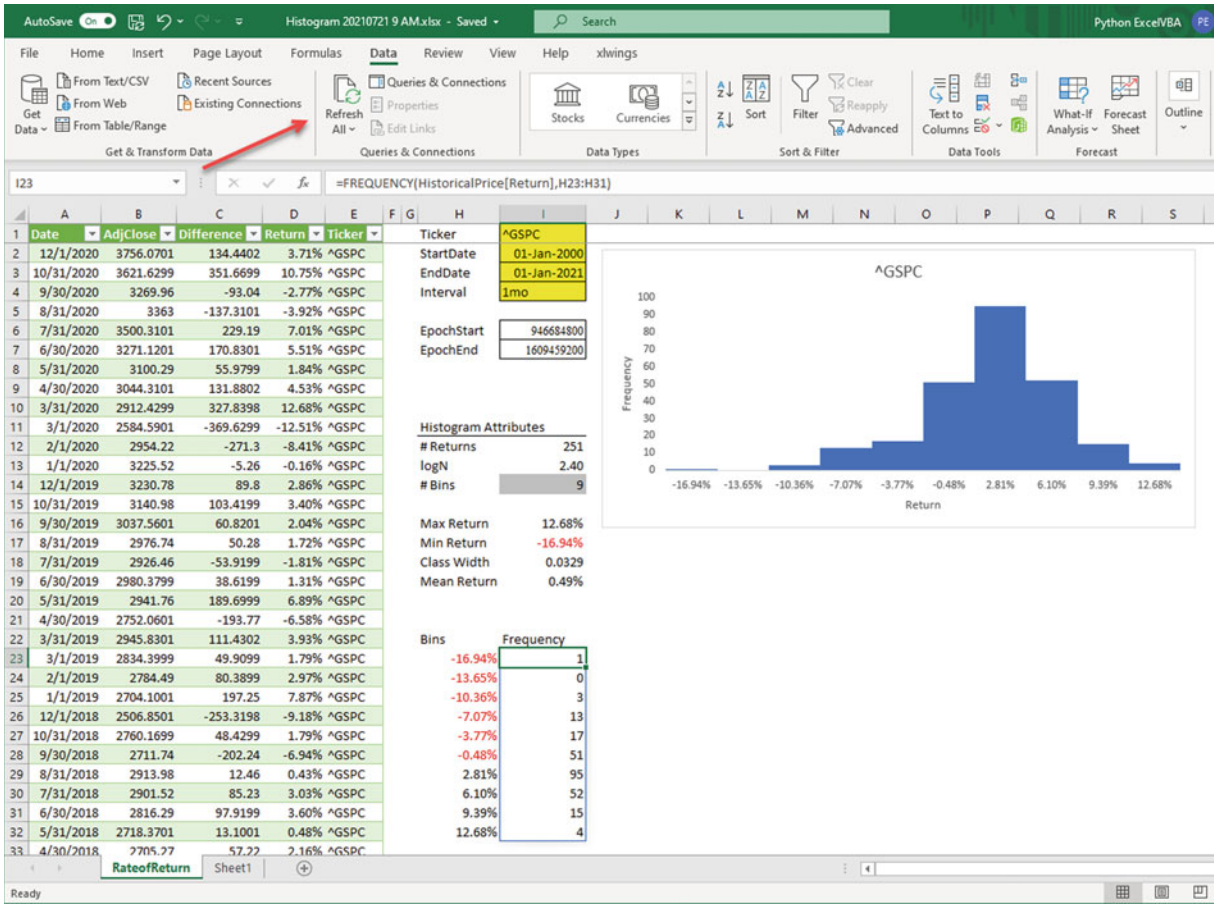


(3) Right-mouse click on the x-*Axis Title* label and select *Edit Text.*
(4) Change the label to *Return.*

3.3.2.3 Excel Histogram Template

We now have an Excel histogram template. We can now quickly create a histogram for any ticker for any date range. This is done by.

(1) Entering the ticker in cell I1.
(2) Entering the start date in cell I2.
(3) End date in cell I3.
(4) Clicking on *Refresh All* in the *Data* tab.



The ticker for the S&P500 in Yahoo! Finance is ^GSPC. Above is the S&P 500 rate of return histogram for January 1, 2000 to January 1, 2021.

3.4 Using Python to Create Johnson & Johnson's Rate of Return Histogram

In this section, we will use the *Python* library *yfinance* to retrieve the historical data of a ticker. The URL for the documentation of *yfinance* is https://aroussi.com/post/python-yahoo-finance.

3.4.1 Code

The Python code below developed using Visual Studio Code.

The code has a function call *SturgeRule* to calculate the number of intervals for a histogram based on the sampleSize. Line 43 saves the histogram in the same folder as the *Python* code.

C: > Demo > Chapter3 > Chapter3_Return_Histogram.py > graph_return

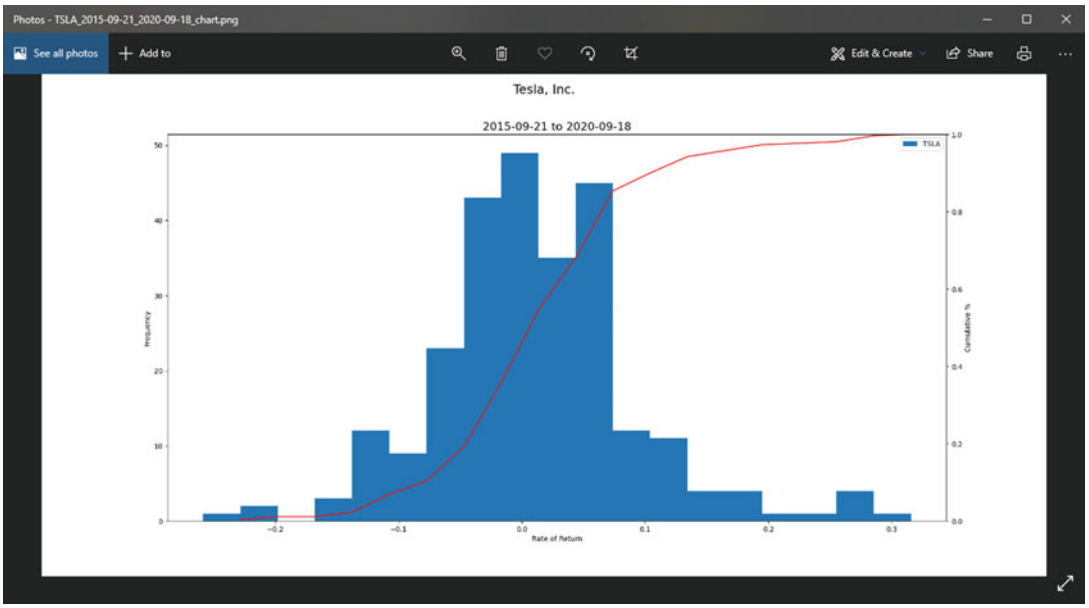
```
#Sep-19-2020 3PM
import yfinance as yf
import numpy as np, pandas as pd, datetime as dt, requests,json,os
import math, matplotlib.pyplot as plt,datetime as dt#get current date

#https://query1.finance.yahoo.com/v7/finance/quote?symbols=AAPL sample Yahoo! Finance API
def yahoo_TickerInfo(Ticker,Attribute):
    response = requests.get("https://query1.finance.yahoo.com/v7/finance/quote?symbols=" + Ticker)
    responseText = json.loads(response.text)
    quoteResponse = responseText["quoteResponse"]
    return(quoteResponse["result"][0][Attribute])

#Sturge's Rules determine the number of intervals for a histogram based on the sample size
def SturgeRule(sampleSize):
    kinterval = 1 + 3.3 * math.log(sampleSize)
    kinterval = round(kinterval)
    return( kinterval)

# this function is to fetch the stock info and data and then plot it
def graph_return(Ticker, time_interval, start_t, end_t):
  Data=yf.download(Ticker,start=start_t,end=end_t,interval = time_interval)
  Adjclose=Data['Adj Close']
  AdjcloseDr = Adjclose.dropna()
  R=AdjcloseDr.pct_change()[1:]

  fig = plt.figure(figsize=(20,10))

  #graph chart
  ax1 = fig.add_subplot(111)
  (counts, bins, patch)=ax1.hist(R, bins=SturgeRule(len(AdjcloseDr)))
  ax1.set_xlabel('Rate of Return')
  ax1.set_ylabel('Frequency')
  ax1.legend([Ticker])
  ax2 = ax1.twinx()  # set another Y axis with label
  pct_counts = counts/sum(counts) # calculate the percent change count
  Cum_sum = np.cumsum(pct_counts)# calculate cumulative percent change
  ax2.plot(bins[1:],Cum_sum,'r') # set the bins
  ax2.set_ylim([0, 1]) #set the scale of the right Y-axis
  ax2.set_ylabel('Cumulative %')
  plt.title(start_t+' to '+end_t,fontsize=16)#set font here
  plt.suptitle(yahoo_TickerInfo(Ticker,"longName") ,fontsize=18)
  print("")
  plt.savefig(Ticker + "_" + Start_Date + "_" + End_Date + "_chart.png")

  AdjcloseDr = AdjcloseDr.round(2) # AdjcloseDr is a series not a dataframe
  DataAdj = pd.concat([AdjcloseDr,R],axis=1)
  DataAdj.columns =['Adjclose','Rate of return']
  DataAdj = DataAdj.sort_values(by='Date', ascending=False)

  return DataAdj # return histogram data
```

```
#initialize dates
previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
years_ago = dt.date.today()-dt.timedelta(days=5 * 365)
Default_start = years_ago.strftime("%Y-%m-%d")

# request user input
# set default values if users does not enter a value
Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
Start_Date = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
End_Date = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
T_interval = input('Enter date interval(1d,1wk,1mo Default = 1wk)-->') or '1wk'

print("")

DataAdj = graph_return(Ticker, T_interval,Start_Date,End_Date)

print(DataAdj)
DataAdj.to_csv(Ticker + "_" + Start_Date + "_" + End_Date + '_histogram.csv')

print("")

print("Histrogram chart and data saved to " + os.getcwd())
```

3.4.2 Output

```
Python 3.8.5 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============ RESTART: C:\Demo\Chapter3\Chapter3_Return_Histogram.py ============
Please enter ticker(Default = JNJ)-->TSLA
Enter start date(Default = 5 Years ago, 2015-09-21)-->2015-09-21
Enter end date(Default = Yesterday, 2020-09-18)-->2020-09-18
Enter date interval(1d,1wk,1mo Default = 1wk)-->1wk

[**********************100%**********************]  1 of 1 completed

            Adjclose  Rate of return
Date
2020-09-14    442.15        0.186279
2020-09-07    372.72       -0.109007
2020-08-31    418.32       -0.055028
2020-08-24    442.68        0.079718
2020-08-17    410.00        0.241878
...              ...             ...
2015-10-19     41.82       -0.078939
2015-10-12     45.40        0.028637
2015-10-05     44.14       -0.108575
2015-09-28     49.51       -0.036355
2015-09-21     51.38             NaN

[261 rows x 2 columns]

Histrogram chart and data saved to C:\Demo\Chapter3
>>>
```

The TSLA histogram produced by *Python* is shown below:



3.5 Financial Statements

Financial accounting statements are used to understand a firm's financial condition. The three required financial accounting statements for publicly traded companies are the *balance sheet, income statement, and statement of cash flows.* We can use Yahoo! Finance to see a company's financial statement for the past 4 years.

The URL https://finance.yahoo.com/quote/AAPL/balance-sheet shows Apple Inc.'s balance sheet for the last 4 years.

The URL https://finance.yahoo.com/quote/AAPL/financials shows Apple Inc.'s income statement for the last 4 years.

The URL https://finance.yahoo.com/quote/AAPL/cash-flow shows Apple Inc.'s statement of cash flows for the last 4 years.

Once you know the link to these financial statements for one company, it is easy to get another company's financial statements.

The URL https://finance.yahoo.com/quote/WMT/balance-sheet shows Walmart's balance sheet for the last 4 years.

The URL https://finance.yahoo.com/quote/WMT/financials shows Walmart's income statement for the last 4 years.

The URL https://finance.yahoo.com/quote/WMT/cash-flows shows Walmart's income statement for the last 4 years.

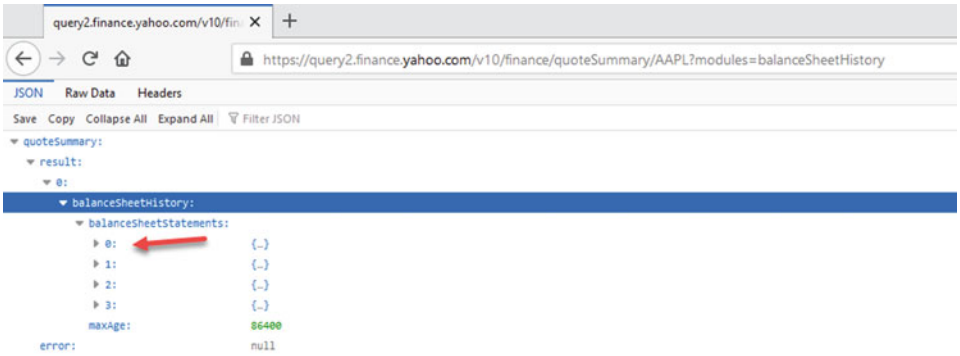
3.5.1 Power Query

3.5.1.1 Balance Sheet

On July 23, 2021, the URL https://finance.yahoo.com/quote/AAPL/balance-sheet showed AAPL's balance sheet for the past 4 years.

The following is the Yahoo! Finance API to retrieve AAPL's balance sheet:
https://query2.finance.yahoo.com/v10/finance/quoteSummary/AAPL?modules=balanceSheetHistory.

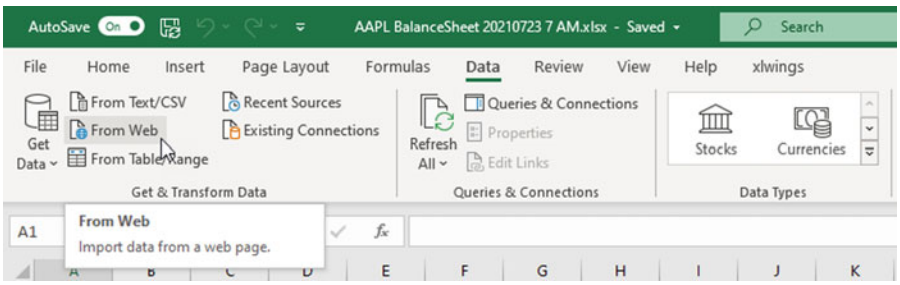
Reviewing the Yahoo! Finance API in Firefox will help guide us on how to set up *Power Query* to consume the Yahoo! Finance balance sheet API. The 4 years of balance sheets are under the element *balancesSheetstatements*. The child element *0* contains the balance sheet for the first year.



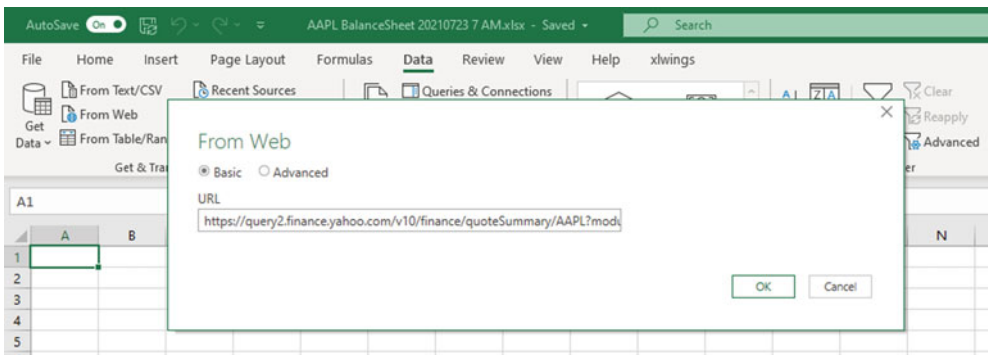
.

2020 Balance Sheet

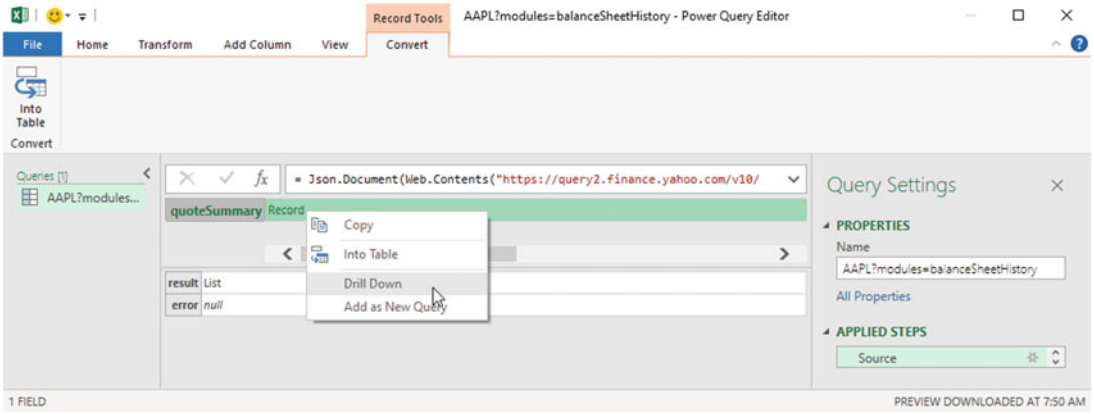
(1) On the *Data* tab click on *From Web*.



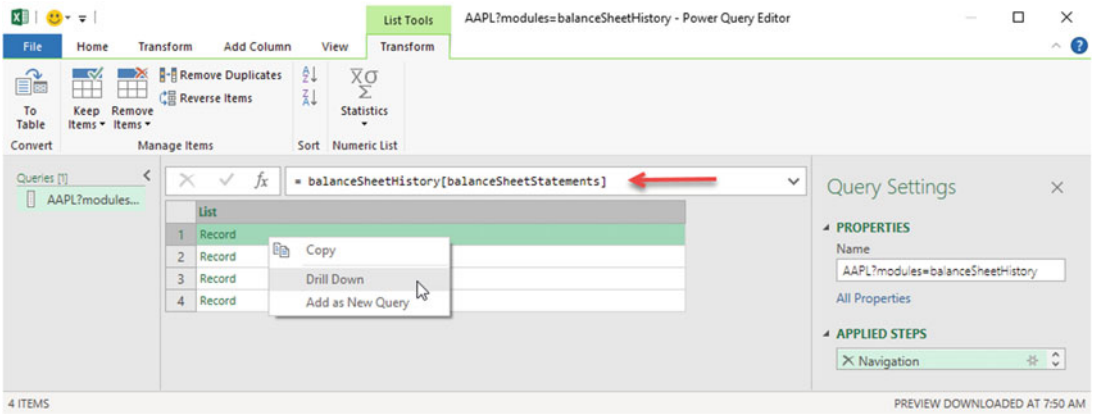
Enter APPL's API for the balance sheet.



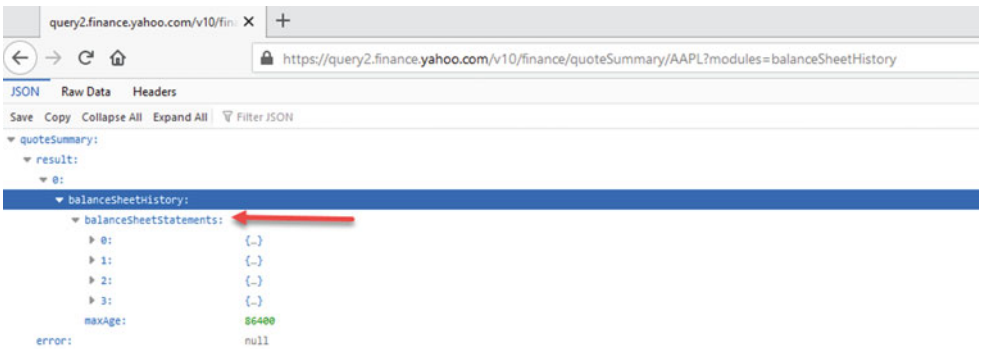
(2) Right-mouse click on *Record* and choose drill down.



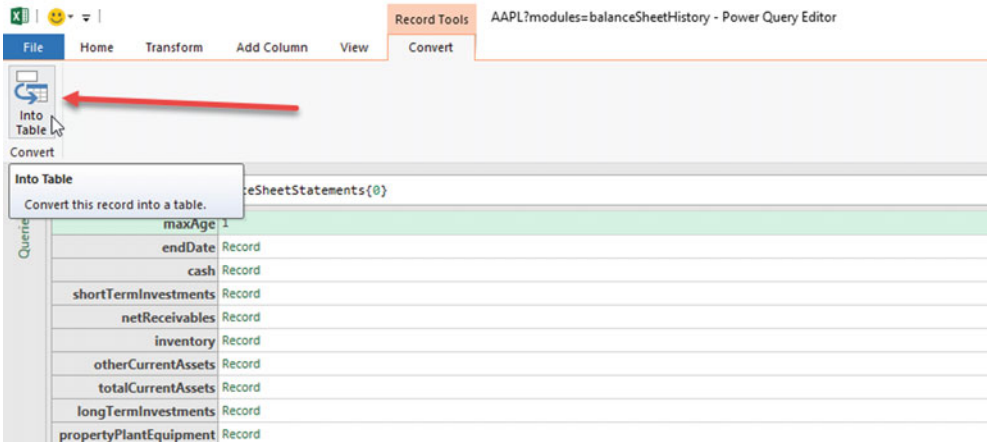
(3) Drill down until you reach the following screen. When you see this screen, drill down on the first record.



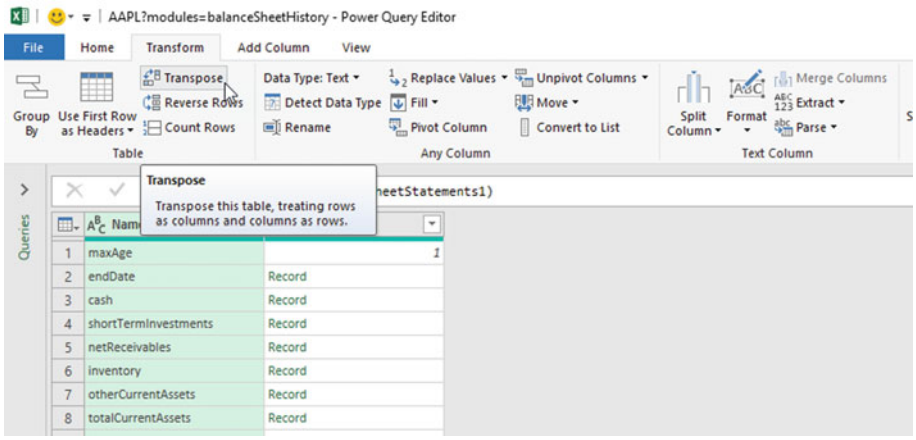
Note that *Power Query* mimics Firefox. Both has a *balancesheetstatements* that has four children.



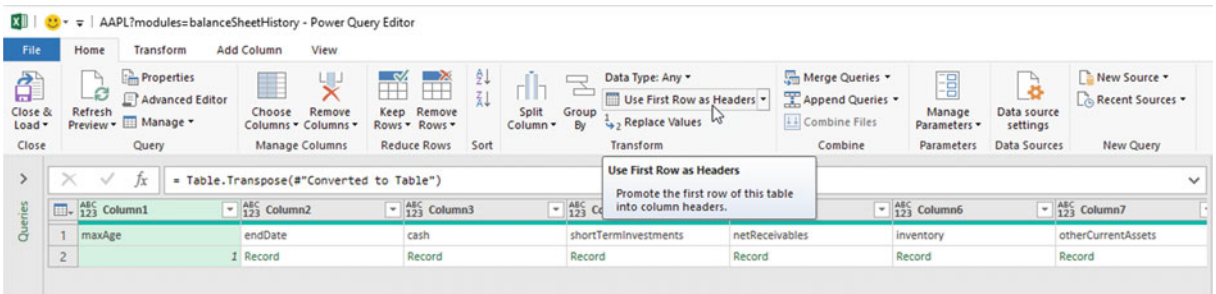
(4) Click *into Table.*



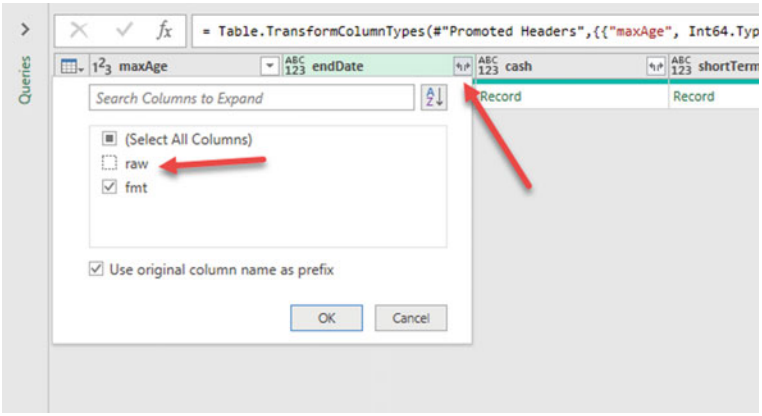
(5) Click on *Transpose* in the *Transform* tab.



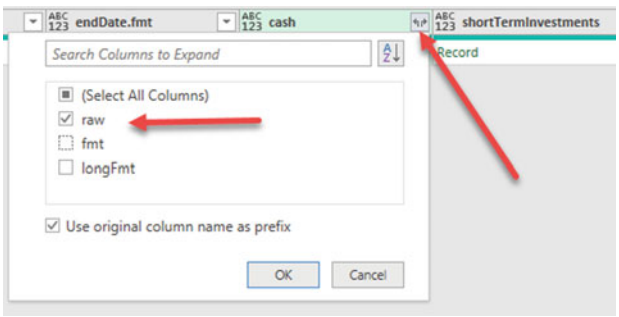
(6) Click on *Use First Row as Headers* in the *Home* tab.



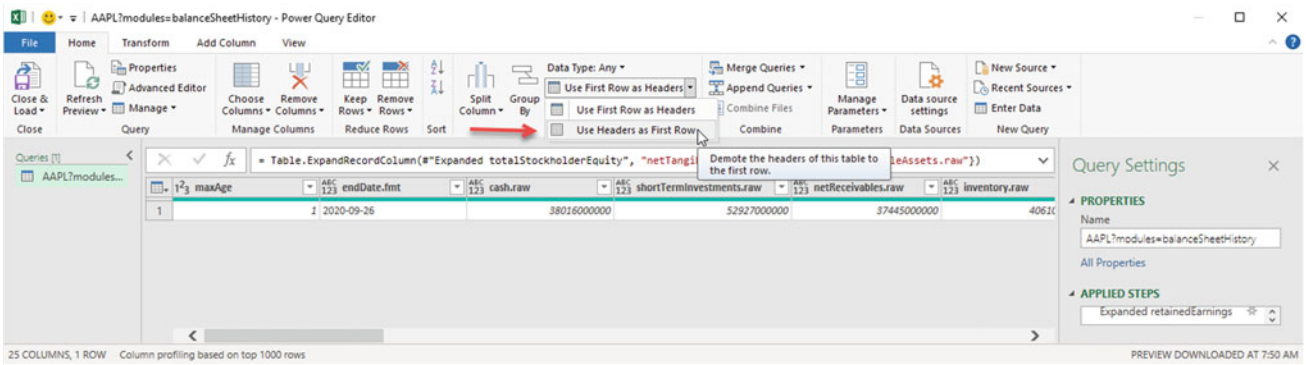
(7) Choose the expand button in the *endDate* field. Uncheck *raw*.



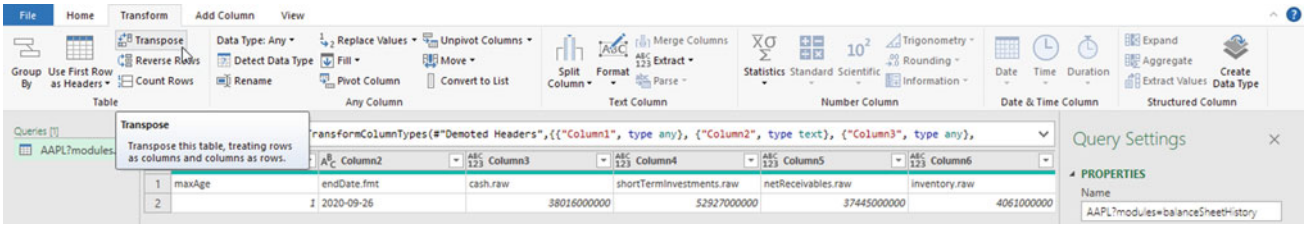
(8) Choose the expand button in the *cash* field. Keep *raw* checked and uncheck *fmt* and *longFmt.*



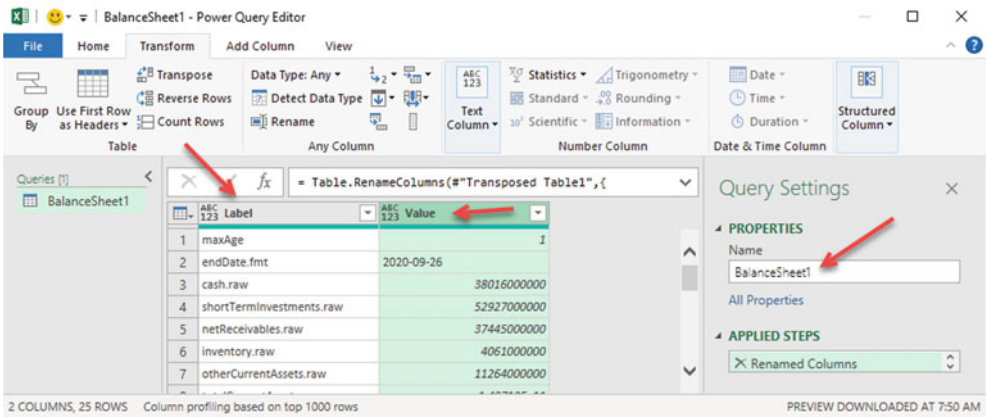
(9) For all the other fields, keep *raw* checked and uncheck *fmt* and *longFmt*.
(10) From the *Home* tab, click on *Use Headers as First Row*.



(11) From the *Transform* tab, click on *Transpose*.



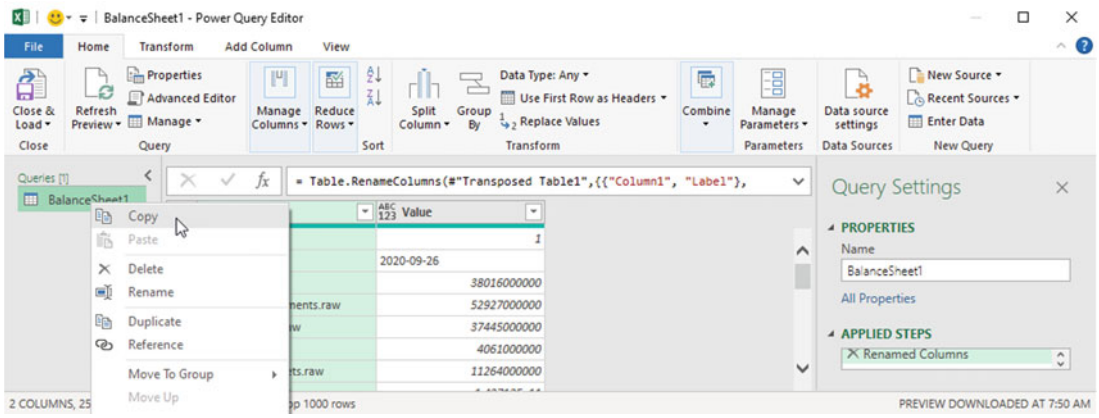
(12) Change query name to *BalanceSheet1*. Change first column name to *Label*. Change second column name to *Value*.



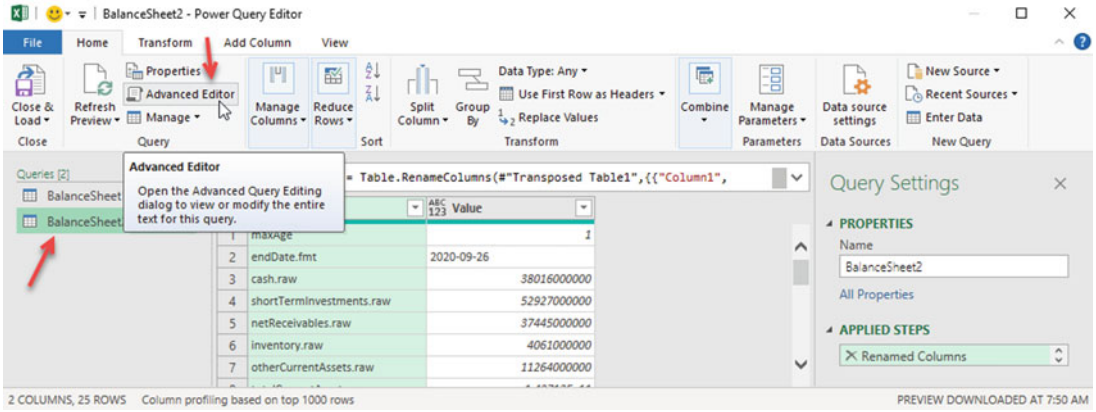
2019 Balance Sheet

Creating the query for the balance sheet for the year 2019 will be based on the query for the balance sheet for the year 2020.

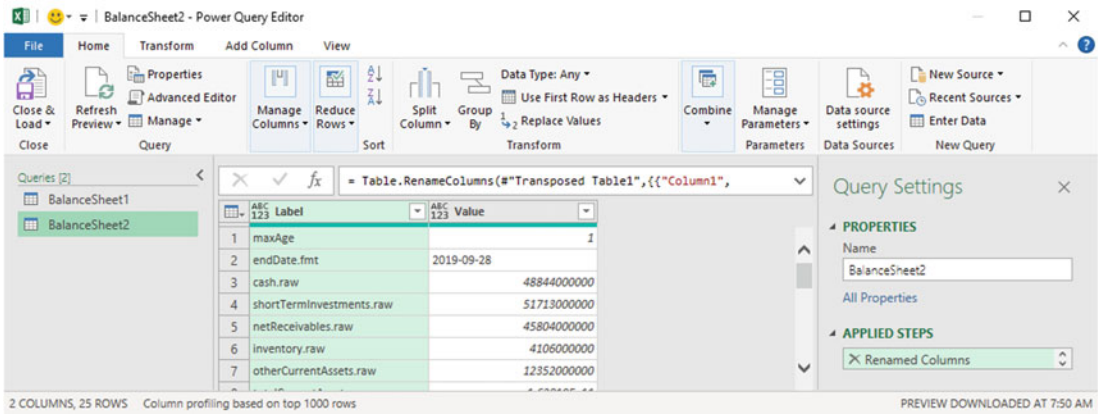
(1) Copy the query *BalanceSheet1* and rename it to *BalanceSheet2*.



With the query *BalanceSheet2* selected, click on *Advanced Editor*.



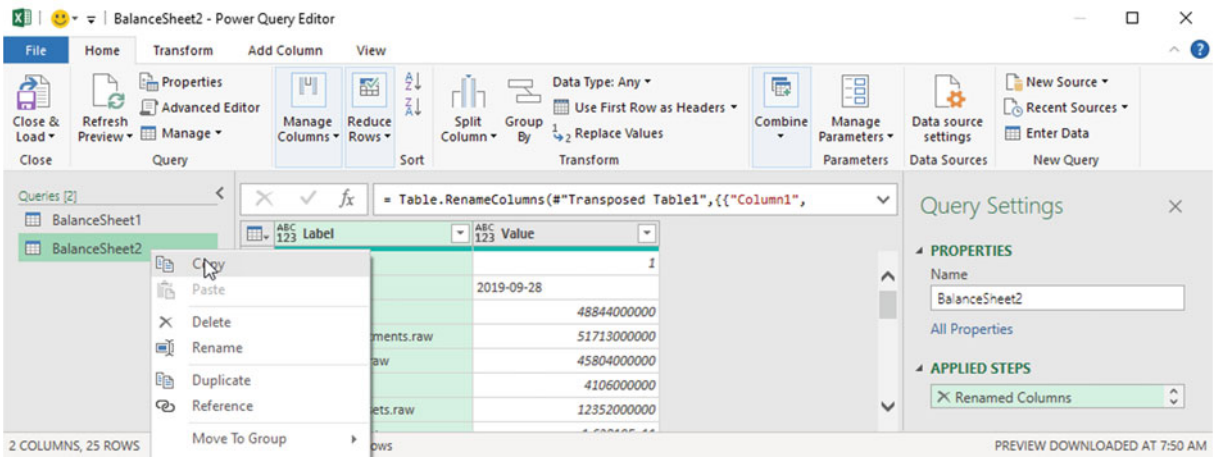
(2) In line 8 change *balanceSheetStatements{0}* to *balanceSheetStatements{1}*.



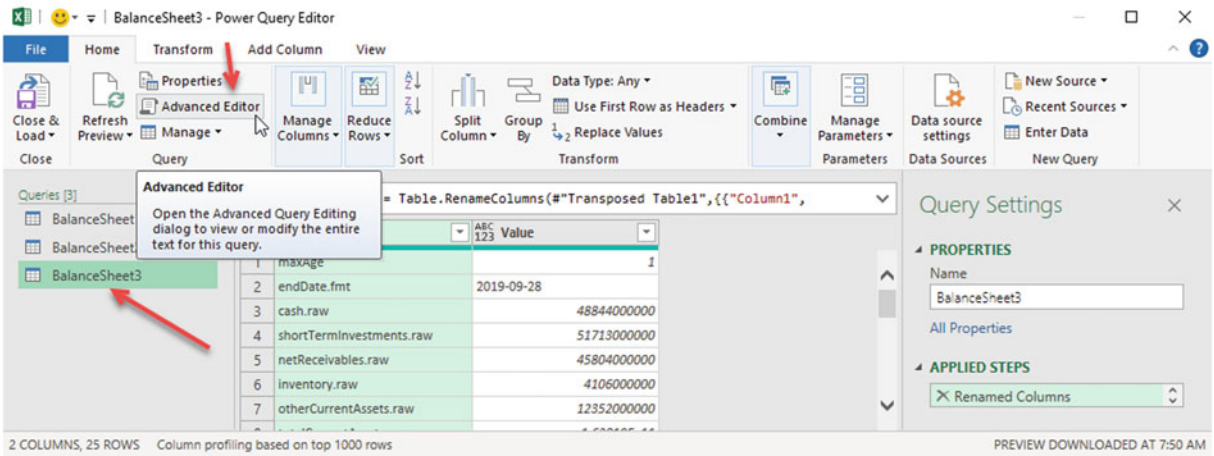
2018 Balance Sheet

Creating the query for the balance sheet for the year 2018 will be based on the query for the balance sheet for the year 2019.

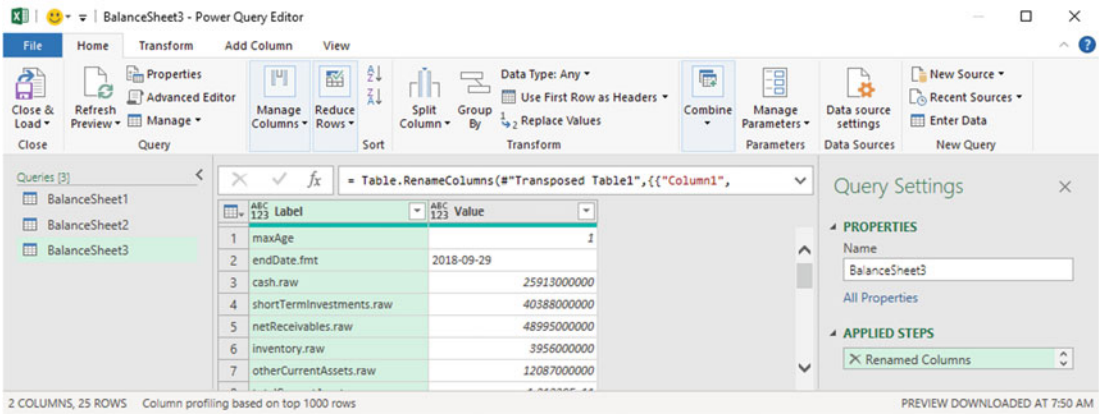
(1) Copy the query *BalanceSheet2* and rename it to *BalanceSheet3*.



(2) With the query *BalanceSheet3* selected, click on *Advanced Editor*.



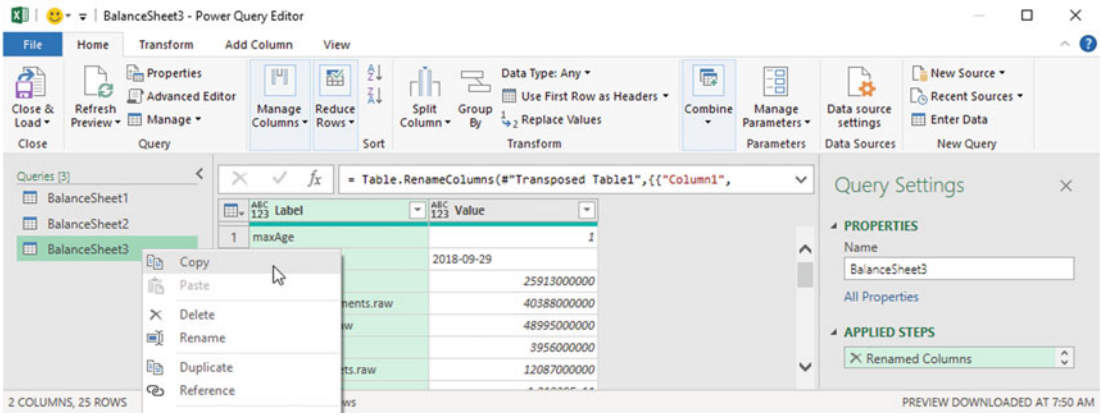
(3) In line 8 change *balanceSheetStatements{1}* to *balanceSheetStatements{2}*.



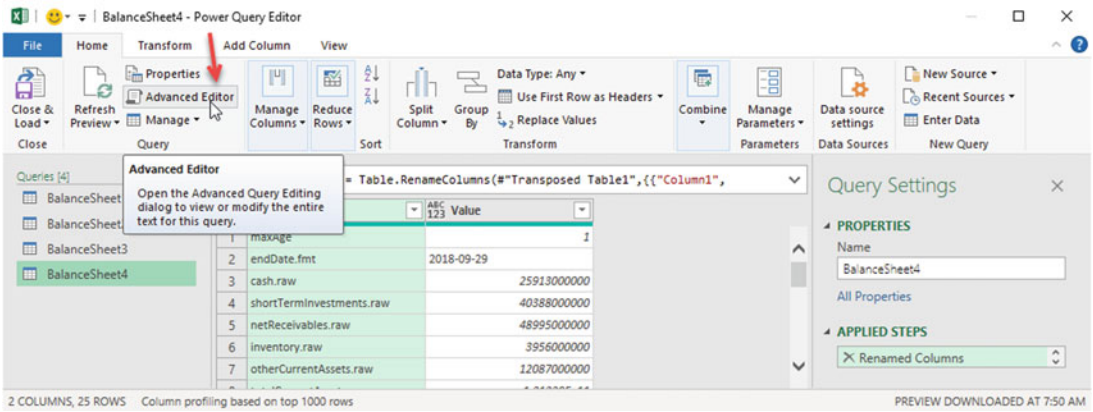
2017 Balance Sheet.

Creating the query for the balance sheet for the year 2017 will be based on the query for the balance sheet for the year 2018.

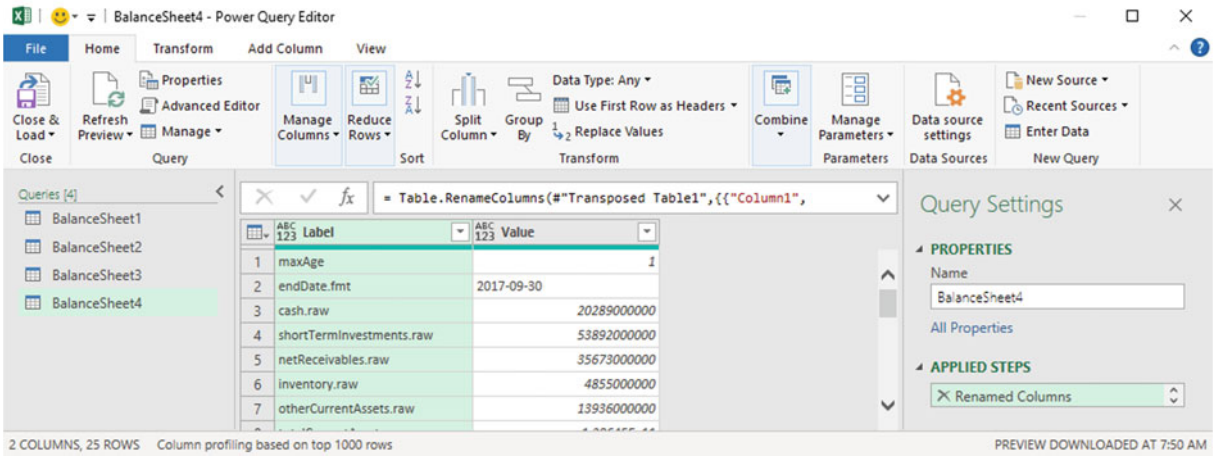
(1) Copy the query *BalanceSheet3* and rename it to *BalanceSheet4*.



(2) With the query *BalanceSheet4* selected, click on *Advanced Editor*.

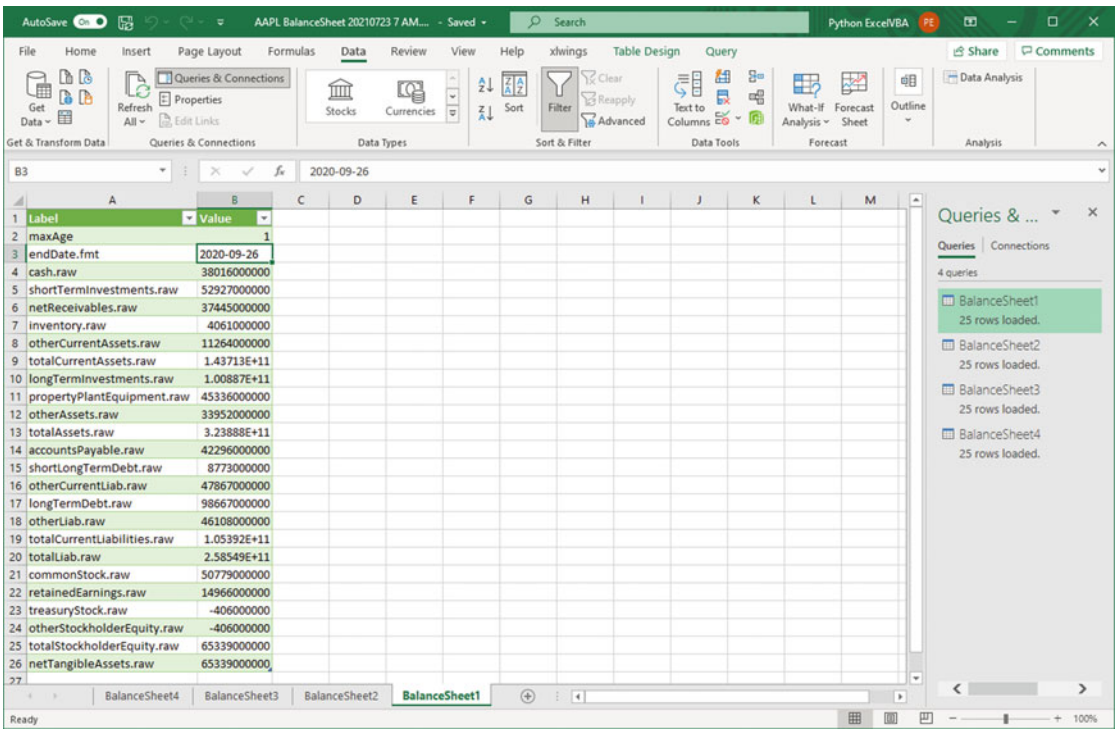


(3) In line 8 change *balanceSheetStatements{2}* to *balanceSheetStatements{3}*.



Combined Balance Sheet

APPL'S 2020 balance sheet is loaded to *BalanceSheet1*. APPL's 2019 balance sheet is loaded to *BalanceSheet2*. APPL's 2018 balance sheet is loaded to *BalanceSheet3*. APPL's 2017 balance sheet is loaded to *BalanceSheet4*.

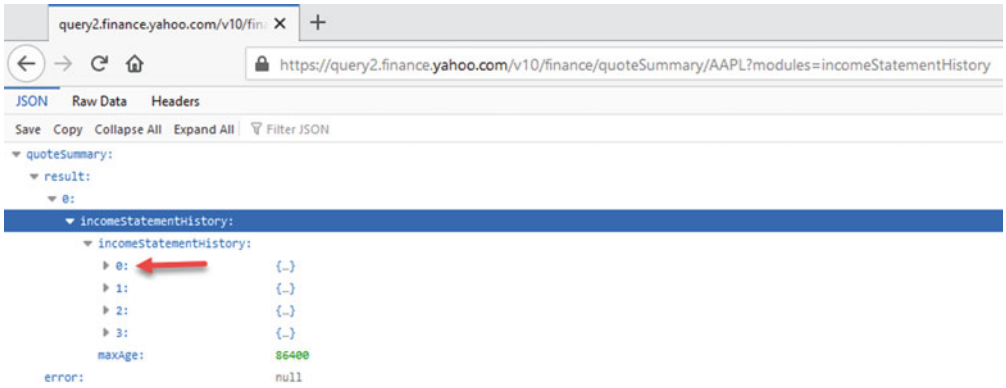


3.5.1.2 Income Statement

On July 23, 2021, the URL https://finance.yahoo.com/quote/AAPL/financials shows AAPL's income statement for the past 4 years.

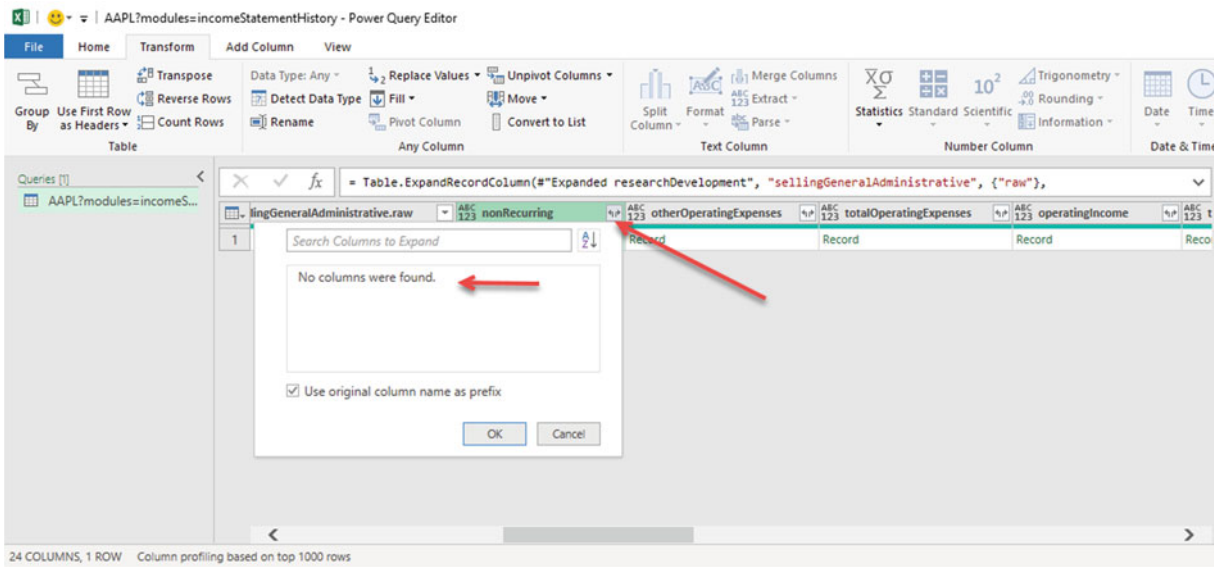
The following is the Yahoo! Finance API to retrieve AAPL's income statement:
https://query2.finance.yahoo.com/v10/finance/quoteSummary/AAPL?modules=incomeStatementHistory.

Reviewing the Yahoo! Finance income statement API in Firefox will help guide us on how to set up *Power Query* to consume the Yahoo! Finance income statement API. The 4 years of income statements are under the element *incomeStatementHistory*. The child element *0* contains the income statement for the first year.

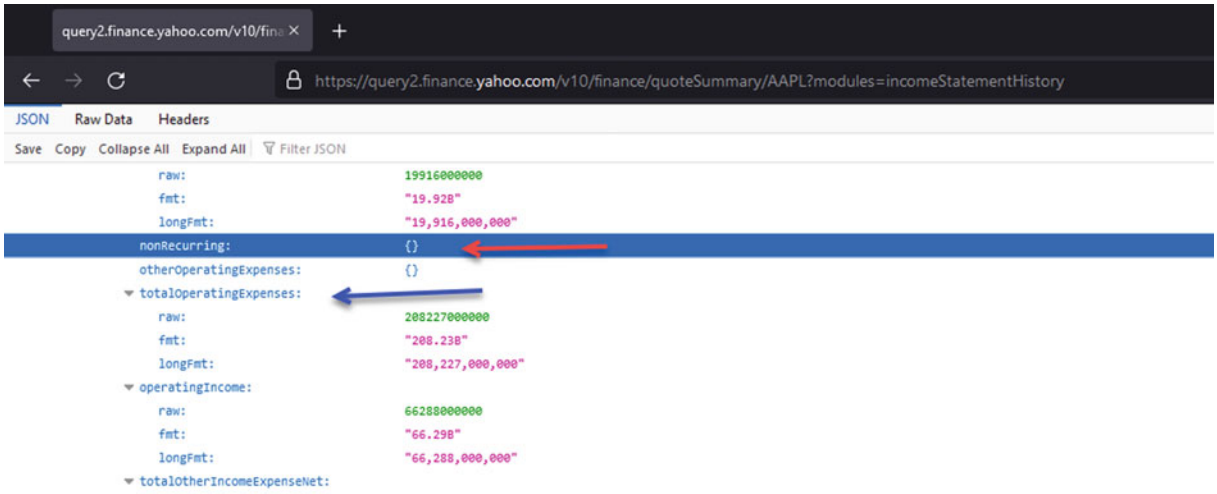


The concepts and approach to configuring *Power Query* to retrieve the income statement for AAPL are the same as the concepts and approach to configuring *Power Query* to retrieve the balance sheet for AAPL.

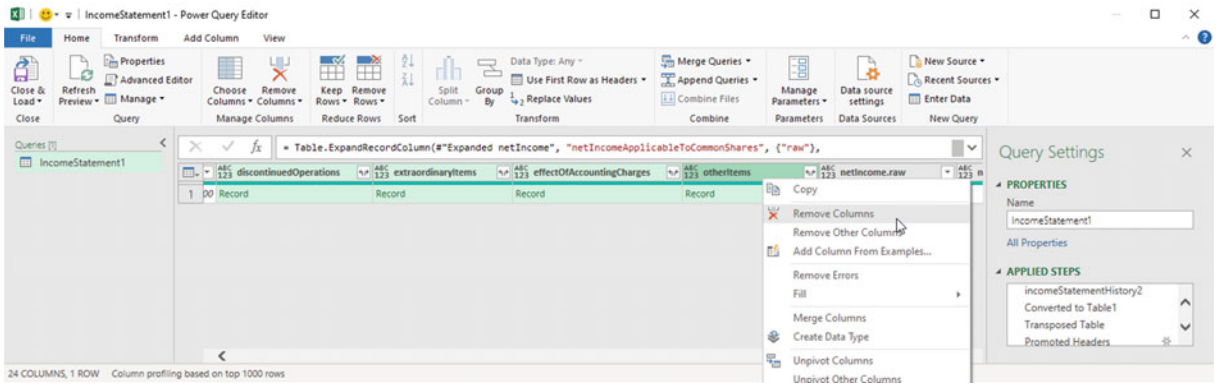
When configuring *Power Query* to retrieve the income statement, you will come across columns that you try to expand but have *No columns were found* issue.



For example, for the 2020 income statement for the column *nonRecurring* there were no columns found. This makes sense after reviewing the corresponding JSON response in Firefox.



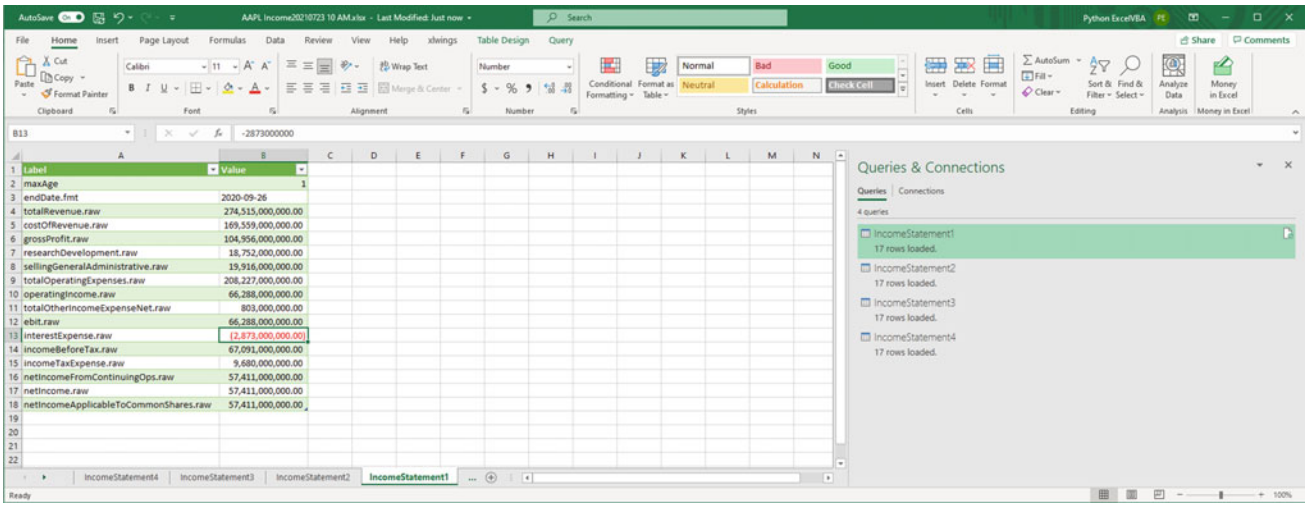
We can see that the *nonRecurring* element was empty while the *totalOperatingExpenses* element was not empty. We can remove all columns in *Power Query* that are empty.



Update line 8 in the advance editor to create the four different income statements.

Combined Income Statement

APPL'S 2020 income statement is loaded to *IncomeStatement1*. APPL's 2019 income statement is loaded to *IncomeStatement2*. APPL's 2018 income statement is loaded to *IncomeStatement3*. APPL's 2017 income statement is loaded to *IncomeStatement4*.

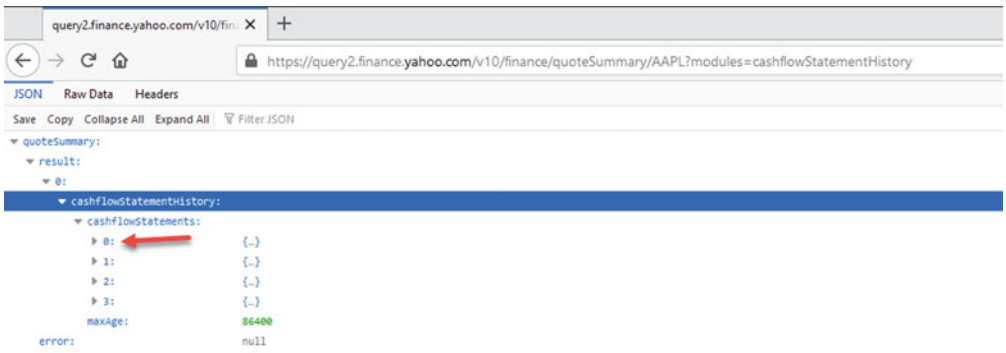


3.5.1.3 Cash Flow Statement

On July 23, 2021, the URL https://finance.yahoo.com/quote/AAPL/cash-flow shows AAPL's cash flow statement for the past 4 years.

The following is the Yahoo! Finance API to retrieve AAPL's cash flow statement:
https://query2.finance.yahoo.com/v10/finance/quoteSummary/AAPL?modules=cashflowStatementHistory.

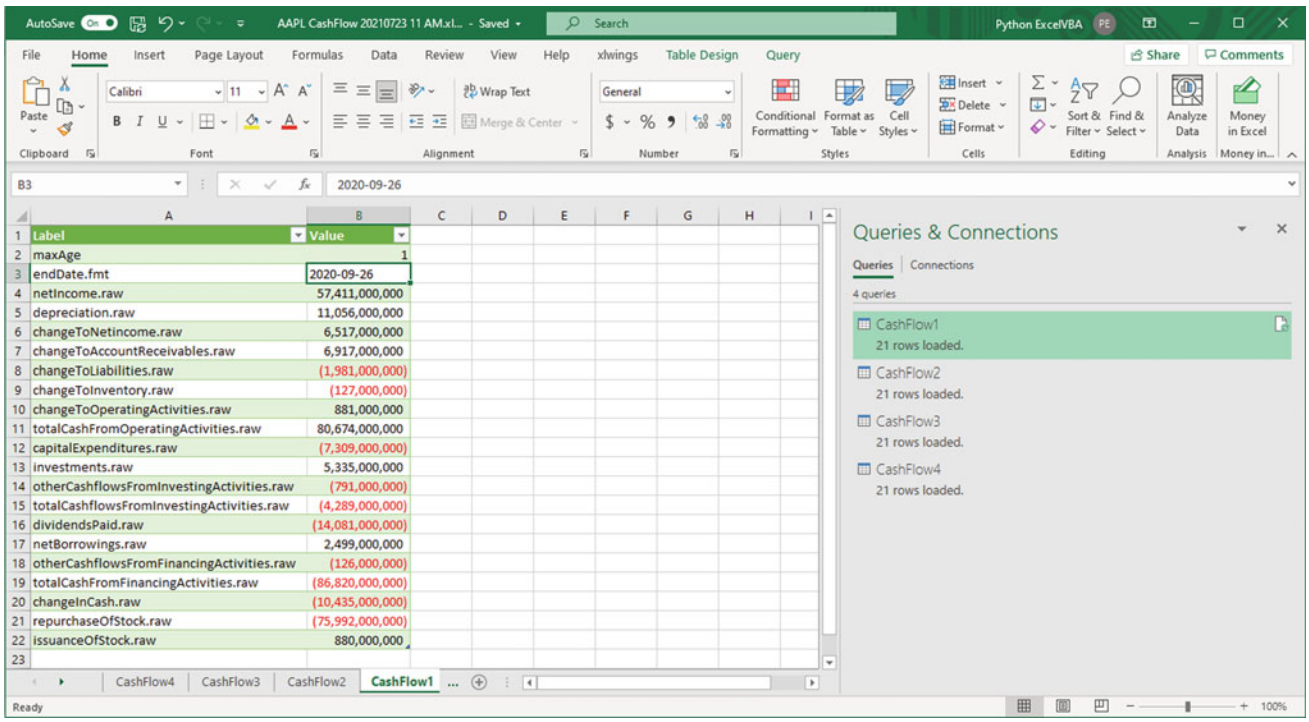
Reviewing the Yahoo! Finance cash flow statement API in Firefox will help guide us on how to set up *Power Query* to consume the Yahoo! Finance cash flow API. The 4 years of cash flow statements are under the element *cashflowstatements.* The child element *0* contains the cash flow statement for the first year.



Update line 8 in the advance editor to create the four different cash flow statements.

Combined Cash Flow Statement

APPL'S 2020 cash flow statement is loaded to *CashFlow1*. APPL's 2019 cash flow statement is loaded to *CashFlow2*. APPL's 2018 cash flow statement is loaded to *CashFlow3*. APPL's 2017 cash flow statement is loaded to *CashFlow4*.



3.5.2 Python

The Python code below was developed in Visual Studio Code.

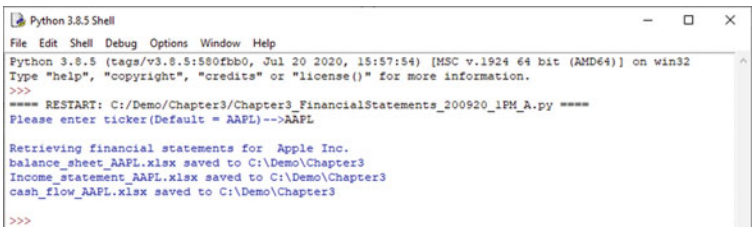
3.5.2.1 Code

Chapter3_FinancialStatements_200920_12PM_C.py

```
#Date: Sep-20-2020 12PM
from yahoo_fin.stock_info import get_financials # get all financials at once
import pandas as pd, json, requests, os # library used to find path

def yahoo_TickerInfo(Ticker,Attribute):
    response = requests.get("https://query1.finance.yahoo.com/v7/finance/quote?symbols=" + Ticker)
    responseText = json.loads(response.text)
    quoteResponse = responseText["quoteResponse"]
    return(quoteResponse["result"][0][Attribute])

#use python library xlsxwriter to format excel data columns
def saveToExcel(financialType,df_financial):
    # Create a Pandas Excel writer using XlsxWriter as the engine.
    # set default date format
    writer = pd.ExcelWriter(financialType + '_' + Ticker + '.xlsx', engine='xlsxwriter',
        datetime_format='mmm-d-yyyy')

    # Convert the dataframe to an XlsxWriter Excel object.
    df_financial.to_excel(writer, sheet_name=financialType,index=True)

    # Get the xlsxwriter workbook and worksheet objects.
    workbook  = writer.book
    worksheet = writer.sheets[financialType]

    #Add cell formatting to workbook
    formatComma = workbook.add_format({'num_format': '#,##0;[Red](#,##0)'})

    worksheet.set_column('A:A',37)
    worksheet.set_column('B:F', 17,formatComma)
    worksheet.set_column('G:G', 8)
    worksheet.set_column('H:H', 25)

    # Close the Pandas Excel writer and output the Excel file.
    writer.save()

def financialstatement(financialtype,finStatements):
    statement = finStatements['yearly_' + financialtype ]
    statement['type']=financialtype
    statement['ticker']=Ticker
    statement['longName'] = TickerLongName
    return statement # return modified data frame

#***************************** Main ***********************************************************
# request user input
# set default values if users does not enter a value
Ticker = input('Please enter ticker(Default = AAPL)-->') or 'AAPL'
TickerLongName = yahoo_TickerInfo(Ticker,"longName")

print("")
print ("Retrieving financial statements for " +TickerLongName )

#retrieves all financial statements at once
financials =get_financials(Ticker, yearly = True, quarterly = False)

saveToExcel('balance_sheet',financialstatement('balance_sheet',financials))
print('balance_sheet_' + Ticker + '.xlsx' + ' saved to ' + os.getcwd())

saveToExcel('income_statement',financialstatement('income_statement',financials))
print('Income_statement_' + Ticker + '.xlsx' + ' saved to ' + os.getcwd())

saveToExcel('cash_flow',financialstatement('cash_flow',financials))
print('cash_flow_' + Ticker + '.xlsx' + ' saved to ' + os.getcwd())

```

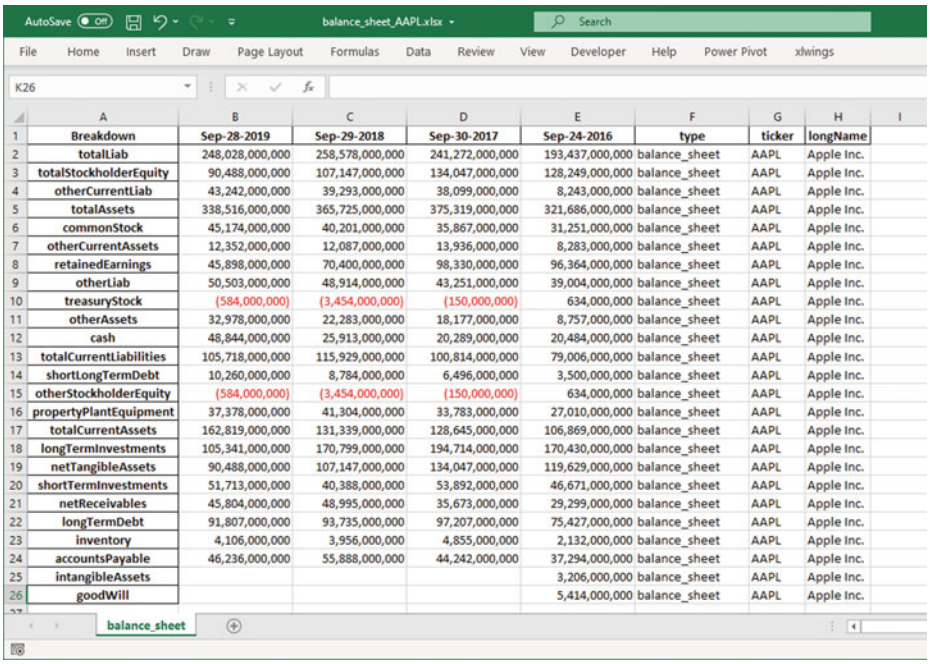
Readers should start reading the *Python* code with line 46. Line 53 of the above code retrieves the three standard financial statements and stores it in the DataFrame variable called "financials."

3.5.2.2 Output



The three financial statements are saved in its own Microsoft Excel workbook.

Balance Sheet



AutoSave Off balance_sheet_AAPL.xlsx Search

File Home Insert Draw Page Layout Formulas Data Review View Developer Help Power Pivot xlwings

K26

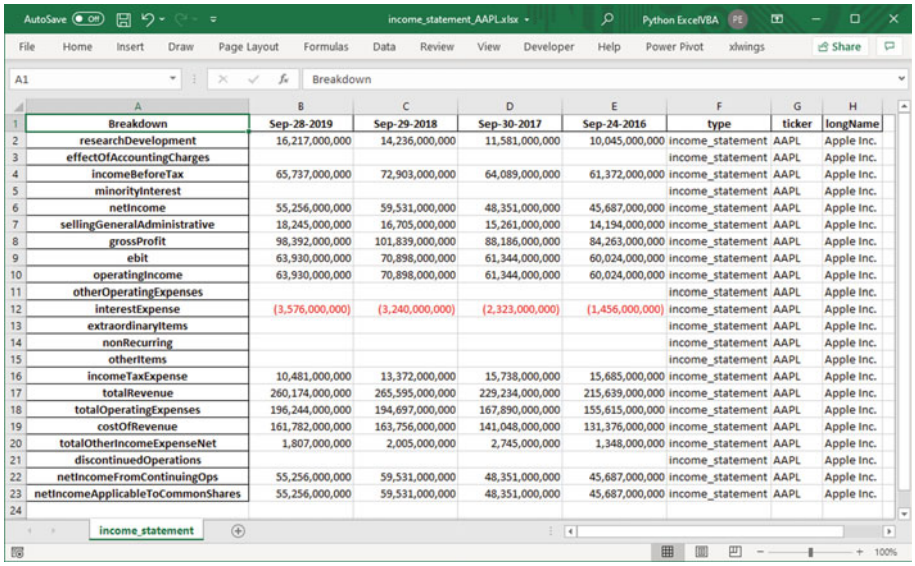
	A	B	C	D	E	F	G	H	I
1	Breakdown	Sep-28-2019	Sep-29-2018	Sep-30-2017	Sep-24-2016	type	ticker	longName	
2	totalLiab	248,028,000,000	258,578,000,000	241,272,000,000	193,437,000,000	balance_sheet	AAPL	Apple Inc.	
3	totalStockholderEquity	90,488,000,000	107,147,000,000	134,047,000,000	128,249,000,000	balance_sheet	AAPL	Apple Inc.	
4	otherCurrentLiab	43,242,000,000	39,293,000,000	38,099,000,000	8,243,000,000	balance_sheet	AAPL	Apple Inc.	
5	totalAssets	338,516,000,000	365,725,000,000	375,319,000,000	321,686,000,000	balance_sheet	AAPL	Apple Inc.	
6	commonStock	45,174,000,000	40,201,000,000	35,867,000,000	31,251,000,000	balance_sheet	AAPL	Apple Inc.	
7	otherCurrentAssets	12,352,000,000	12,087,000,000	13,936,000,000	8,283,000,000	balance_sheet	AAPL	Apple Inc.	
8	retainedEarnings	45,898,000,000	70,400,000,000	98,330,000,000	96,364,000,000	balance_sheet	AAPL	Apple Inc.	
9	otherLiab	50,503,000,000	48,914,000,000	43,251,000,000	39,004,000,000	balance_sheet	AAPL	Apple Inc.	
10	treasuryStock	(584,000,000)	(3,454,000,000)	(150,000,000)	634,000,000	balance_sheet	AAPL	Apple Inc.	
11	otherAssets	32,978,000,000	22,283,000,000	18,177,000,000	8,757,000,000	balance_sheet	AAPL	Apple Inc.	
12	cash	48,844,000,000	25,913,000,000	20,289,000,000	20,484,000,000	balance_sheet	AAPL	Apple Inc.	
13	totalCurrentLiabilities	105,718,000,000	115,929,000,000	100,814,000,000	79,006,000,000	balance_sheet	AAPL	Apple Inc.	
14	shortLongTermDebt	10,260,000,000	8,784,000,000	6,496,000,000	3,500,000,000	balance_sheet	AAPL	Apple Inc.	
15	otherStockholderEquity	(584,000,000)	(3,454,000,000)	(150,000,000)	634,000,000	balance_sheet	AAPL	Apple Inc.	
16	propertyPlantEquipment	37,378,000,000	41,304,000,000	33,783,000,000	27,010,000,000	balance_sheet	AAPL	Apple Inc.	
17	totalCurrentAssets	162,819,000,000	131,339,000,000	128,645,000,000	106,869,000,000	balance_sheet	AAPL	Apple Inc.	
18	longTermInvestments	105,341,000,000	170,799,000,000	194,714,000,000	170,430,000,000	balance_sheet	AAPL	Apple Inc.	
19	netTangibleAssets	90,488,000,000	107,147,000,000	134,047,000,000	119,629,000,000	balance_sheet	AAPL	Apple Inc.	
20	shortTermInvestments	51,713,000,000	40,388,000,000	53,892,000,000	46,671,000,000	balance_sheet	AAPL	Apple Inc.	
21	netReceivables	45,804,000,000	48,995,000,000	35,673,000,000	29,299,000,000	balance_sheet	AAPL	Apple Inc.	
22	longTermDebt	91,807,000,000	93,735,000,000	97,207,000,000	75,427,000,000	balance_sheet	AAPL	Apple Inc.	
23	inventory	4,106,000,000	3,956,000,000	4,855,000,000	2,132,000,000	balance_sheet	AAPL	Apple Inc.	
24	accountsPayable	46,236,000,000	55,888,000,000	44,242,000,000	37,294,000,000	balance_sheet	AAPL	Apple Inc.	
25	intangibleAssets				3,206,000,000	balance_sheet	AAPL	Apple Inc.	
26	goodWill				5,414,000,000	balance_sheet	AAPL	Apple Inc.	

balance_sheet

Above is the balance sheet for AAPL in Microsoft Excel.

We can go to https://finance.yahoo.com/quote/AAPL/balance-sheet to confirm that the balance sheet retrieved by the *Python* library *yahoo_fin* is correct.

Income Statement



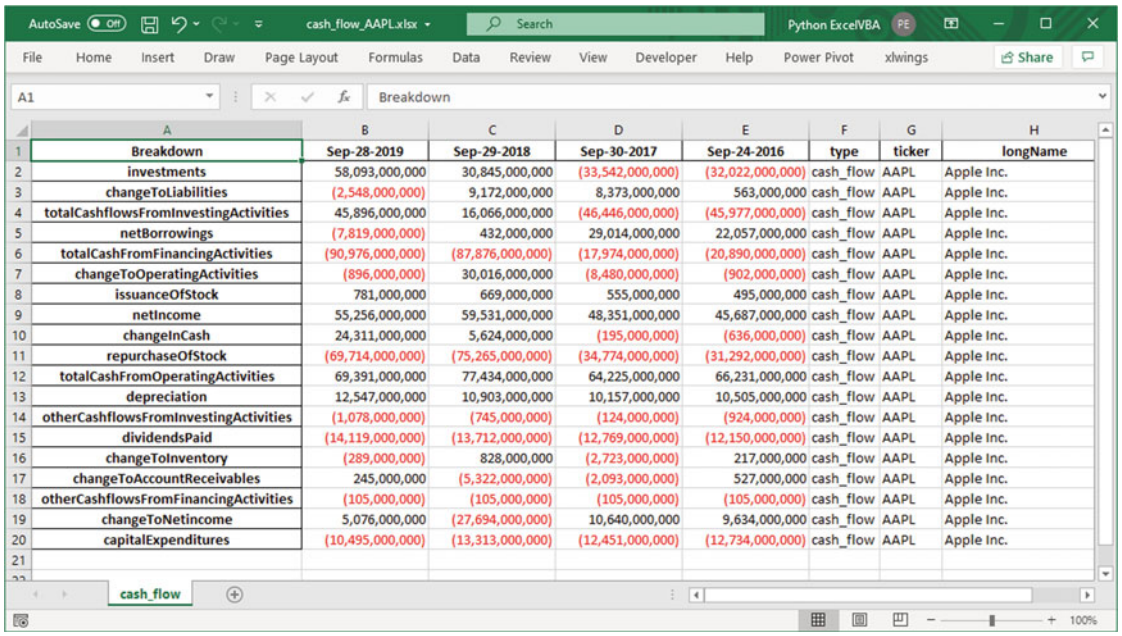
Breakdown	Sep-28-2019	Sep-29-2018	Sep-30-2017	Sep-24-2016	type	ticker	longName
researchDevelopment	16,217,000,000	14,236,000,000	11,581,000,000	10,045,000,000	income_statement	AAPL	Apple Inc.
effectOfAccountingCharges					income_statement	AAPL	Apple Inc.
incomeBeforeTax	65,737,000,000	72,903,000,000	64,089,000,000	61,372,000,000	income_statement	AAPL	Apple Inc.
minorityInterest					income_statement	AAPL	Apple Inc.
netIncome	55,256,000,000	59,531,000,000	48,351,000,000	45,687,000,000	income_statement	AAPL	Apple Inc.
sellingGeneralAdministrative	18,245,000,000	16,705,000,000	15,261,000,000	14,194,000,000	income_statement	AAPL	Apple Inc.
grossProfit	98,392,000,000	101,839,000,000	88,186,000,000	84,263,000,000	income_statement	AAPL	Apple Inc.
ebit	63,930,000,000	70,898,000,000	61,344,000,000	60,024,000,000	income_statement	AAPL	Apple Inc.
operatingIncome	63,930,000,000	70,898,000,000	61,344,000,000	60,024,000,000	income_statement	AAPL	Apple Inc.
otherOperatingExpenses					income_statement	AAPL	Apple Inc.
interestExpense	(3,576,000,000)	(3,240,000,000)	(2,323,000,000)	(1,456,000,000)	income_statement	AAPL	Apple Inc.
extraordinaryItems					income_statement	AAPL	Apple Inc.
nonRecurring					income_statement	AAPL	Apple Inc.
otherItems					income_statement	AAPL	Apple Inc.
incomeTaxExpense	10,481,000,000	13,372,000,000	15,738,000,000	15,685,000,000	income_statement	AAPL	Apple Inc.
totalRevenue	260,174,000,000	265,595,000,000	229,234,000,000	215,639,000,000	income_statement	AAPL	Apple Inc.
totalOperatingExpenses	196,244,000,000	194,697,000,000	167,890,000,000	155,615,000,000	income_statement	AAPL	Apple Inc.
costOfRevenue	161,782,000,000	163,756,000,000	141,048,000,000	131,376,000,000	income_statement	AAPL	Apple Inc.
totalOtherIncomeExpenseNet	1,807,000,000	2,005,000,000	2,745,000,000	1,348,000,000	income_statement	AAPL	Apple Inc.
discontinuedOperations					income_statement	AAPL	Apple Inc.
netIncomeFromContinuingOps	55,256,000,000	59,531,000,000	48,351,000,000	45,687,000,000	income_statement	AAPL	Apple Inc.
netIncomeApplicableToCommonShares	55,256,000,000	59,531,000,000	48,351,000,000	45,687,000,000	income_statement	AAPL	Apple Inc.

Above is the income statement retrieved by *Python*.

We can go to https://finance.yahoo.com/quote/AAPL/financials to confirm that the income statement retrieved by the *Python* library *yahoo_fin* is correct.

Cash Flow

Below is the cash flow statement retrieved by *Python*:



Breakdown	Sep-28-2019	Sep-29-2018	Sep-30-2017	Sep-24-2016	type	ticker	longName
investments	58,093,000,000	30,845,000,000	(33,542,000,000)	(32,022,000,000)	cash_flow	AAPL	Apple Inc.
changeToLiabilities	(2,548,000,000)	9,172,000,000	8,373,000,000	563,000,000	cash_flow	AAPL	Apple Inc.
totalCashflowsFromInvestingActivities	45,896,000,000	16,066,000,000	(46,446,000,000)	(45,977,000,000)	cash_flow	AAPL	Apple Inc.
netBorrowings	(7,819,000,000)	432,000,000	29,014,000,000	22,057,000,000	cash_flow	AAPL	Apple Inc.
totalCashFromFinancingActivities	(90,976,000,000)	(87,876,000,000)	(17,974,000,000)	(20,890,000,000)	cash_flow	AAPL	Apple Inc.
changeToOperatingActivities	(896,000,000)	30,016,000,000	(8,480,000,000)	(902,000,000)	cash_flow	AAPL	Apple Inc.
issuanceOfStock	781,000,000	669,000,000	555,000,000	495,000,000	cash_flow	AAPL	Apple Inc.
netIncome	55,256,000,000	59,531,000,000	48,351,000,000	45,687,000,000	cash_flow	AAPL	Apple Inc.
changeInCash	24,311,000,000	5,624,000,000	(195,000,000)	(636,000,000)	cash_flow	AAPL	Apple Inc.
repurchaseOfStock	(69,714,000,000)	(75,265,000,000)	(34,774,000,000)	(31,292,000,000)	cash_flow	AAPL	Apple Inc.
totalCashFromOperatingActivities	69,391,000,000	77,434,000,000	64,225,000,000	66,231,000,000	cash_flow	AAPL	Apple Inc.
depreciation	12,547,000,000	10,903,000,000	10,157,000,000	10,505,000,000	cash_flow	AAPL	Apple Inc.
otherCashflowsFromInvestingActivities	(1,078,000,000)	(745,000,000)	(124,000,000)	(924,000,000)	cash_flow	AAPL	Apple Inc.
dividendsPaid	(14,119,000,000)	(13,712,000,000)	(12,769,000,000)	(12,150,000,000)	cash_flow	AAPL	Apple Inc.
changeToInventory	(289,000,000)	828,000,000	(2,723,000,000)	217,000,000	cash_flow	AAPL	Apple Inc.
changeToAccountReceivables	245,000,000	(5,322,000,000)	(2,093,000,000)	527,000,000	cash_flow	AAPL	Apple Inc.
otherCashflowsFromFinancingActivities	(105,000,000)	(105,000,000)	(105,000,000)	(105,000,000)	cash_flow	AAPL	Apple Inc.
changeToNetincome	5,076,000,000	(27,694,000,000)	10,640,000,000	9,634,000,000	cash_flow	AAPL	Apple Inc.
capitalExpenditures	(10,495,000,000)	(13,313,000,000)	(12,451,000,000)	(12,734,000,000)	cash_flow	AAPL	Apple Inc.

We can go to https://finance.yahoo.com/quote/AAPL/cash-flow to confirm that income statement retrieved by the *Python* library *yahoo_fin* is correct.

3.6 Summary

In this chapter, we demonstrate techniques in both Excel and Python on how to automate the retrieval of market data and calculate the rate of returns. We also demonstrate how to automate the retrieval of financial statements from Yahoo! Finance.

We used histograms to display the rate of returns to provide information about the shape and form of the distribution. We looked at the histogram of the rate of return of Johnson & Johnson.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. Excel 2019 Power Programming With Vba. Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. Python for Professionals: Learning Python as a Second Language. BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media, 2017.

4 Numerical Summary Measures on Rate of Returns of Stocks and Market Indexes

4.1 Introduction

Statistics is about getting information from a data set. What type of information should we get from a data set? Statisticians tell us two of the most important information to get from a data set are *central location* and *variability* or *spread* of a data set. In addition to these characteristics of a data set, it is useful to get the *shape* of a data set. It is also useful to understand the *relative standing* of a particular data item relative to the other data items of the data set. When analyzing a data set, a frequent interest is to see if there is a *linear relationship* in the data set.

To demonstrate these concepts, we will calculate these summary measures of the rates of return of the S&P500, Walmart, Target, Microsoft, and Oracle. We will look at the monthly rate of returns from 2000 to 2020. The benchmark rate of return is the S&P 500. Often the S&P 500 is a proxy for the economy.

To analyze the rate of returns more meaningful, we will compare the rate of returns of companies from two different market sector industries against the S&P 500. We will look at Walmart and Target from the *Consumer Defensive* sector *Discount Stores* industry. We will look at Microsoft and Oracle from the *Technology* sector *Software—Infrastructure* industry.

The chapter is broken down as follows. In Sect. 4.2, we discuss the measure of central tendency. In Sect. 4.3, we discuss the measure of dispersion; in Sect. 4.4, we talk about the measure of relative position. Measure of shape is discussed in Sect. 4.5, and measure of linear relationship is talked about in Sect. 4.6. The box and whisker plot is examined in Sect. 4.7. An Excel rate of return box and whisker workbook is given in Sect. 4.8. Section 4.9 details how to create a box and whisker plot. In Sect. 4.10, we use Python to calculate the 5-year numerical measure of the rate of return of AAPL, MSFT, and the S&P 500. Finally, in Sect. 4.11, we summarize the chapter.

4.1.1 Summary Measures Excel Workbook

At the beginning of this chapter, we will describe the structure of the *Summary Measures* Excel workbook used in this chapter. We will then study the result of the summary measures for the S&P500, Walmart, and Target.

4.1.2 Summary Worksheet

The summary measure workbook contains four worksheets. The first worksheet is a summary worksheet that shows numerical summary measures for three tickers. The summary worksheet also has a histogram for each ticker and a histogram that has all three tickers. The workbook then has three worksheets that show return information for the three tickers. The

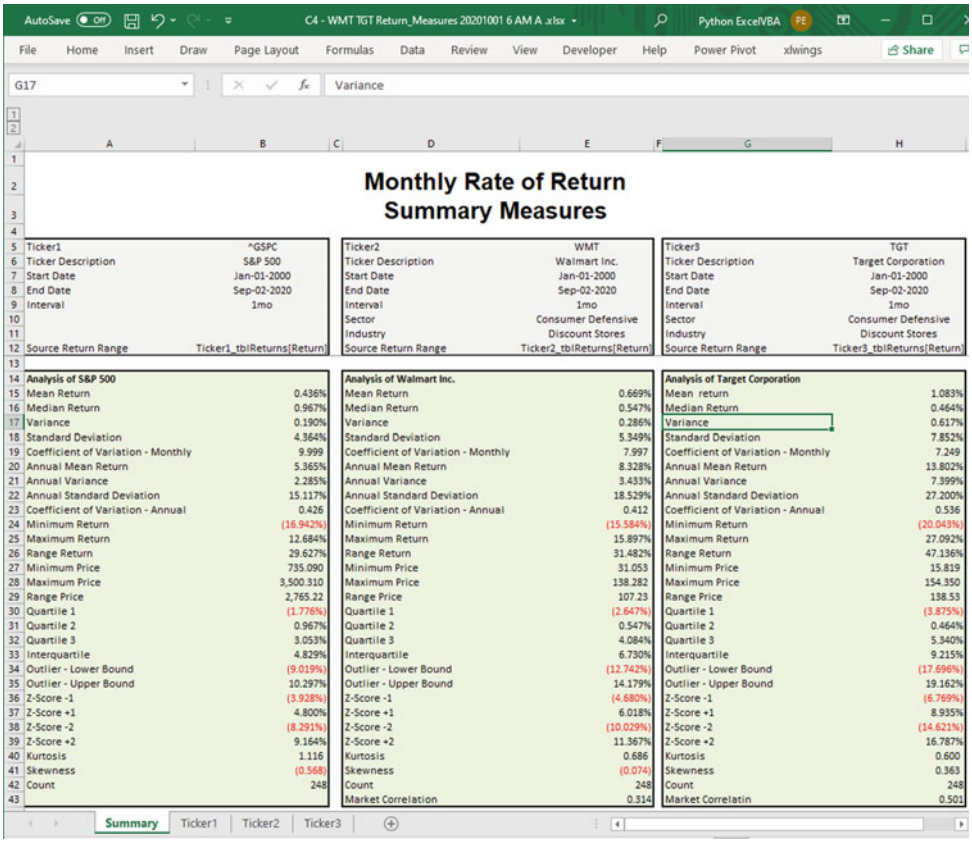
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_4

return information for ticker 1 is shown in the worksheet "Ticker1." The return information for ticker 2 is shown in the worksheet "Ticker2." The return information for ticker 3 is shown in the worksheet "Ticker3." The worksheet "Summary" uses the return information from the other 3 worksheets to produce numerical summary measures for the three tickers.

4.1.2.1 Numerical Summary Measures

The following shows the numerical summary measures of Walmart, Target, Microsoft, and Oracle. Walmart and Target companies are from the *Consumer Defensive* sector *Discount Stores* industry. Microsoft and Oracle are companies from the *Technology* sector *Software—Infrastructure* industry.



Monthly Rate of Return Summary Measures

Ticker1	^GSPC
Ticker Description	S&P 500
Start Date	Jan-01-2000
End Date	Sep-02-2020
Interval	1mo
Source Return Range	Ticker1_tblReturns[Return]

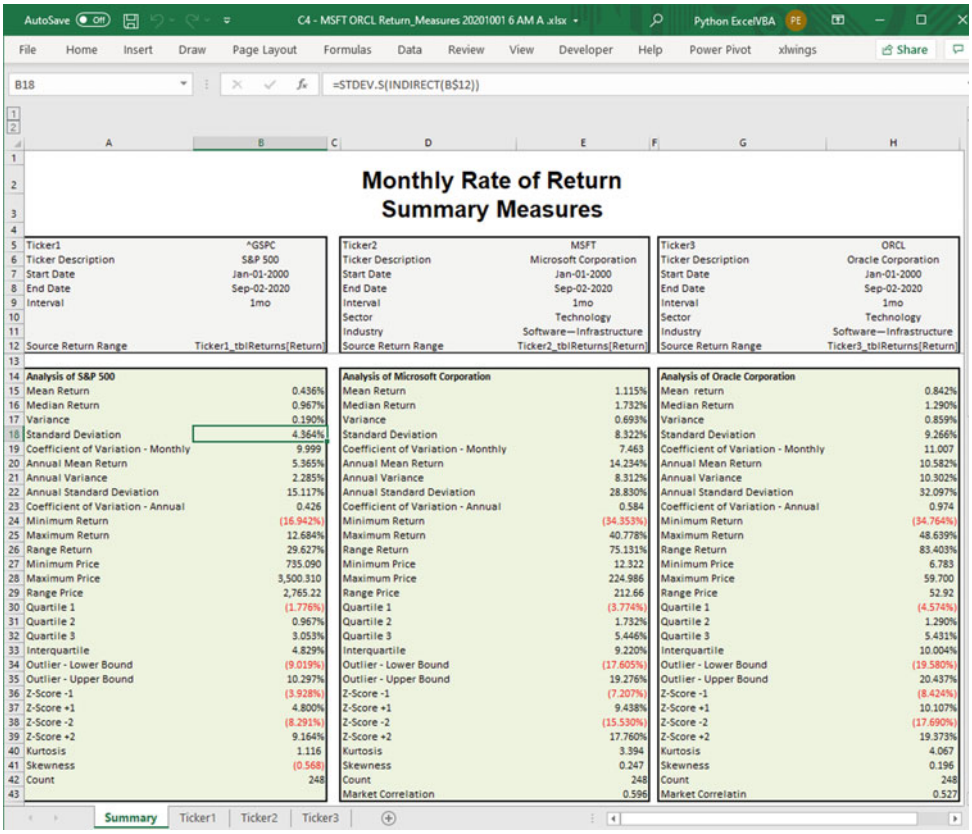
Analysis of S&P 500	
Mean Return	0.436%
Median Return	0.967%
Variance	0.190%
Standard Deviation	4.364%
Coefficient of Variation - Monthly	9.999
Annual Mean Return	5.365%
Annual Variance	2.285%
Annual Standard Deviation	15.117%
Coefficient of Variation - Annual	0.426
Minimum Return	(16.942%)
Maximum Return	12.684%
Range Return	29.627%
Minimum Price	735.090
Maximum Price	3,500.310
Range Price	2,765.22
Quartile 1	(1.776%)
Quartile 2	0.967%
Quartile 3	3.053%
Interquartile	4.829%
Outlier - Lower Bound	(9.019%)
Outlier - Upper Bound	10.297%
Z-Score -1	(3.928%)
Z-Score +1	4.800%
Z-Score -2	(8.291%)
Z-Score +2	9.164%
Kurtosis	1.116
Skewness	(0.568)
Count	248

Ticker2	WMT
Ticker Description	Walmart Inc.
Start Date	Jan-01-2000
End Date	Sep-02-2020
Interval	1mo
Sector	Consumer Defensive
Industry	Discount Stores
Source Return Range	Ticker2_tblReturns[Return]

Analysis of Walmart Inc.	
Mean Return	0.669%
Median Return	0.547%
Variance	0.286%
Standard Deviation	5.349%
Coefficient of Variation - Monthly	7.997
Annual Mean Return	8.328%
Annual Variance	3.433%
Annual Standard Deviation	18.529%
Coefficient of Variation - Annual	0.412
Minimum Return	(15.584%)
Maximum Return	15.897%
Range Return	31.482%
Minimum Price	31.053
Maximum Price	138.282
Range Price	107.23
Quartile 1	(2.647%)
Quartile 2	0.547%
Quartile 3	4.084%
Interquartile	6.730%
Outlier - Lower Bound	(12.742%)
Outlier - Upper Bound	14.179%
Z-Score -1	(4.680%)
Z-Score +1	6.018%
Z-Score -2	(10.029%)
Z-Score +2	11.367%
Kurtosis	0.686
Skewness	(0.074)
Count	248
Market Correlation	0.314

Ticker3	TGT
Ticker Description	Target Corporation
Start Date	Jan-01-2000
End Date	Sep-02-2020
Interval	1mo
Sector	Consumer Defensive
Industry	Discount Stores
Source Return Range	Ticker3_tblReturns[Return]

Analysis of Target Corporation	
Mean return	1.083%
Median Return	0.464%
Variance	0.617%
Standard Deviation	7.852%
Coefficient of Variation - Monthly	7.249
Annual Mean Return	13.802%
Annual Variance	7.399%
Annual Standard Deviation	27.200%
Coefficient of Variation - Annual	0.536
Minimum Return	(20.043%)
Maximum Return	27.092%
Range Return	47.136%
Minimum Price	15.819
Maximum Price	154.350
Range Price	138.53
Quartile 1	(3.875%)
Quartile 2	0.464%
Quartile 3	5.340%
Interquartile	9.215%
Outlier - Lower Bound	(17.696%)
Outlier - Upper Bound	19.162%
Z-Score -1	(6.769%)
Z-Score +1	8.935%
Z-Score -2	(14.621%)
Z-Score +2	16.787%
Kurtosis	0.600
Skewness	0.363
Count	248
Market Correlatin	0.501



C4 - MSFT ORCL Return_Measures 20201001 6 AM A .xlsx

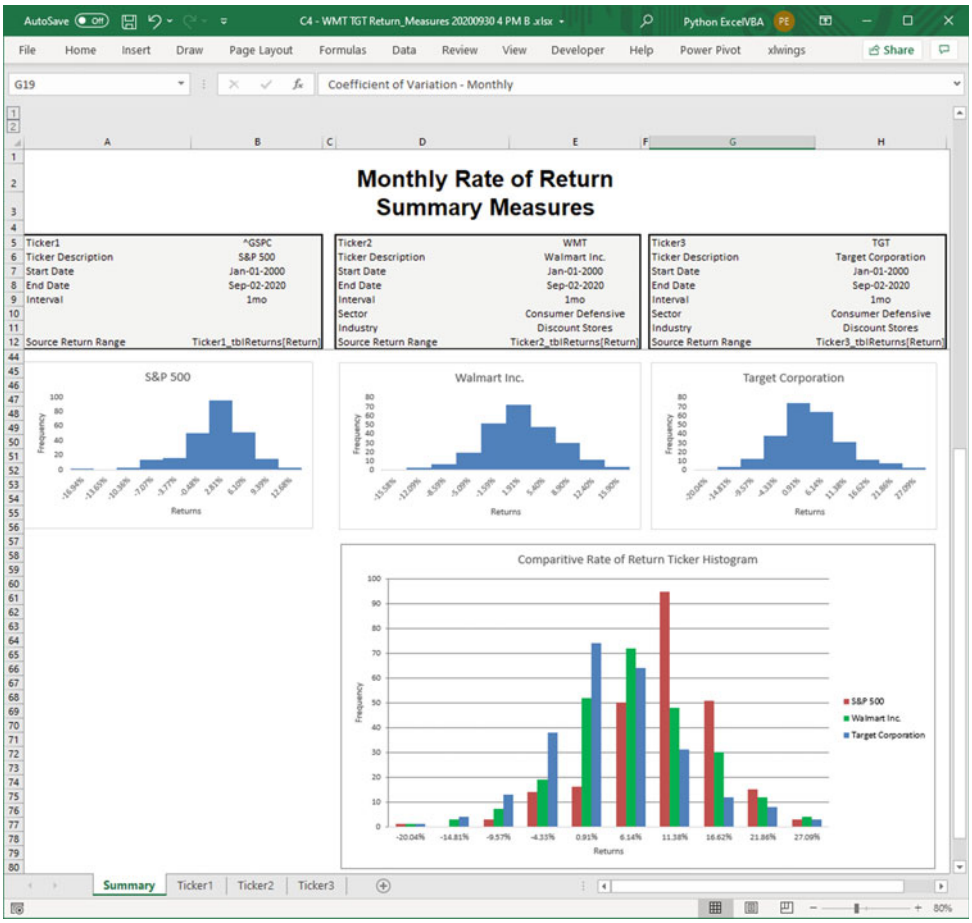
=STDEV.S(INDIRECT(B$12))

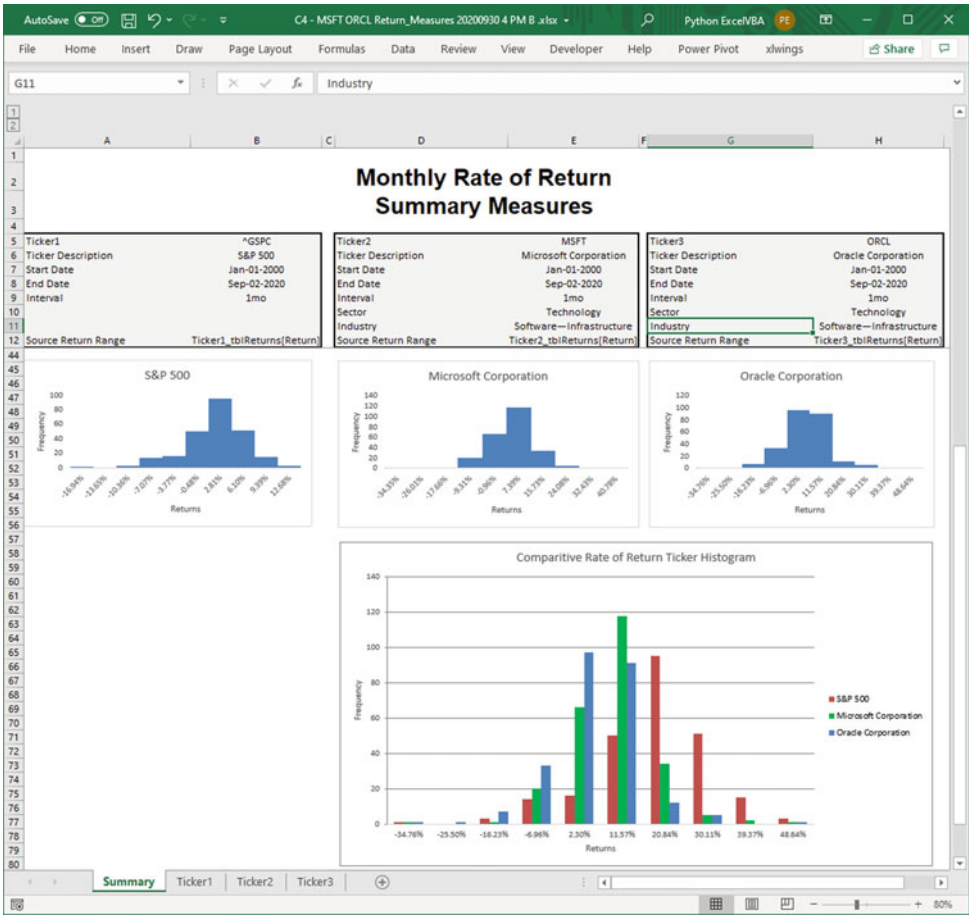
Monthly Rate of Return
Summary Measures

Ticker1	^GSPC		Ticker2	MSFT		Ticker3	ORCL	
Ticker Description	S&P 500		Ticker Description	Microsoft Corporation		Ticker Description	Oracle Corporation	
Start Date	Jan-01-2000		Start Date	Jan-01-2000		Start Date	Jan-01-2000	
End Date	Sep-02-2020		End Date	Sep-02-2020		End Date	Sep-02-2020	
Interval	1mo		Interval	1mo		Interval	1mo	
			Sector	Technology		Sector	Technology	
			Industry	Software—Infrastructure		Industry	Software—Infrastructure	
Source Return Range	Ticker1_tblReturns[Return]		Source Return Range	Ticker2_tblReturns[Return]		Source Return Range	Ticker3_tblReturns[Return]	
Analysis of S&P 500			**Analysis of Microsoft Corporation**			**Analysis of Oracle Corporation**		
Mean Return	0.436%		Mean Return	1.115%		Mean return	0.842%	
Median Return	0.967%		Median Return	1.732%		Median Return	1.290%	
Variance	0.190%		Variance	0.693%		Variance	0.859%	
Standard Deviation	4.364%		Standard Deviation	8.322%		Standard Deviation	9.266%	
Coefficient of Variation - Monthly	9.999		Coefficient of Variation - Monthly	7.463		Coefficient of Variation - Monthly	11.007	
Annual Mean Return	5.365%		Annual Mean Return	14.234%		Annual Mean Return	10.582%	
Annual Variance	2.285%		Annual Variance	8.312%		Annual Variance	10.302%	
Annual Standard Deviation	15.117%		Annual Standard Deviation	28.830%		Annual Standard Deviation	32.097%	
Coefficient of Variation - Annual	0.426		Coefficient of Variation - Annual	0.584		Coefficient of Variation - Annual	0.974	
Minimum Return	(16.942%)		Minimum Return	(34.353%)		Minimum Return	(34.764%)	
Maximum Return	12.684%		Maximum Return	40.778%		Maximum Return	48.639%	
Range Return	29.627%		Range Return	75.131%		Range Return	83.403%	
Minimum Price	735.090		Minimum Price	12.322		Minimum Price	6.783	
Maximum Price	3,500.310		Maximum Price	224.986		Maximum Price	59.700	
Range Price	2,765.22		Range Price	212.66		Range Price	52.92	
Quartile 1	(1.776%)		Quartile 1	(3.774%)		Quartile 1	(4.574%)	
Quartile 2	0.967%		Quartile 2	1.732%		Quartile 2	1.290%	
Quartile 3	3.053%		Quartile 3	5.446%		Quartile 3	5.431%	
Interquartile	4.829%		Interquartile	9.220%		Interquartile	10.004%	
Outlier - Lower Bound	(9.019%)		Outlier - Lower Bound	(17.605%)		Outlier - Lower Bound	(19.580%)	
Outlier - Upper Bound	10.297%		Outlier - Upper Bound	19.276%		Outlier - Upper Bound	20.437%	
Z-Score -1	(3.928%)		Z-Score -1	(7.207%)		Z-Score -1	(8.424%)	
Z-Score +1	4.800%		Z-Score +1	9.438%		Z-Score +1	10.107%	
Z-Score -2	(8.291%)		Z-Score -2	(15.530%)		Z-Score -2	(17.690%)	
Z-Score +2	9.164%		Z-Score +2	17.760%		Z-Score +2	19.373%	
Kurtosis	1.116		Kurtosis	3.394		Kurtosis	4.067	
Skewness	(0.568)		Skewness	0.247		Skewness	0.196	
Count	248		Count	248		Count	248	
			Market Correlation	0.596		Market Correlatin	0.527	

Summary | Ticker1 | Ticker2 | Ticker3

4.1.2.2 Histograms of Rate of Returns

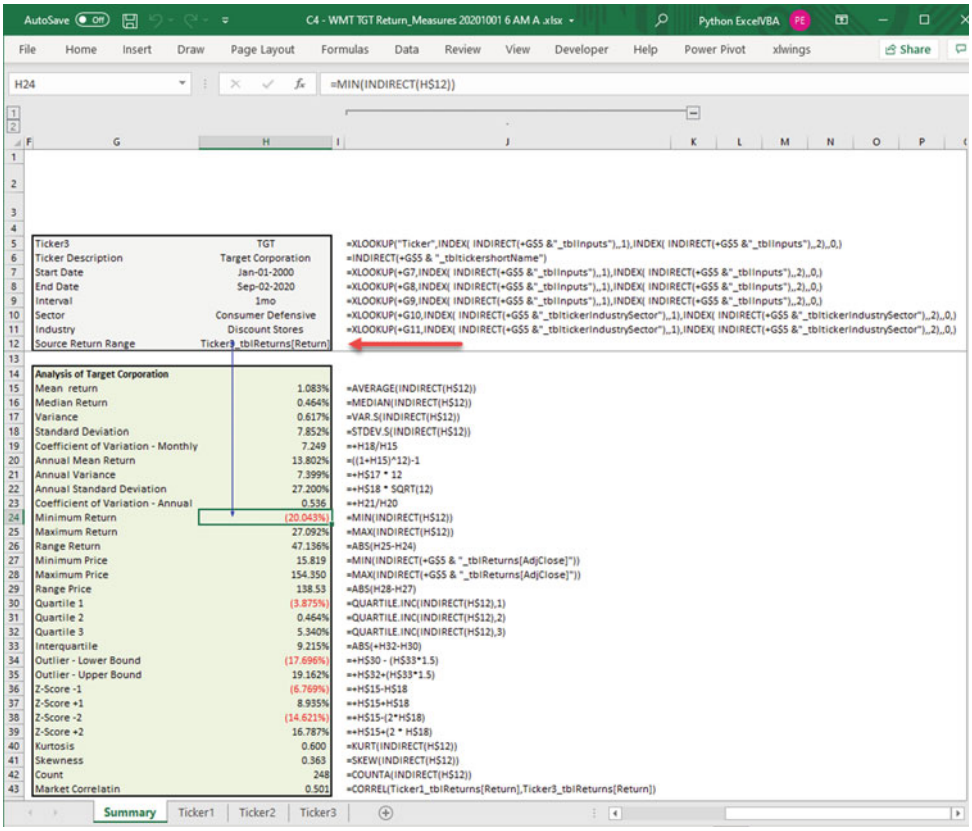




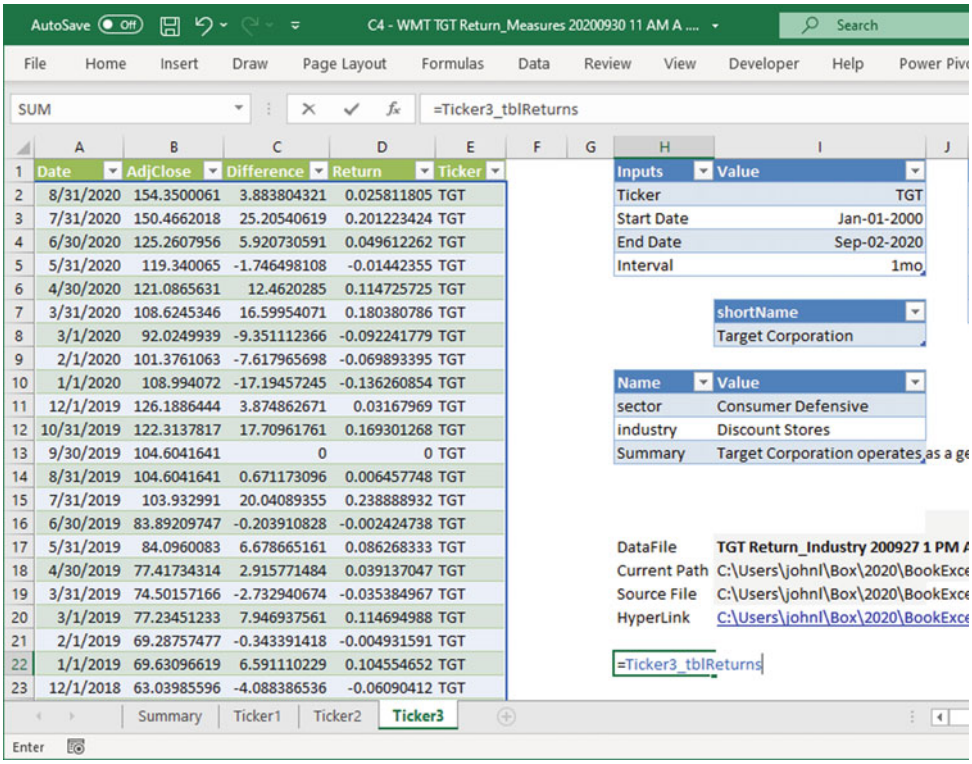
The rate of return histograms for Walmart, Target, Microsoft, Oracle, and the S&P 500 is shown above.

4.1.2.3 Excel Formulas

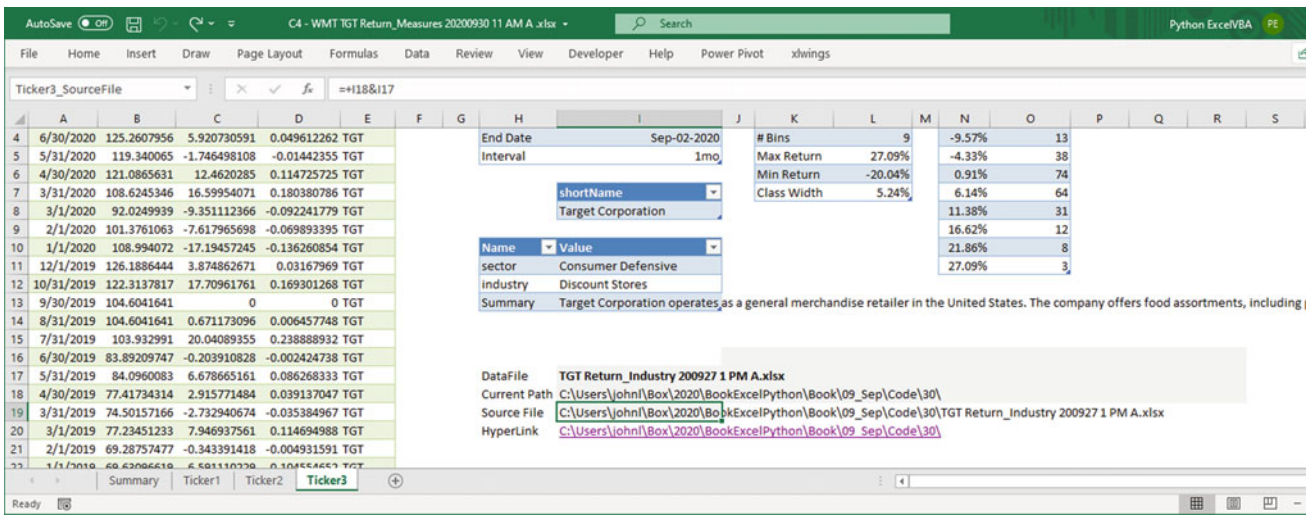
The following shows the Excel functions and formulas used to calculate the numerical summary measures for all tickers:



For ticker 3, the formulas reference a range name called *Ticker3_tblReturns[Return]*. This range name is referencing an Excel Table column in the worksheet *Ticker3*. *Ticker3_tblReturns* is the name of the Excel Data Table. *[Return]* is the name of a column in the Excel Data Table.



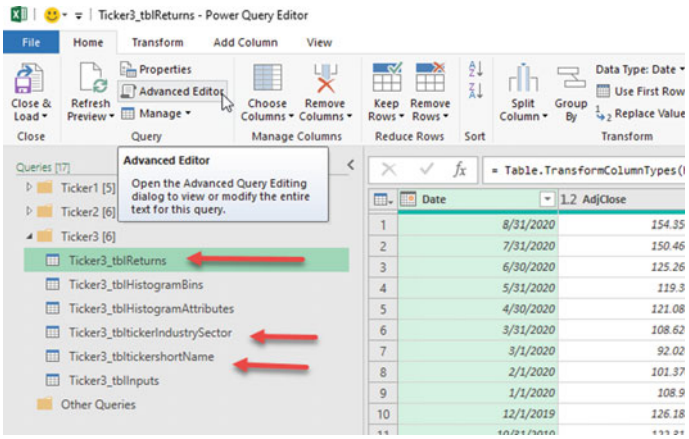
4.1.2.4 *Ticker2, Ticker3* Worksheets



The data source for the "Ticker3" worksheet is another Excel workbook. The data source workbook is shown in cell I19. The data source workbook is the workbook that we created in Chap. 33.

M Code

We will adjust the *M Code* for the *Power Queries* in the "Ticker3" to look for the range name *Ticker3_SourceFile* for the Excel data source name. This *M Code* technique was demonstrated in Chap. 3.

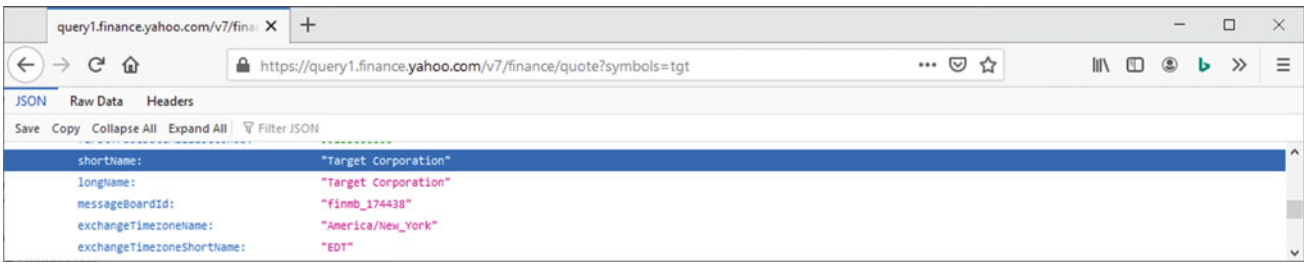


The *M Code* for the queries *Ticker3_tblReturns, Ticker3_tbltickerIndustrySector,* and *Ticker3_tbltickershortName* will be modified.

Yahoo! Finance API

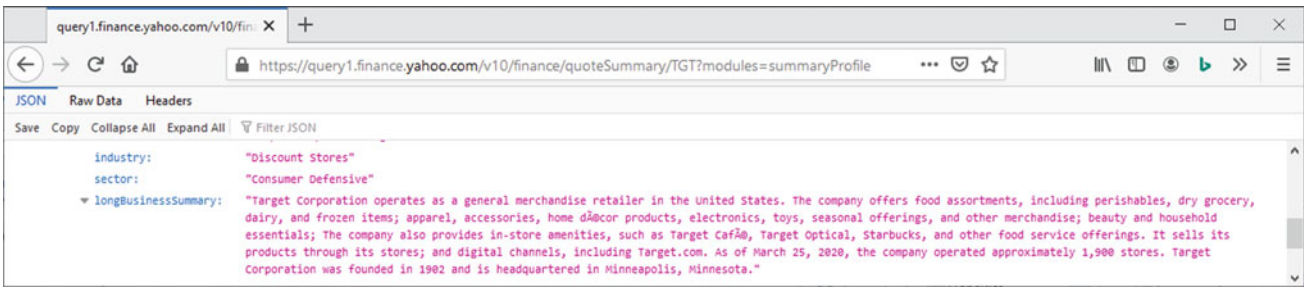
In this chapter, we enhanced Chap. 3's workbook by using two additional Yahoo! Finance APIs. This chapter used the following Yahoo! Finance API to get the shortName of a ticker in cell I8. The following is the API to retrieve the shortName for TGT:

https://query1.finance.yahoo.com/v7/finance/quote?symbols=tgt

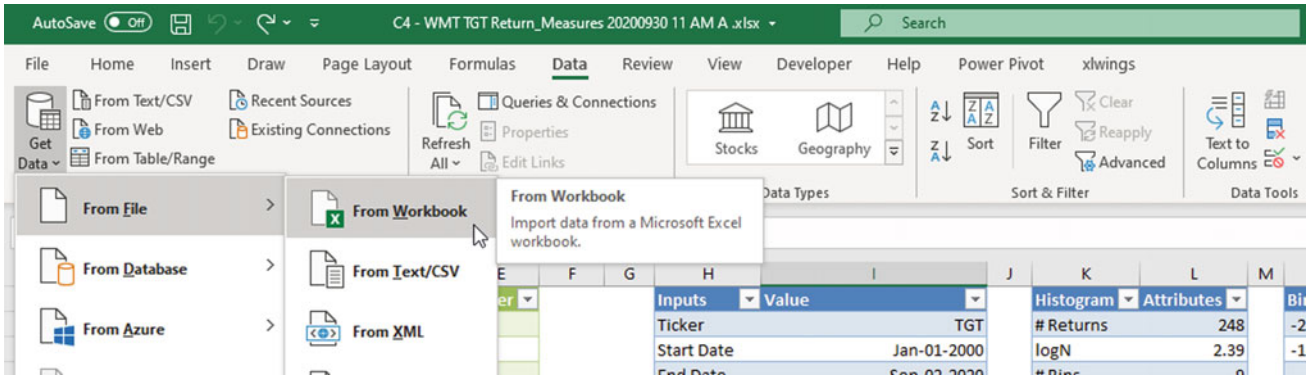


This chapter used the following Yahoo! Finance API to get the sector, industry, and summary of a ticker in cell H11. The following is the API to retrieve the sector, industry, and summary for TGT:

https://query1.finance.yahoo.com/v10/finance/quoteSummary/TGT?modules=summaryProfile

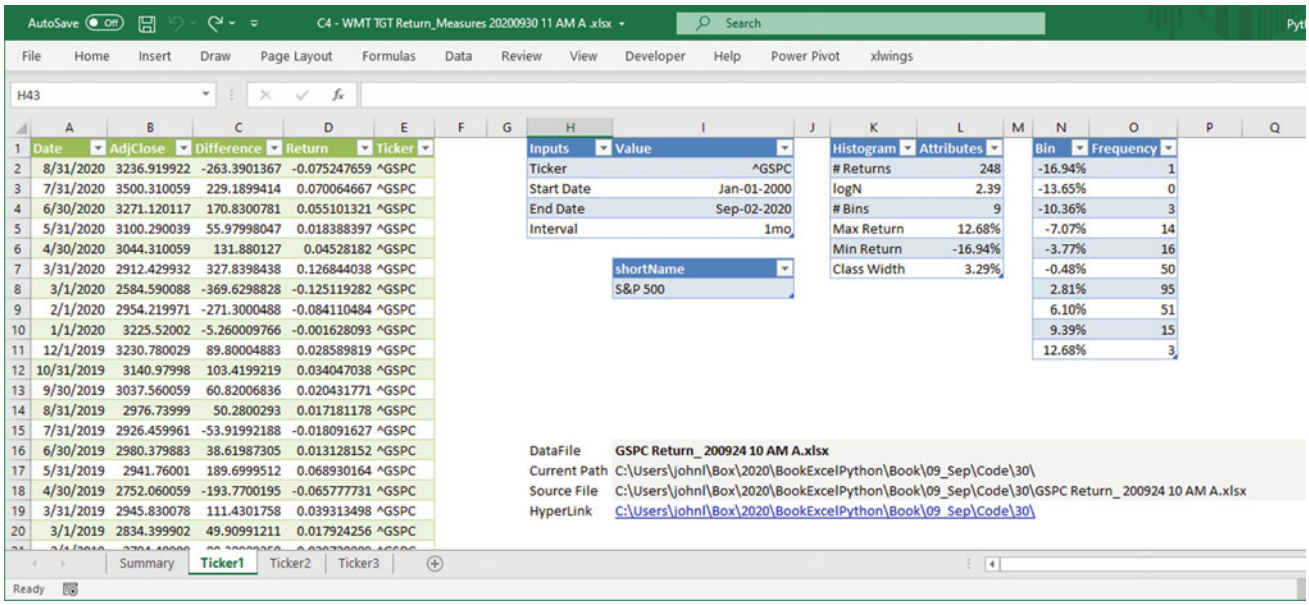


We will use *Power Query* to retrieve data from a closed workbook.



Before *Power Query,* it was very cumbersome for an open Excel workbook to retrieve data from a closed workbook.

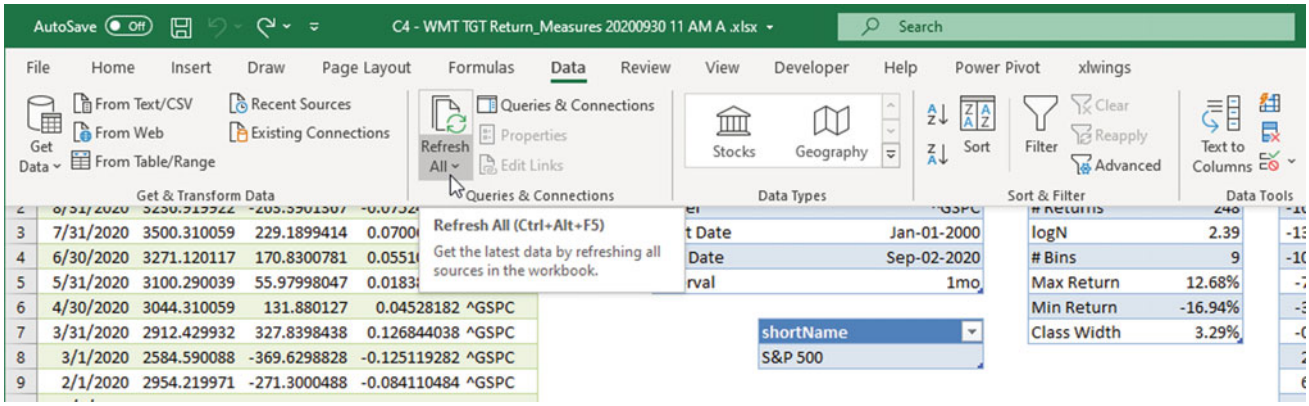
4.1.2.5 *Ticker1* Worksheets



The structure of the "Ticker1" worksheet is the same as "Ticker2" and "Ticker3." The only missing is the sector, industry, and summary table. There are many types of tickers. There are many tickers that do not have sector, industry, and summary information.

4.1.2.6 Data Refresh All

We will click on *Refresh All* from the *Data* tab to retrieve the data from the three close-source Excel workbooks.



4.2 Measure of Central Tendency

The purpose of a *measure of central tendency* is to determine where a distribution of a data set cluster. Below we will talk about two measures of central tendency, the mean and the median. The mean is the more important of the two even though there are pros and cons to both the mean and median.

4.2.1 Arithmetic Mean (Average)

We will look at two measures of the central tendency of a data set. The first one is the arithmetic mean or average. Mathematically, it is defined as

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

The *AVERAGE* Excel function is used to calculate the arithmetic mean. It is often said that averages are influenced by extreme values.

4.2.2 Annualized Monthly Returns

Frequently, we have monthly returns data that we need to annualize. The formula to convert monthly returns to annual returns is

$$((1 + \text{average monthly return})\text{^}12) + 1$$

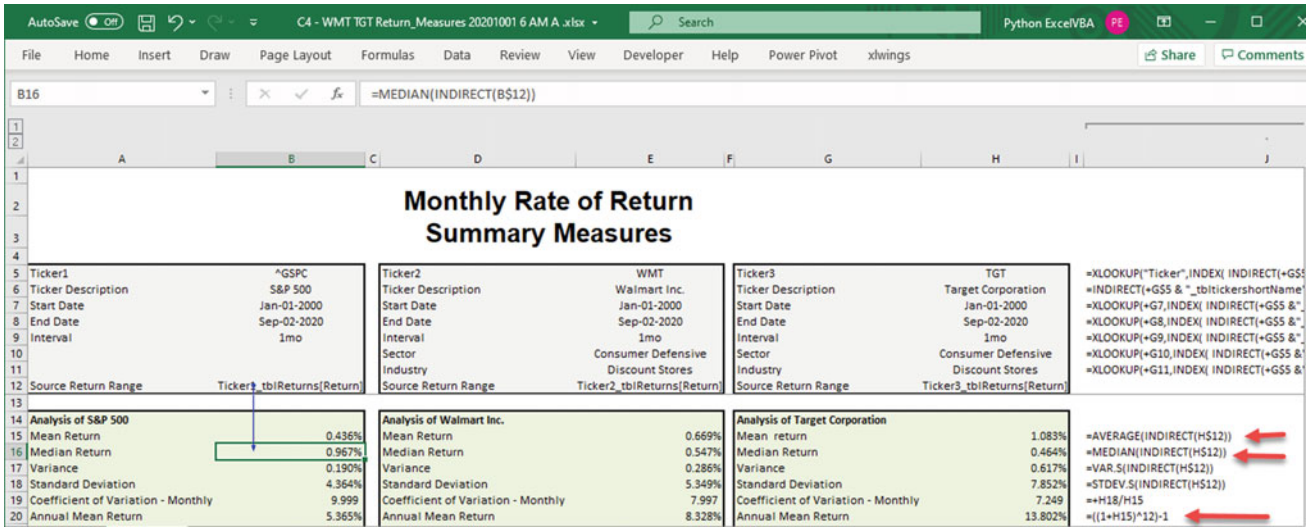
4.2.3 Median

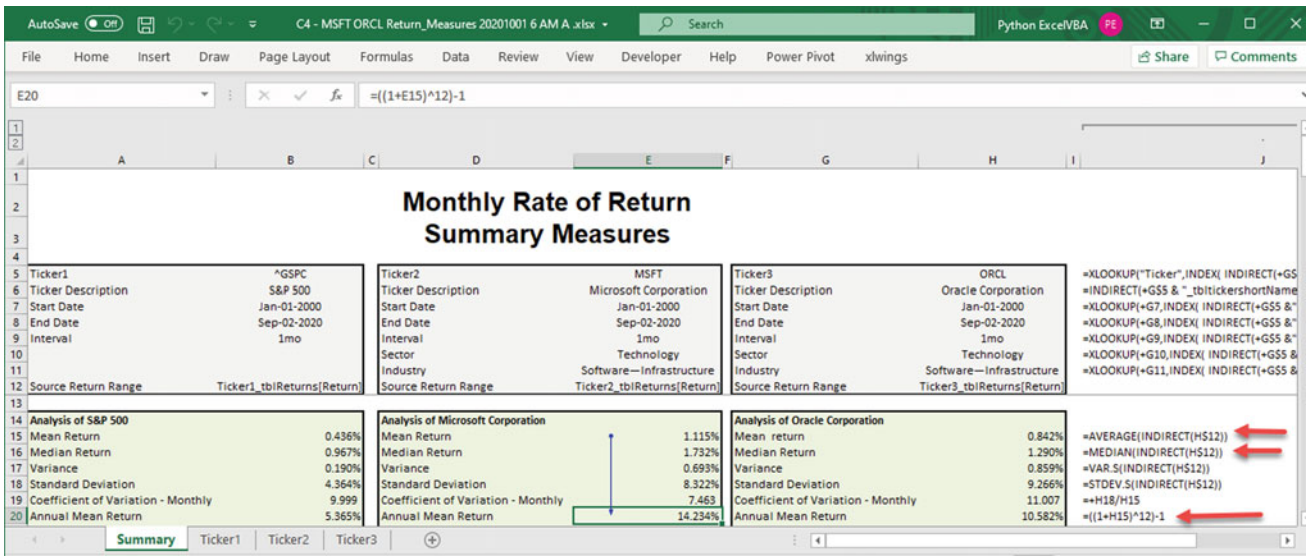
Another measure of the central tendency of a data set is the median. The median is the middle observation of a data set.

The *MEDIAN* Excel formula is used to calculate the arithmetic mean. It is often said that medians are not influenced by extreme values.

4.2.4 Excel Functions

The following shows the Excel functions for the mean, median, and annualize mean for companies for the *Consumer Defensive* sector, *Discount Stores* industry, the *Technology* sector *Software—Infrastructure* industry, and the S&P 500.





We can see that companies from the same sector, and industry cluster around the same median, i.e., extreme values do not influence the median value. While companies from the same sector, industry companies do not cluster around the same mean, i.e., extreme value influences the mean.

4.3 Measure of Dispersion

The purpose of the measure of central tendency is to tell us how tightly or loosely the data cluster around the measure of central tendency or mean.

4.3.1 Variance

For each data in a data set, we would like to know how far it's from the mean. There can be a positive difference or a negative difference. One way to figure out how tightly or loosely a data set is clustered around the mean is to sum the difference. The problem with this approach is the negative and positive numbers cancel each other out. Statisticians have

come up with the approach to sum the square of all differences with each difference. Statisticians then divide the sum of the square difference by the data set size to get the average difference per data in a data set. The formula for the population variance is

$$\sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}$$

The formula for the sample variance is

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n - 1}$$

The *VAR.S* EXCEL function is used to calculate the sample variance.

4.3.2 Annualized Monthly Variance

The formula to convert monthly variance to annual variance is

$$\sigma_T = \sigma_t * \sqrt{T}$$

where

σ_t = monthly standard deviation
T = number of months.

4.3.3 Standard Deviation

One problem with the variance is that the unit for a variance is the square of the units of the data it is for. This is why the standard deviation is often quoted. The standard deviation is the square root of the variance.

The formula for the population *standard deviation* is

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}}$$

The formula for the sample standard deviation is

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n - 1}}$$

The *STDEV.S* EXCEL function is used to calculate the sample variance.

4.3.4 Annualized Monthly Standard Deviation

The formula to convert monthly standard deviations to annual standard deviation is

$$\sigma_T = \sigma_t * \sqrt{T}$$

where

σ_t = monthly standard deviation
T = number of months.

4.3.5 Coefficient of Variation

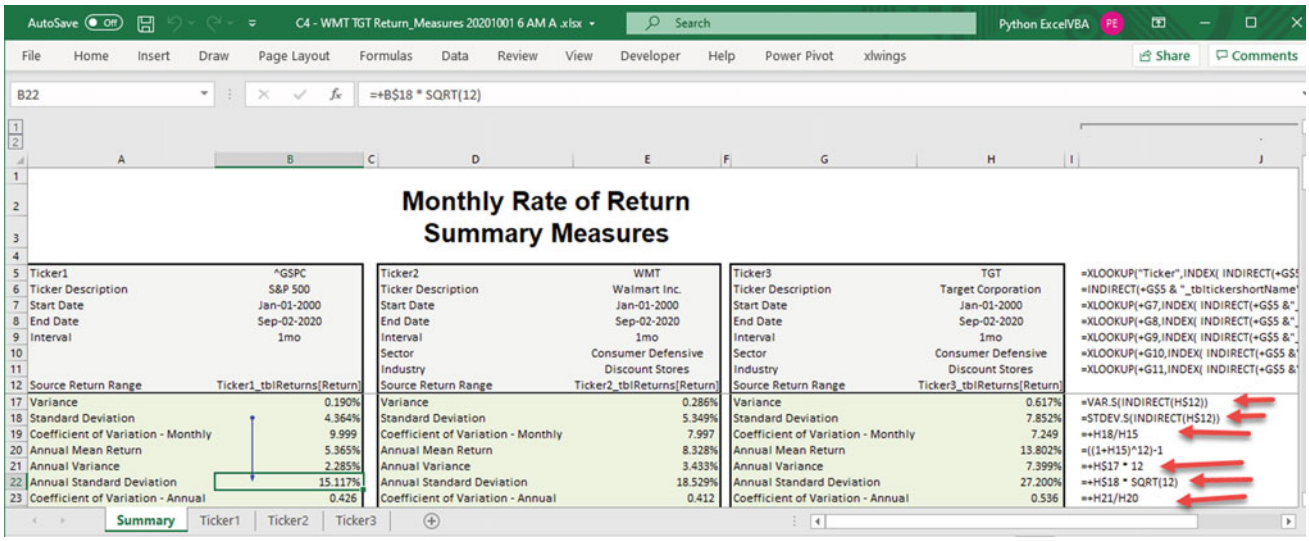
One problem with the interpretation of the standard deviation and variance is it's hard to tell the significance of the number reported. In finance, a larger standard deviation or larger variance equates to more risk. It turns out that a standard deviation number can be deceiving.

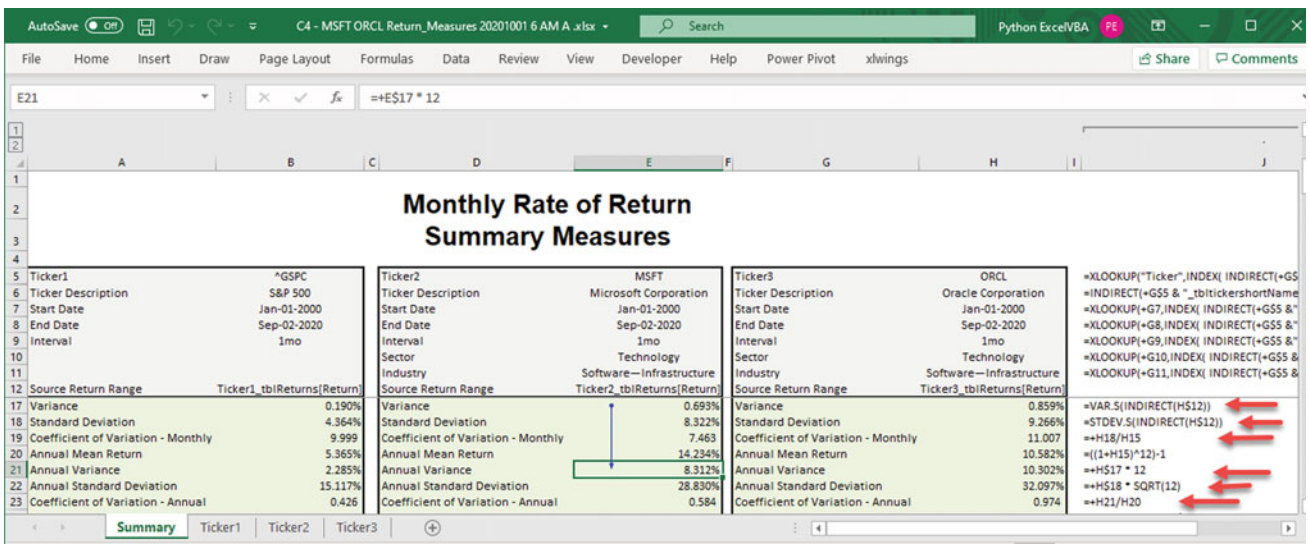
In finance, we want a high average and a low standard deviation. What we want is the lowest amount of risk per mean, which is the ratio of standard deviation divided by the mean. This ratio is called the *coefficient of variation*. The formula for the *coefficient of variation* is

$$cv_x = \frac{s}{\bar{x}}$$

4.3.6 Excel Functions

The following shows the Excel functions and formulas for the mean, variance, annualized mean, and annualized variance for companies for the *Consumer Defensive* sector, *Discount Stores* industry, the *Technology* sector *Software—Infrastructure* industry, and the S&P 500.





4.4 Measure of Relative Position

The *measure of relative position* describes the relative position of a particular measurement in a data set. When analyzing relative positions, we first assume that the data set is sorted in ascending order.

4.4.1 Quartiles

When we are talking about quartiles, we are breaking up the data set into four equal parts. The data set is ordered from smallest to largest. The anchor point viewpoint for quartiles is the smallest number in the data set.

The first quartile number means it is larger than the first 25% of the ascending ordered data set. The second quartile number means that it is larger than the first 50% of the ascending ordered data set. The third quartile number means that is larger than the first 75% of the ascending ordered data set. The second quartile number is the same as the median of the data set. The interquartile is the difference between the third quartile number and the first number quartile.

The *QUARTILE.INC* Excel function is used to calculate the quartiles. The function uses a one to indicate the first quartile, a two to indicate the second quartile, and a three to indicate the third quartile.

4.4.2 Interquartile

Interquartile is defined as the third quartile minus the second quartile. Interquartile are used in finding outliers and drawing box and whisker plots. The interquartile contains half of the data set.

4.4.3 Outliers

An outlier is a value that is an extreme value from the mean. An outlier is defined as a value that is greater or less than 1.5 times the value of the interquartile.

4.4.4 Z-Score

The anchor viewpoint for *Z-Score* is the mean of the data set. The *Z-Score* indicates how many standard deviations away a data point is from the mean of the data set. A positive *Z-Score* indicates that the number is greater than the mean of the data set. A negative *Z-Score* indicates that the number is smaller than the mean of the data set.

The formula for a *Z-Score* is

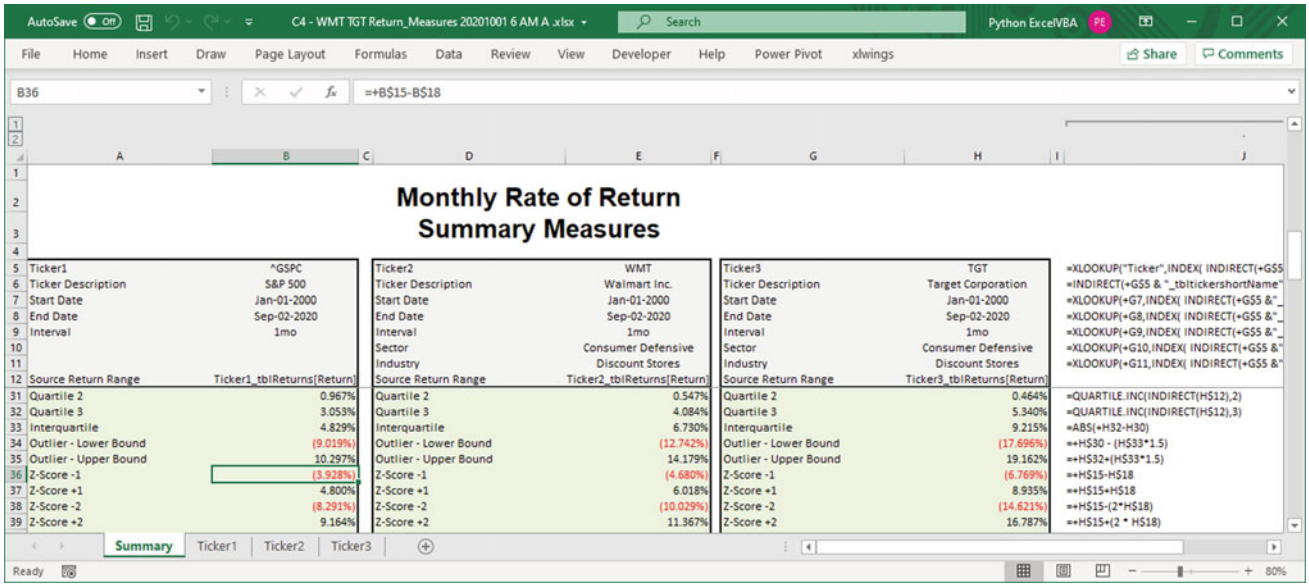
$$z = \frac{x - \overline{x}}{s}$$

The Z-score transforms a data set to have a mean of 0 and a standard deviation of 1.

As a rule of thumb, for mound-shaped data sets, approximately 68% of the observations have a Z-score between −1 and 1, and approximately 95% of the observations have a Z-score between −2 and 2.

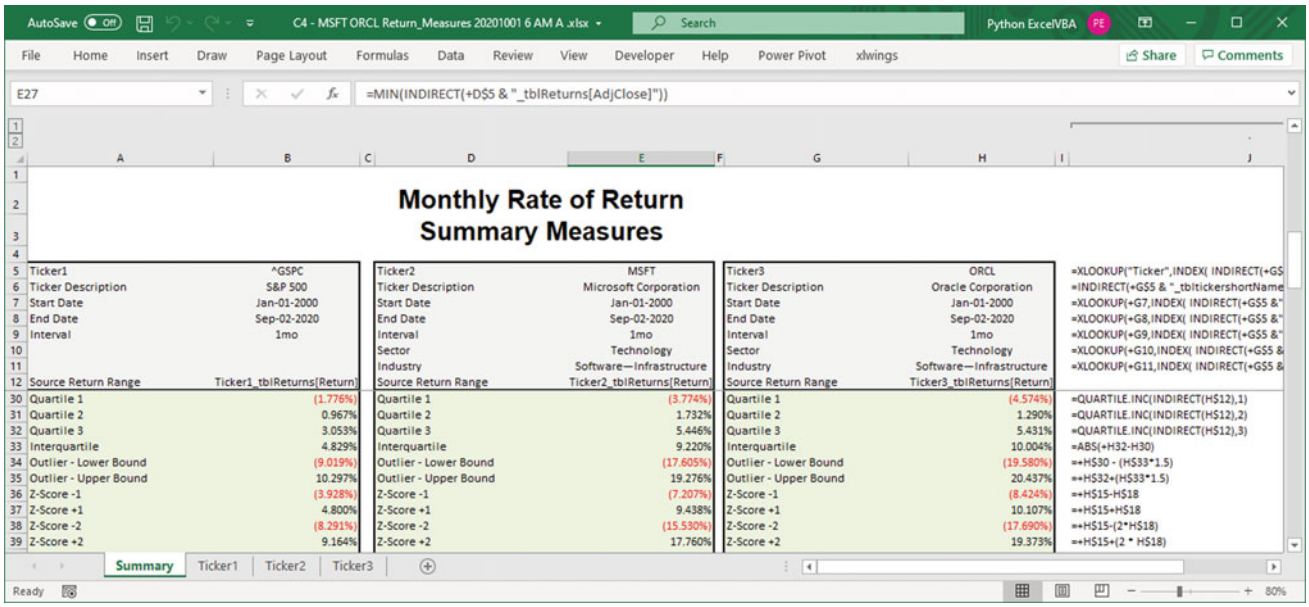
4.4.5 Excel Functions

The following shows the Excel functions and formulas for the mean, variance, annualized mean, and annualized variance for companies for the *Consumer Defensive* sector, *Discount Stores* industry, the *Technology* sector *Software—Infrastructure* industry, and the S&P 500.



Monthly Rate of Return
Summary Measures

Ticker1	^GSPC	Ticker2	WMT	Ticker3	TGT	=XLOOKUP("Ticker",INDEX(INDIRECT(+G$5
Ticker Description	S&P 500	Ticker Description	Walmart Inc.	Ticker Description	Target Corporation	=INDIRECT(+G$5 & "_tblTickershortName"
Start Date	Jan-01-2000	Start Date	Jan-01-2000	Start Date	Jan-01-2000	=XLOOKUP(+G7,INDEX(INDIRECT(+G$5 &"_
End Date	Sep-02-2020	End Date	Sep-02-2020	End Date	Sep-02-2020	=XLOOKUP(+G8,INDEX(INDIRECT(+G$5 &"_
Interval	1mo	Interval	1mo	Interval	1mo	=XLOOKUP(+G9,INDEX(INDIRECT(+G$5 &"_
		Sector	Consumer Defensive	Sector	Consumer Defensive	=XLOOKUP(+G10,INDEX(INDIRECT(+G$5 &"
		Industry	Discount Stores	Industry	Discount Stores	=XLOOKUP(+G11,INDEX(INDIRECT(+G$5 &"
Source Return Range	Ticker1_tblReturns[Return]	Source Return Range	Ticker2_tblReturns[Return]	Source Return Range	Ticker3_tblReturns[Return]	
Quartile 2	0.967%	Quartile 2	0.547%	Quartile 2	0.464%	=QUARTILE.INC(INDIRECT(H$12),2)
Quartile 3	3.053%	Quartile 3	4.084%	Quartile 3	5.340%	=QUARTILE.INC(INDIRECT(H$12),3)
Interquartile	4.829%	Interquartile	6.730%	Interquartile	9.215%	=ABS(+H32-H30)
Outlier - Lower Bound	(9.019%)	Outlier - Lower Bound	(12.742%)	Outlier - Lower Bound	(17.696%)	=+H$30 - (H$33*1.5)
Outlier - Upper Bound	10.297%	Outlier - Upper Bound	14.179%	Outlier - Upper Bound	19.162%	=+H$32+(H$33*1.5)
Z-Score -1	(3.928%)	Z-Score -1	(4.680%)	Z-Score -1	(6.769%)	=+H$15-H$18
Z-Score +1	4.800%	Z-Score +1	6.018%	Z-Score +1	8.935%	=+H$15+H$18
Z-Score -2	(8.291%)	Z-Score -2	(10.029%)	Z-Score -2	(14.621%)	=+H$15-(2*H$18)
Z-Score +2	9.164%	Z-Score +2	11.367%	Z-Score +2	16.787%	=+H$15+(2 * H$18)



Monthly Rate of Return
Summary Measures

Ticker1	^GSPC	Ticker2	MSFT	Ticker3	ORCL	=XLOOKUP("Ticker",INDEX(INDIRECT(+G$
Ticker Description	S&P 500	Ticker Description	Microsoft Corporation	Ticker Description	Oracle Corporation	=INDIRECT(+G$5 & "_tblTickershortName
Start Date	Jan-01-2000	Start Date	Jan-01-2000	Start Date	Jan-01-2000	=XLOOKUP(+G7,INDEX(INDIRECT(+G$5 &"
End Date	Sep-02-2020	End Date	Sep-02-2020	End Date	Sep-02-2020	=XLOOKUP(+G8,INDEX(INDIRECT(+G$5 &"
Interval	1mo	Interval	1mo	Interval	1mo	=XLOOKUP(+G9,INDEX(INDIRECT(+G$5 &"
		Sector	Technology	Sector	Technology	=XLOOKUP(+G10,INDEX(INDIRECT(+G$5 &
		Industry	Software—Infrastructure	Industry	Software—Infrastructure	=XLOOKUP(+G11,INDEX(INDIRECT(+G$5 &
Source Return Range	Ticker1_tblReturns[Return]	Source Return Range	Ticker2_tblReturns[Return]	Source Return Range	Ticker3_tblReturns[Return]	
Quartile 1	(1.776%)	Quartile 1	(3.774%)	Quartile 1	(4.574%)	=QUARTILE.INC(INDIRECT(H$12),1)
Quartile 2	0.967%	Quartile 2	1.732%	Quartile 2	1.290%	=QUARTILE.INC(INDIRECT(H$12),2)
Quartile 3	3.053%	Quartile 3	5.446%	Quartile 3	5.431%	=QUARTILE.INC(INDIRECT(H$12),3)
Interquartile	4.829%	Interquartile	9.220%	Interquartile	10.004%	=ABS(+H32-H30)
Outlier - Lower Bound	(9.019%)	Outlier - Lower Bound	(17.605%)	Outlier - Lower Bound	(19.580%)	=+H$30 - (H$33*1.5)
Outlier - Upper Bound	10.297%	Outlier - Upper Bound	19.276%	Outlier - Upper Bound	20.437%	=+H$32+(H$33*1.5)
Z-Score -1	(3.928%)	Z-Score -1	(7.207%)	Z-Score -1	(8.424%)	=+H$15-H$18
Z-Score +1	4.800%	Z-Score +1	9.438%	Z-Score +1	10.107%	=+H$15+H$18
Z-Score -2	(8.291%)	Z-Score -2	(15.530%)	Z-Score -2	(17.690%)	=+H$15-(2*H$18)
Z-Score +2	9.164%	Z-Score +2	17.760%	Z-Score +2	19.373%	=+H$15+(2 * H$18)

4.5 Measure of Shape

4.5.1 Skewness

The *skewness* indicates whether the distribution of a data set is skewed to the left or right in relation to the mean or is symmetric about the mean.

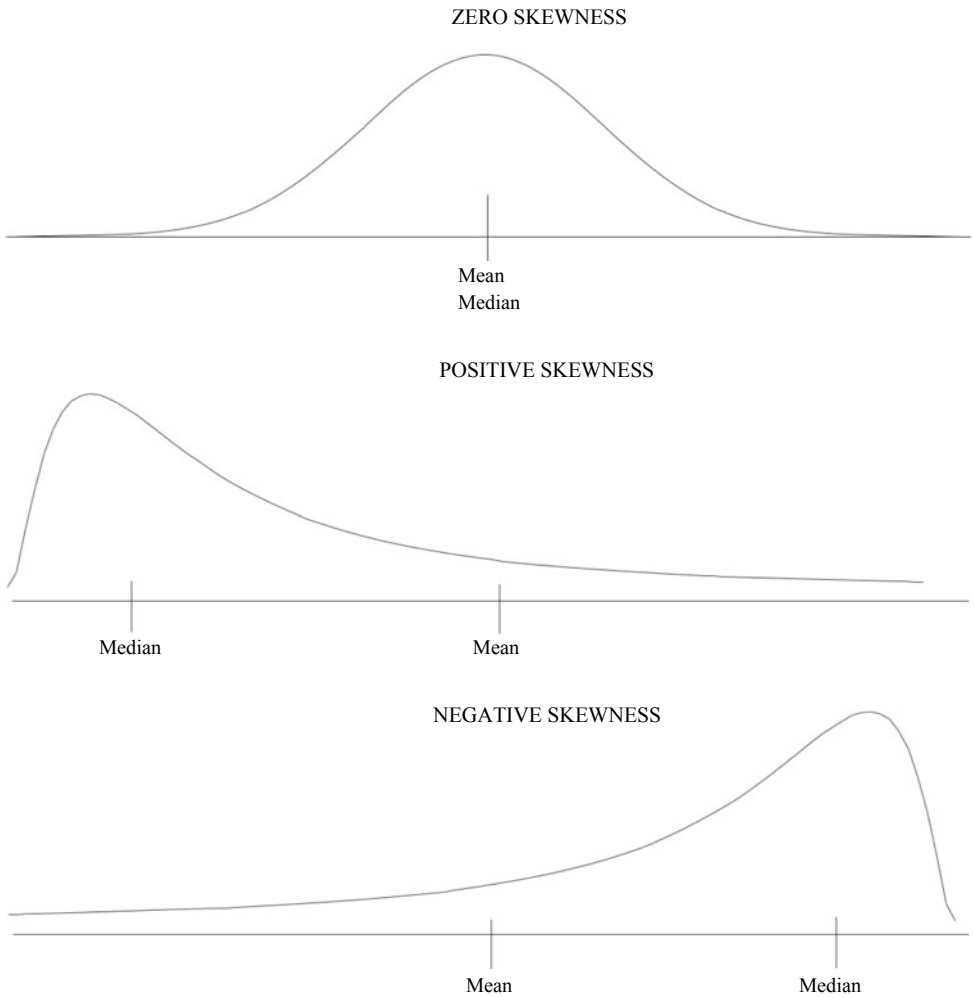
Zero skewness means that the distribution is symmetric, with the mean equal to the median.

Positive skewness means that the distribution is skewed to the right and that the median lies below or to the left of the mean.

Negative skewness means that the distribution is skewed to the left and the median is above or to the right of the mean.

90 4 Numerical Summary Measures on Rate of Returns of Amazon, Walmart, and the S&P 500

These concepts are graphically illustrated below.



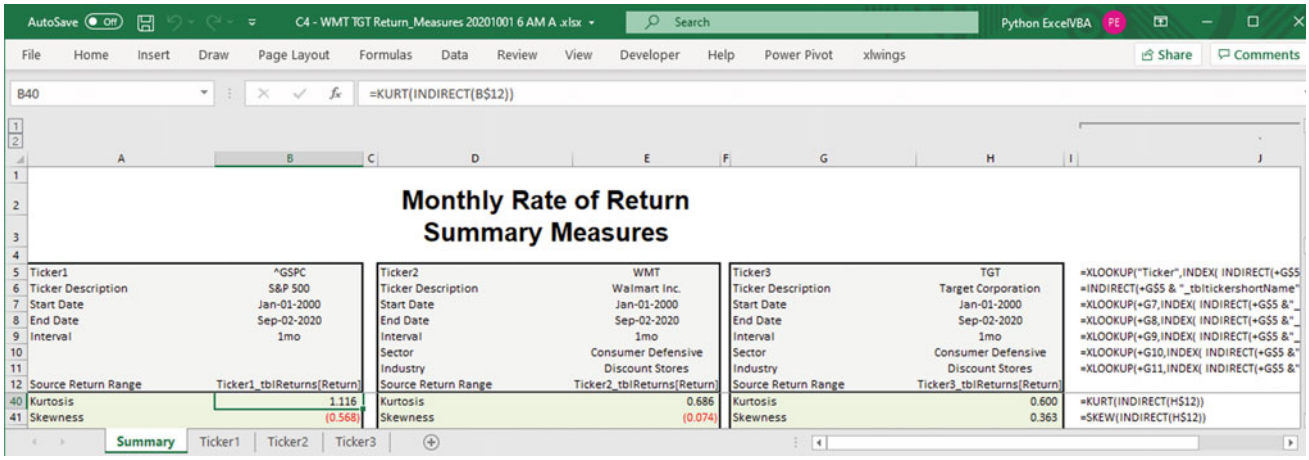
Notice that for the zero skewness, the hump of the distribution is in the middle. The hump of the positive skewness is toward the left. The hump of the negative skewness is toward the right.

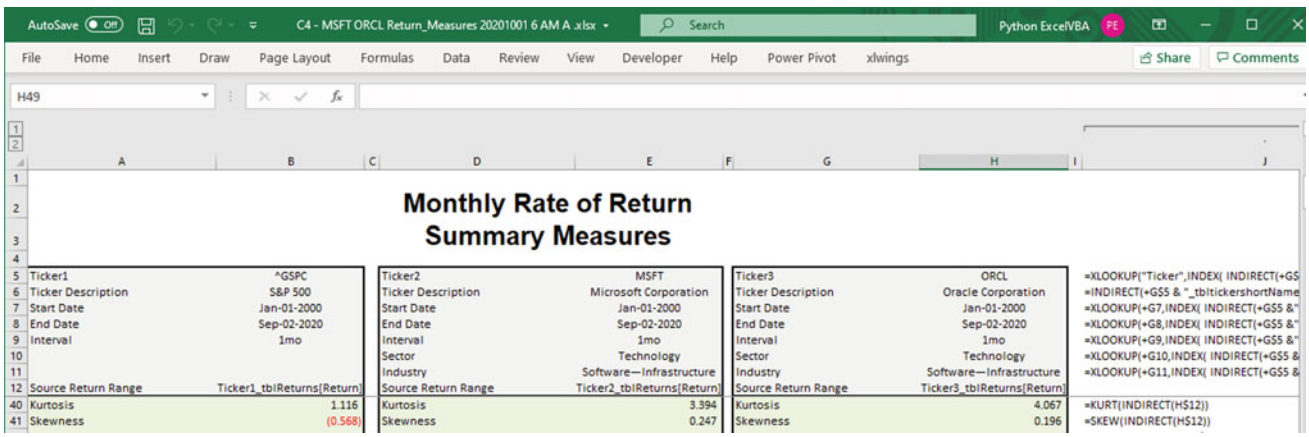
4.5.2 Kurtosis

Kurtosis measures the peakedness or flatness of a distribution. A positive kurtosis indicates a relatively peaked distribution. A negative kurtosis indicates a relatively flat distribution.

4.5.3 Excel Functions

The following shows the Excel functions for the skewness and kurtosis for companies for the *Consumer Defensive* sector, the *Discount Stores* industry, and the *Technology* sector *Software—Infrastructure* industry, and the S&P 500.





4.6 Measure of Linear Relationship

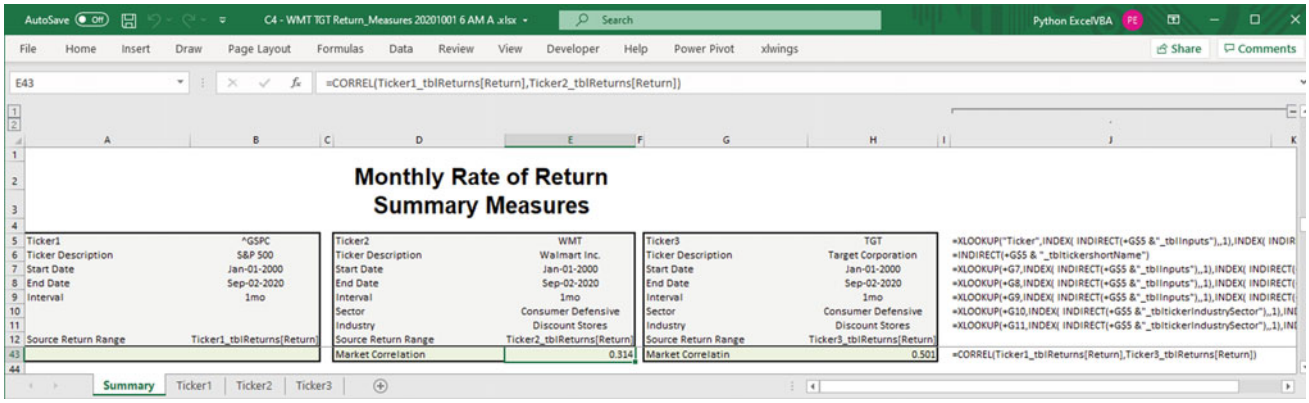
4.6.1 Coefficient of Correlation

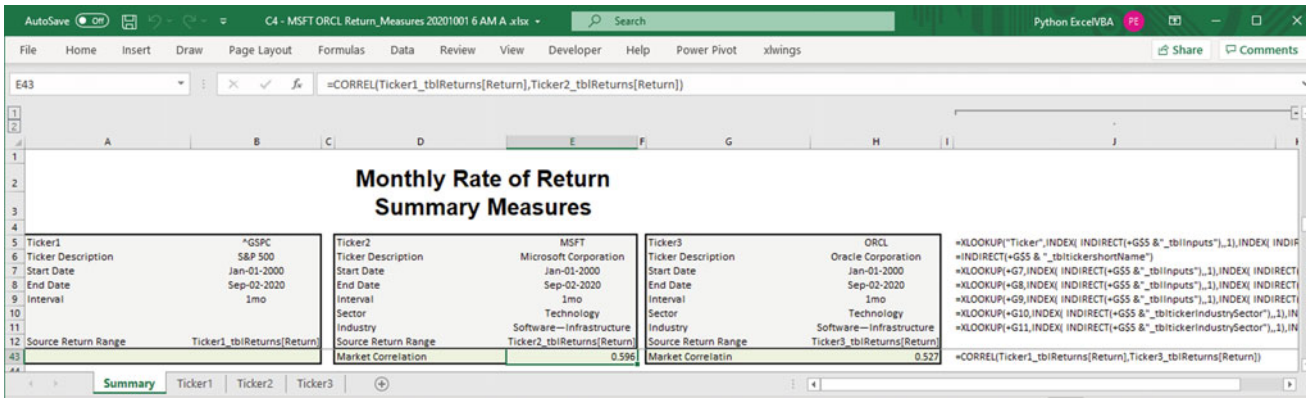
The *coefficient of correlation* is a measurement of the linear relationship between two data sets. The coefficient of correlation is a number between 1 and 1. A 1 means a perfect positive correlation, and a 1 means a perfect negative correlation.

The *CORREL* Excel function is used to calculate the correlation between two data sets.

4.6.2 Excel Functions

The following shows the Excel functions for the rate of return correlations between the S&P 500 companies in the *Consumer Defensive* sector, *Discount Stores* industry, and in the *Technology* sector *Software—Infrastructure* industry.





4.7 Box and Whisker Plot

A *box and whisker plot* is a graphical representation of a data set that illustrates the quartiles of a data set with a visual emphasis on the middle two quartiles. The box and whisker plots will emphasize the 25th percentile, the median and mean, the 75th percentile, the largest or smallest numbers that are not an outlier, and outliers of a data set. *Traditionally, box and whisker plots are shown horizontally. This section will talk about box and whisker plots shown vertically because Excel has a built-in box–whisker plot that is shown vertically.* The Excel box and whisker plot chart was added in Excel 2016.

4.7.1 Outliers

An outlier is defined as 1.5 times the 3rd quartile minus the 2nd quartile greater than the 3rd quartile or 1.5 times the 3rd quartile minus the 2nd quartile less than the 2nd quartile. The 3rd quartile minus the 2nd quartile is known as the interquartile.

The formula for a positive outlier is

$$\text{greater than 3rd quartile} + 1.5 \text{ times the interquartile.}$$

The formula for a negative outlier is

$$\text{less than 2nd quartile} - 1.5 \text{ times the interquartile.}$$

4.7.2 Extreme Outliers

Extreme outliers use a multiplier of 3 instead of 1.5.

The formula for an extreme positive outlier is

$$\text{greater than 3rd quartile} + 3 \text{ times the interquartile.}$$

The formula for an extreme negative outlier is

$$\text{less than 2nd quartile} - 3 \text{ times the interquartile.}$$

4.7.3 Vertical Whiskers

The *box and whisker plot* has vertical lines extending up from the 3rd quartile and vertical lines extending down from the 2nd quartile. These vertical lines are called *whiskers*. Most people defined the two vertical whiskers as extending to the maximum number and the minimum number of a data set. An alternative definition that is used by Excel and more useful definition is for the top whisker to extend up from the top of the box to the largest data element that is less than or equal to 1.5 times the interquartile range and for the bottom whisker to extend down from the bottom of the box to the smallest data element that is larger than 1.5 times the interquartile. With this alternative definition, the whiskers represent data ranges that are not outliers. With this alternative definition, outliers are shown as data points above and below the whiskers.

4.7.4 Median and Mean

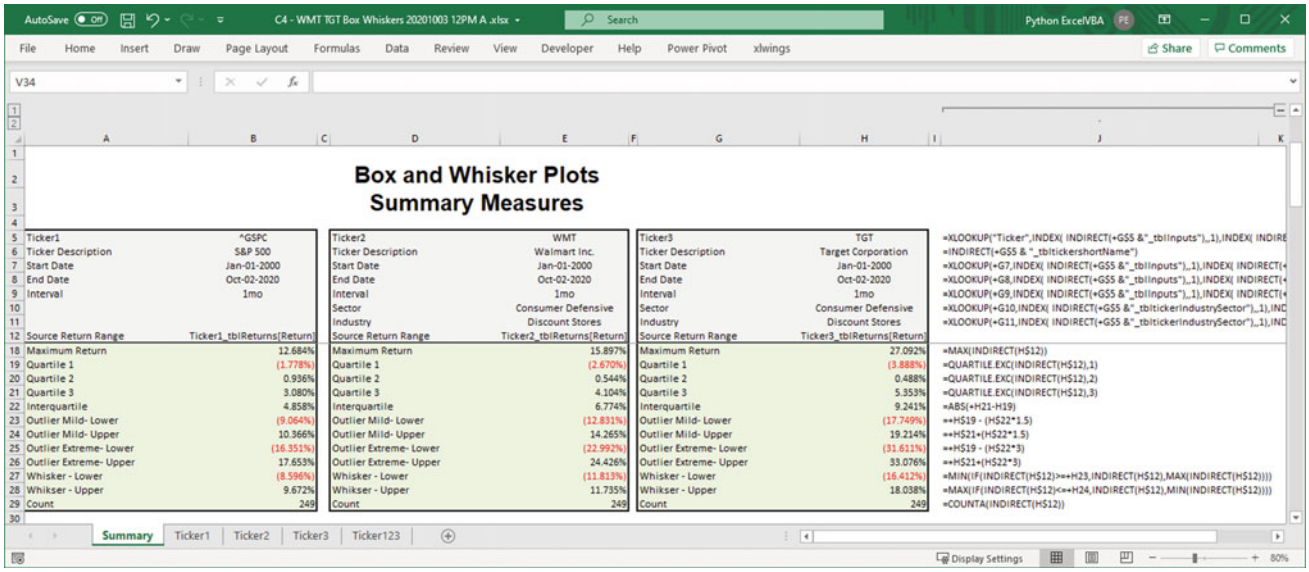
The horizontal line in the middle of the box and whisker plot is the start of the 3rd quartile. The horizontal line also represents the median and the 50th percentile of a data set. Excel's box and whisker plot also show an "x" for the mean. If the "x" is above the horizontal median line, then there are large positive numbers influencing the mean. If the "x" is below the horizontal median line, then there are large negative numbers influencing the median.

4.8 Excel Rate of Return Box and Whisker Workbook

In this section, we will use the Excel Rate of Return workbook to demonstrate and study the *box and whisker plot.* The workbook has a "Summary" sheet that has numerical statistics that is needed to create the *box and whisker plot* and has *box and whisker plots* for the S&P 500, Walmart, and Target. It has four return datasheets. The worksheets "Ticker1," "Ticker2," and "Ticker3" contain the rate of return workbook that we created in Chap. 3. The worksheet "Ticker123" has a table that combines the return data from the worksheets "Ticker1," "Ticker2," and "Ticker3." This combined table of return information is needed to create a combined *box and whisker plot*.

4.8.1 Summary Worksheet

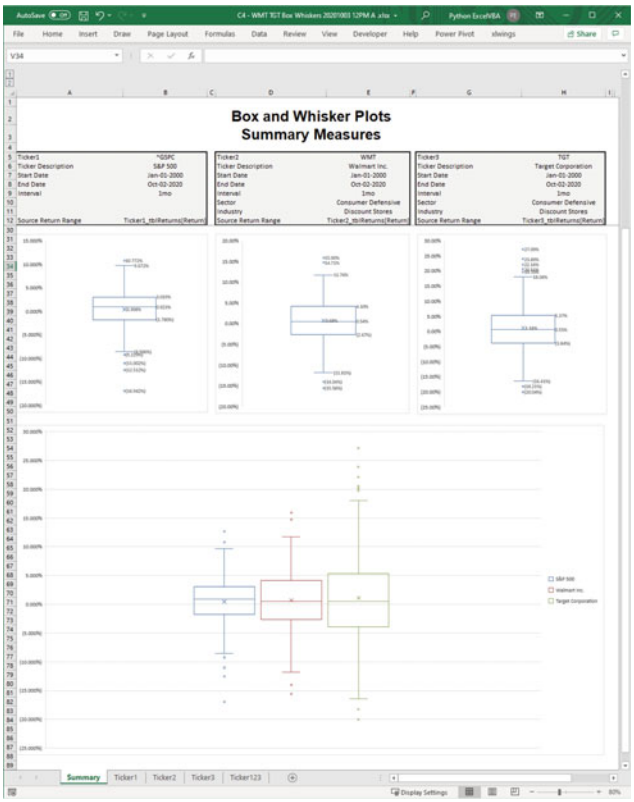
The top of the "Summary" sheet contains the numerical statistics needed to create and describe the *box and whisker plot.*



Box and Whisker Plots
Summary Measures

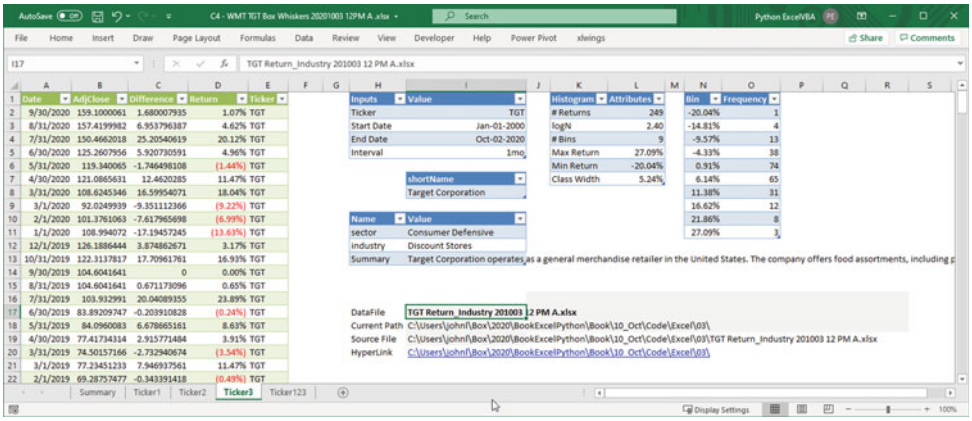
Ticker1	^GSPC	Ticker2	WMT	Ticker3	TGT	=XLOOKUP("Ticker",INDEX(INDIRECT(+G$5 &"_tblInputs"),,1),INDEX(INDIRE
Ticker Description	S&P 500	Ticker Description	Walmart Inc.	Ticker Description	Target Corporation	=INDIRECT(+G$5 & "_tblTickershortName")
Start Date	Jan-01-2000	Start Date	Jan-01-2000	Start Date	Jan-01-2000	=XLOOKUP(+G7,INDEX(INDIRECT(+G$5 &"_tblInputs"),,1),INDEX(INDIRECT(+
End Date	Oct-02-2020	End Date	Oct-02-2020	End Date	Oct-02-2020	=XLOOKUP(+G8,INDEX(INDIRECT(+G$5 &"_tblInputs"),,1),INDEX(INDIRECT(+
Interval	1mo	Interval	1mo	Interval	1mo	=XLOOKUP(+G9,INDEX(INDIRECT(+G$5 &"_tblInputs"),,1),INDEX(INDIRECT(+
		Sector	Consumer Defensive	Sector	Consumer Defensive	=XLOOKUP(+G10,INDEX(INDIRECT(+G$5 &"_tblTickerIndustrySector"),,1),IND
		Industry	Discount Stores	Industry	Discount Stores	=XLOOKUP(+G11,INDEX(INDIRECT(+G$5 &"_tblTickerIndustrySector"),,1),IND
Source Return Range	Ticker1_tblReturns[Return]	Source Return Range	Ticker2_tblReturns[Return]	Source Return Range	Ticker3_tblReturns[Return]	
Maximum Return	12.684%	Maximum Return	15.897%	Maximum Return	27.092%	=MAX(INDIRECT(H$12))
Quartile 1	(1.778%)	Quartile 1	(2.670%)	Quartile 1	(3.888%)	=QUARTILE.EXC(INDIRECT(H$12),1)
Quartile 2	0.936%	Quartile 2	0.544%	Quartile 2	0.488%	=QUARTILE.EXC(INDIRECT(H$12),2)
Quartile 3	3.080%	Quartile 3	4.104%	Quartile 3	5.353%	=QUARTILE.EXC(INDIRECT(H$12),3)
Interquartile	4.858%	Interquartile	6.774%	Interquartile	9.241%	=ABS(+H21-H19)
Outlier Mild- Lower	(9.064%)	Outlier Mild- Lower	(12.831%)	Outlier Mild- Lower	(17.749%)	=+H$19 - (H$22*1.5)
Outlier Mild- Upper	10.366%	Outlier Mild- Upper	14.265%	Outlier Mild- Upper	19.214%	=+H$21+(H$22*1.5)
Outlier Extreme- Lower	(16.351%)	Outlier Extreme- Lower	(22.992%)	Outlier Extreme- Lower	(31.611%)	=+H$19 - (H$22*3)
Outlier Extreme- Upper	17.653%	Outlier Extreme- Upper	24.426%	Outlier Extreme- Upper	33.076%	=+H$21+(H$22*3)
Whisker - Lower	(8.596%)	Whisker - Lower	(11.813%)	Whisker - Lower	(16.412%)	=MIN(IF(INDIRECT(H$12)>=+H23,INDIRECT(H$12),MAX(INDIRECT(H$12))))
Whikser - Upper	9.672%	Whikser - Upper	11.735%	Whikser - Upper	18.038%	=MAX(IF(INDIRECT(H$12)<=+H24,INDIRECT(H$12),MIN(INDIRECT(H$12))))
Count	249	Count	249	Count	249	=COUNTA(INDIRECT(H$12))

The bottom of the "Summary" sheet contains the *box and whisker plots* for the S&P 500, Walmart, and Target.

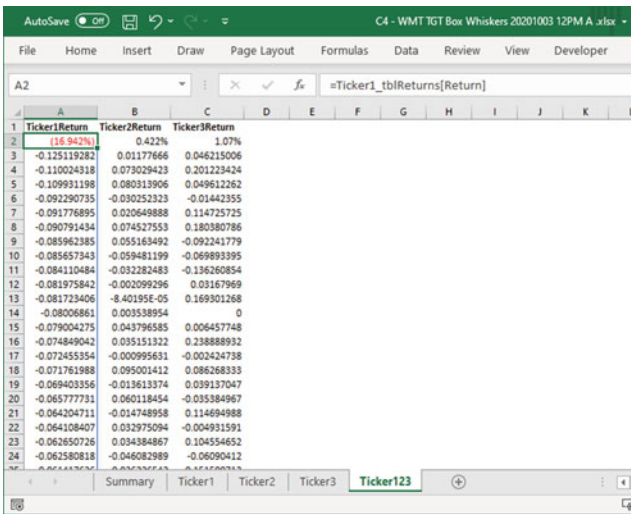


4.8.2 Ticker1, Ticker2, Ticker3 Worksheets

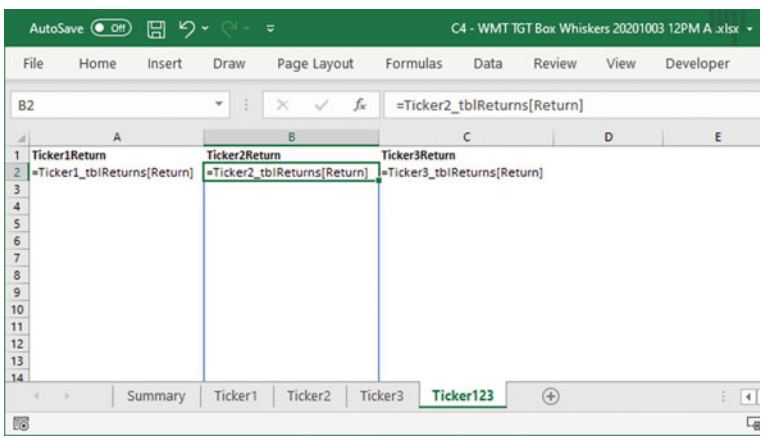
Ticker1, Ticker2, and Ticker3 worksheets contain return data for each ticker. The source data for each of these worksheets is a separate Excel workbook. That separate Excel workbook is the return workbook that we created in Chap. 3. We use *Power Query* to retrieve the data from the separate workbook. Below shows the worksheet from the "Ticker3" worksheet.



4.8.3 Ticker123 Worksheet



Ticker123 worksheet contains a table that a column for the returns of the other worksheets. Each column is populated by a dynamic array formula. Because cell a2 has a dynamic formula, selecting cell a2 will cause Excel to highlight the cells that the dynamic formula represents. Pressing the key combination *ctrl*+ ~ shows the dynamic formulas for cells a2, b3, b3. Dynamic formulas are only available in Excel 365.

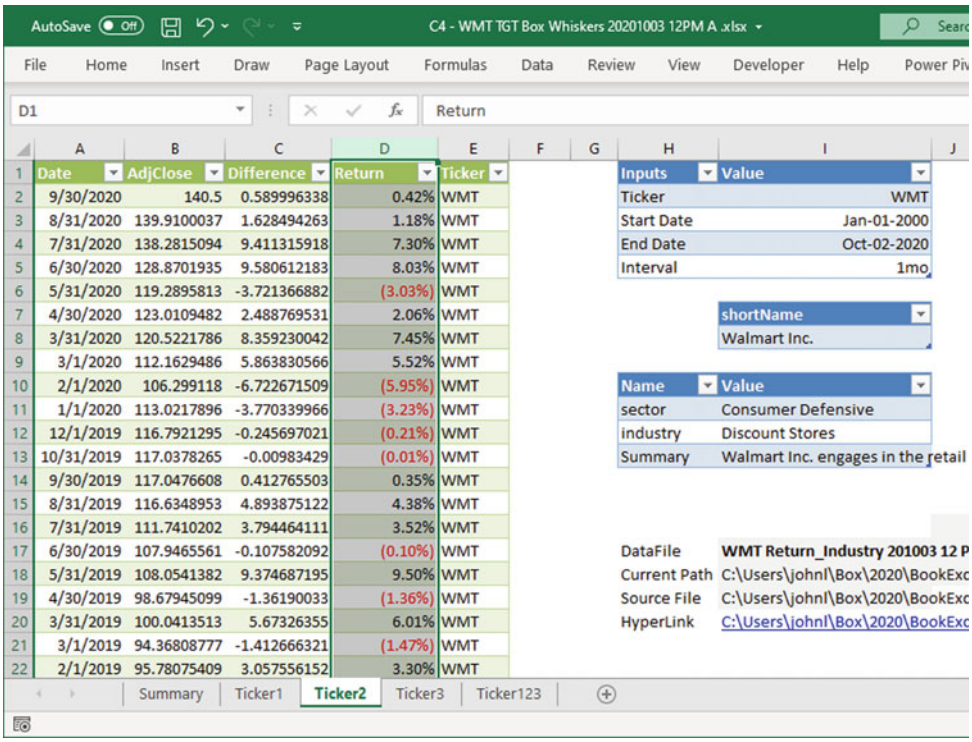


4.9 Creating Box and Whisker Plot in Excel

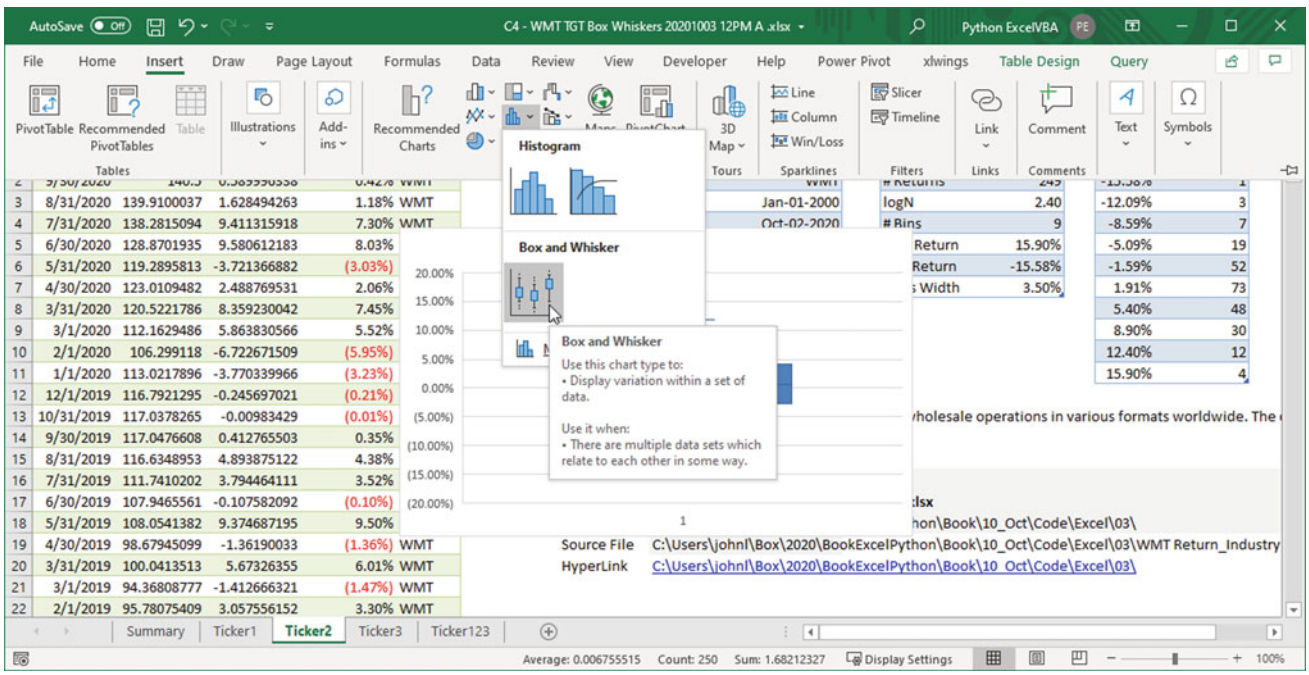
4.9.1 Single Box and Whisker Plot

In this section, we will create a *box and whisker plot* for the return data for the *Ticker2* worksheet.

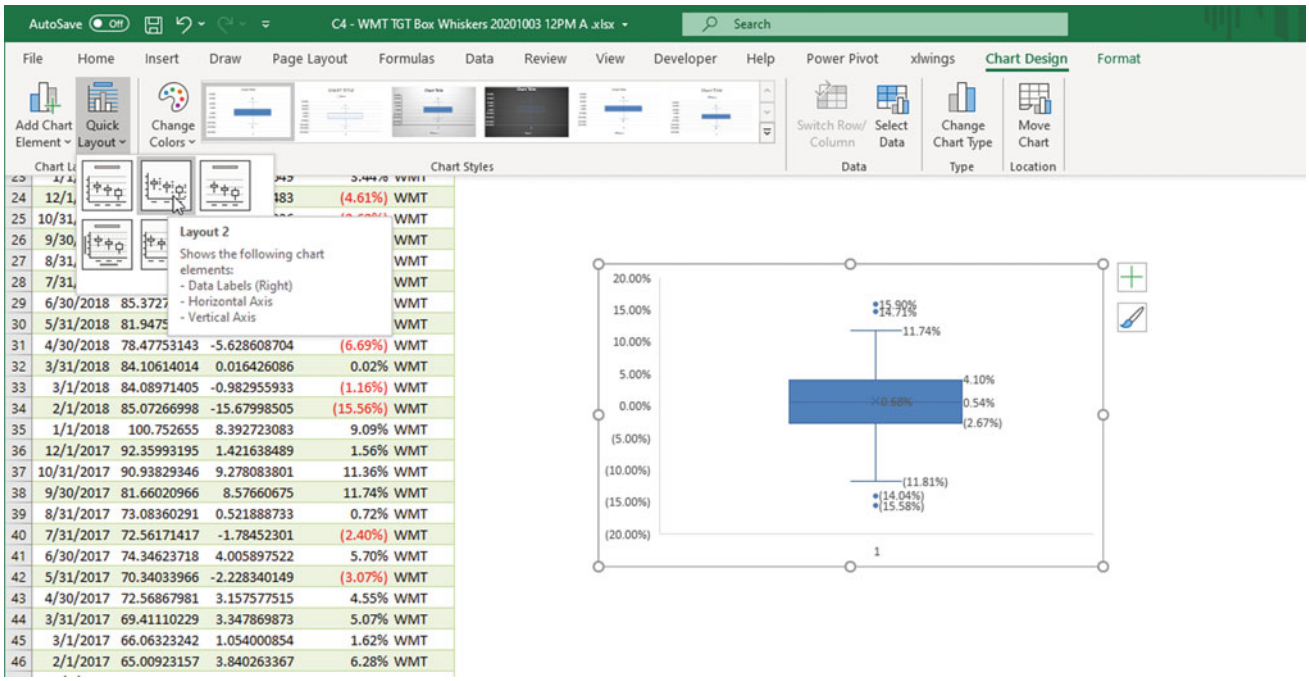
(1) Select the *Return* column.



(2) From the *Insert* tab, click on the *Box and Whisker* chart.

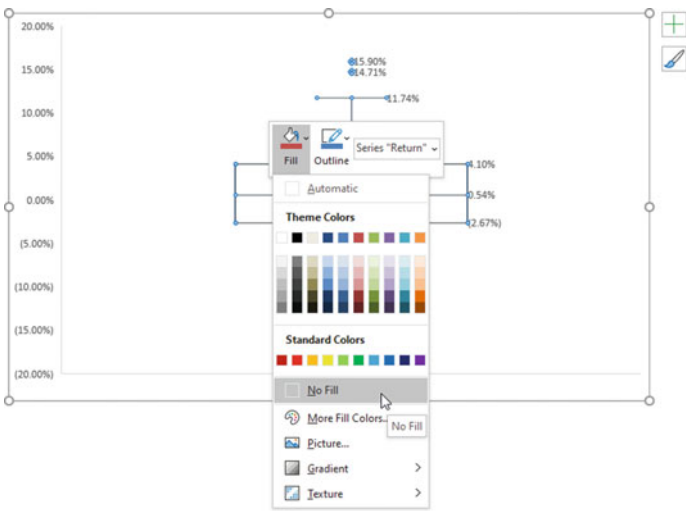


(3) From the *Chart Design* tab, choose *Layout 2* in the *Quick Layout.*

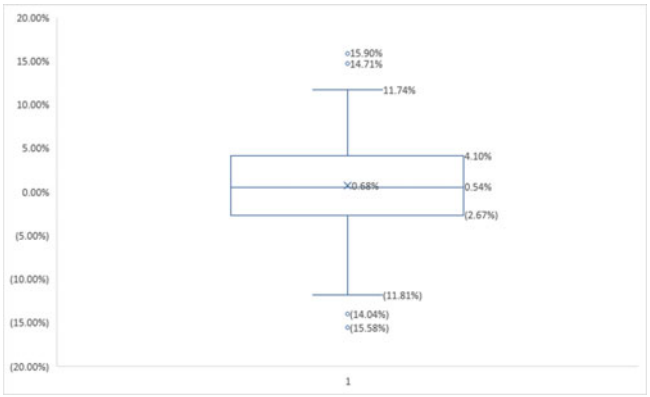


24	12/1/		483	(4.61%)	WMT	
25	10/31/				WMT	
26	9/30/				WMT	
27	8/31/				WMT	
28	7/31/				WMT	
29	6/30/2018	85.3727			WMT	
30	5/31/2018	81.9475			WMT	
31	4/30/2018	78.47753143	-5.628608704	(6.69%)	WMT	
32	3/31/2018	84.10614014	0.016426086	0.02%	WMT	
33	3/1/2018	84.08971405	-0.982955933	(1.16%)	WMT	
34	2/1/2018	85.07266998	-15.67998505	(15.56%)	WMT	
35	1/1/2018	100.752655	8.392723083	9.09%	WMT	
36	12/1/2017	92.35993195	1.421638489	1.56%	WMT	
37	10/31/2017	90.93829346	9.278083801	11.36%	WMT	
38	9/30/2017	81.66020966	8.57660675	11.74%	WMT	
39	8/31/2017	73.08360291	0.521888733	0.72%	WMT	
40	7/31/2017	72.56171417	-1.78452301	(2.40%)	WMT	
41	6/30/2017	74.34623718	4.005897522	5.70%	WMT	
42	5/31/2017	70.34033966	-2.228340149	(3.07%)	WMT	
43	4/30/2017	72.56867981	3.157577515	4.55%	WMT	
44	3/31/2017	69.41110229	3.347869873	5.07%	WMT	
45	3/1/2017	66.06323242	1.054000854	1.62%	WMT	
46	2/1/2017	65.00923157	3.840263367	6.28%	WMT	

(4) Right-mouse click on the chart and choose *No Fill.*



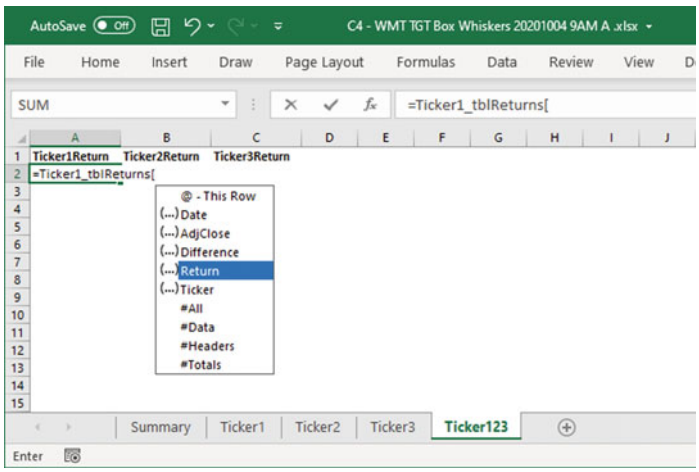
The resulting *box and whisker* chart for WMT shows that the non-outlier returns are in the range from 11.74 to −11.81%. There are two upper outlier returns and two lower outlier returns.



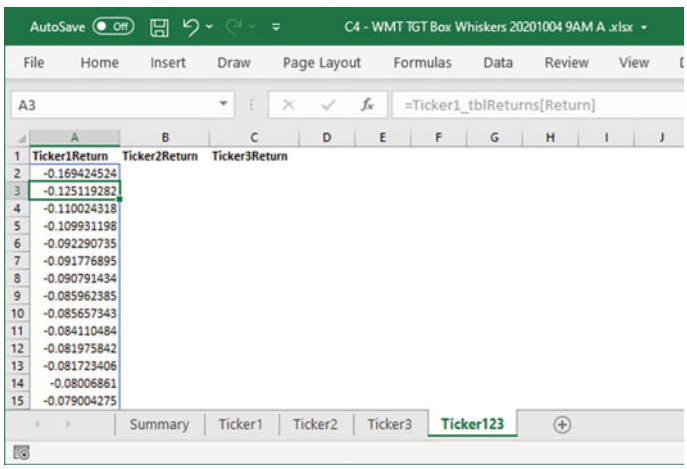
4.9.2 Combined Box and Whisker Plot

To create a combined box and whisker plot, we first need to create a table that has the returns of the 3 tickers in one table, each return in a column.

To get the return data for Ticker 1, enter the dynamic array range name *Ticker1_tblReturns[Return]* in cell a2.

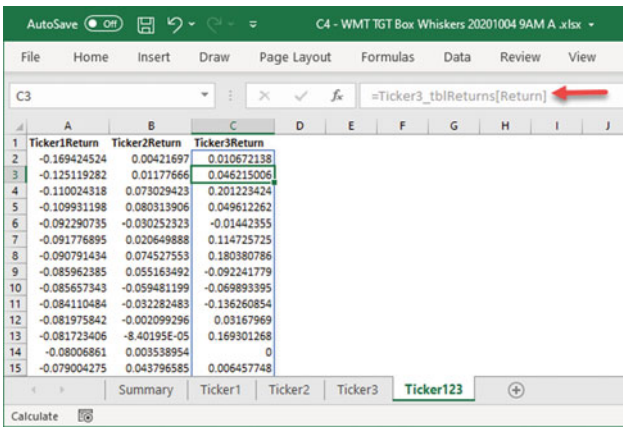


After entering the dynamic array range name, Excel will automatically fill column A with ticker1's return information. This is indicated by highlighting the data in column A.

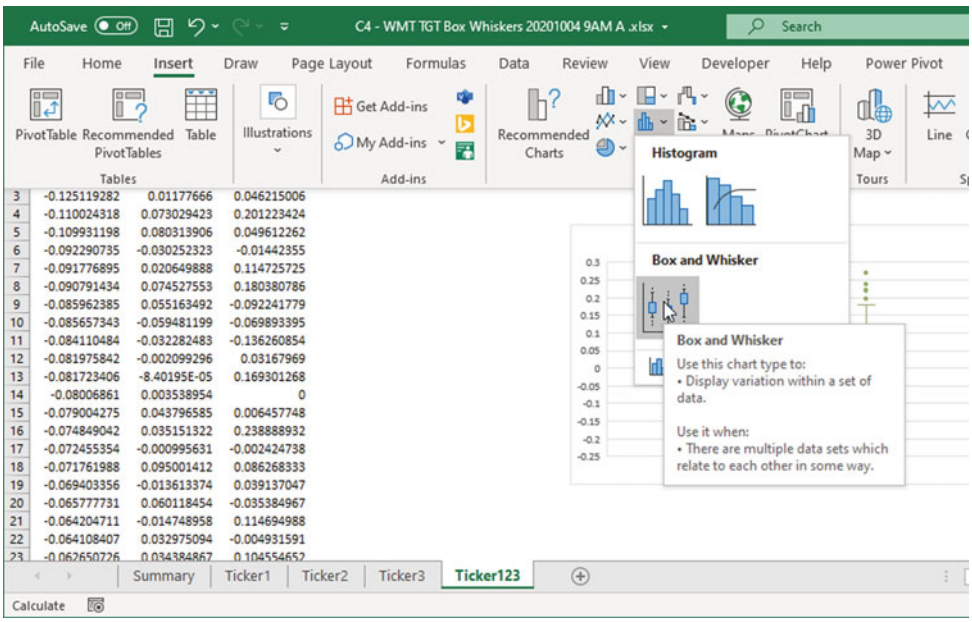


Do the same for ticker2's and ticker3's return information by entering the dynamic array range name *Ticker2_tblReturns [Return]* in cell b2 and *Ticker3_tblReturns[Return]* in cell c2.

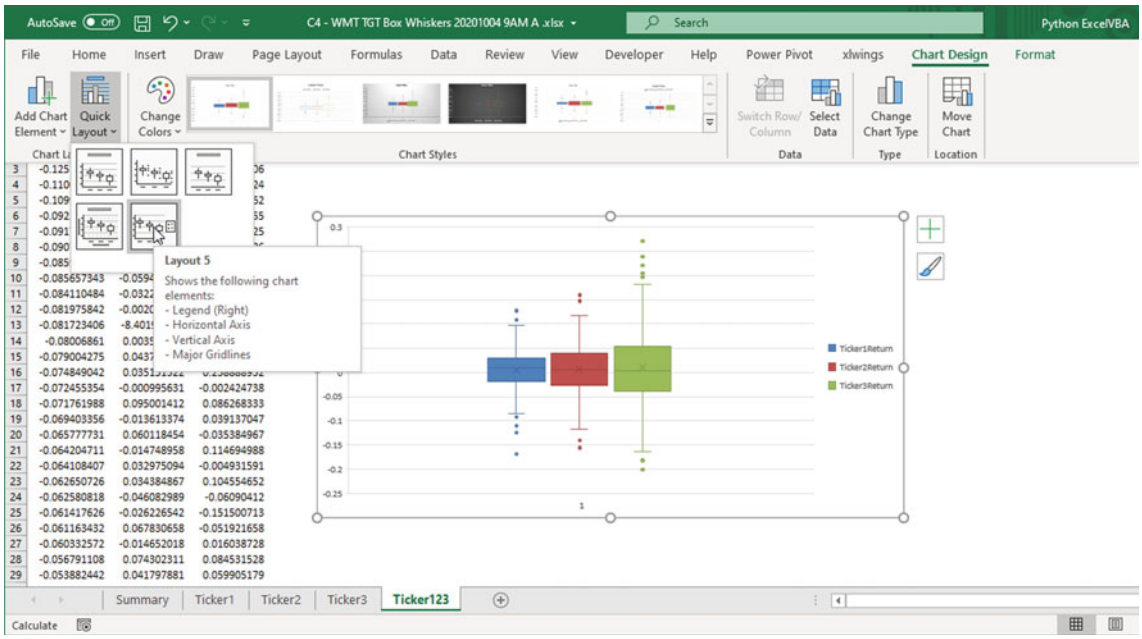
Selecting cell c3 you will notice that the formula bar shows a gray *Ticker3_tblReturns[Return]* dynamic formula. This indicates that cell c3 is part of the *Ticker3_tblReturns[Return]* dynamic formula.



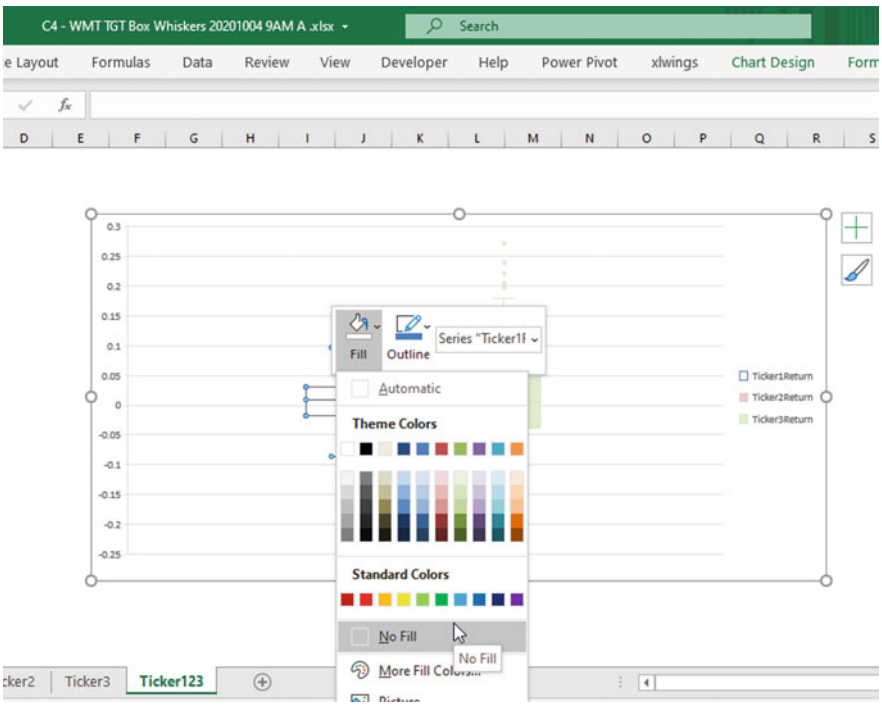
Next, select the *box and whisker* graph from the insert tab.



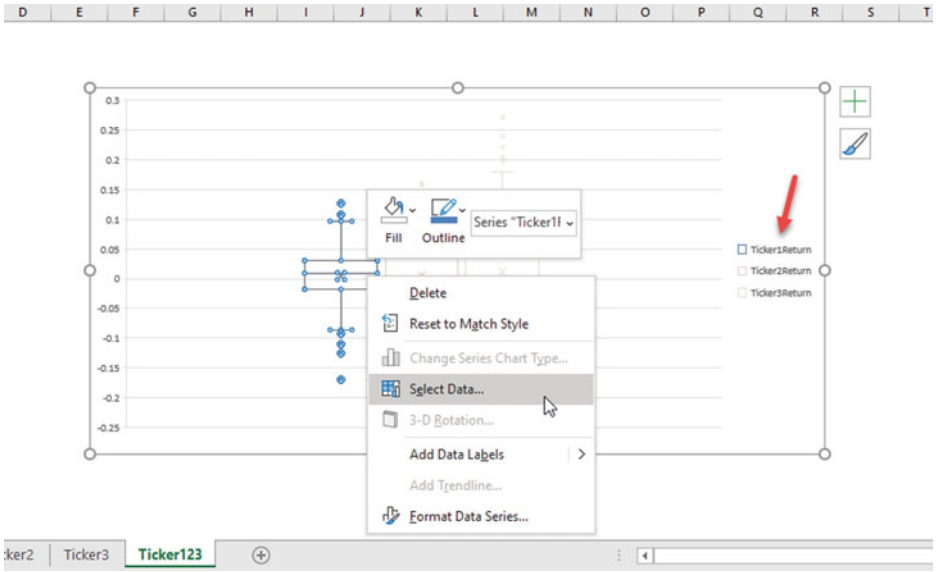
Next, choose *Layout 5* in *Quick Layout* from the *Chart Design* tab.



Right-mouse click on the first box and whisker chart and choose *No Fill.*

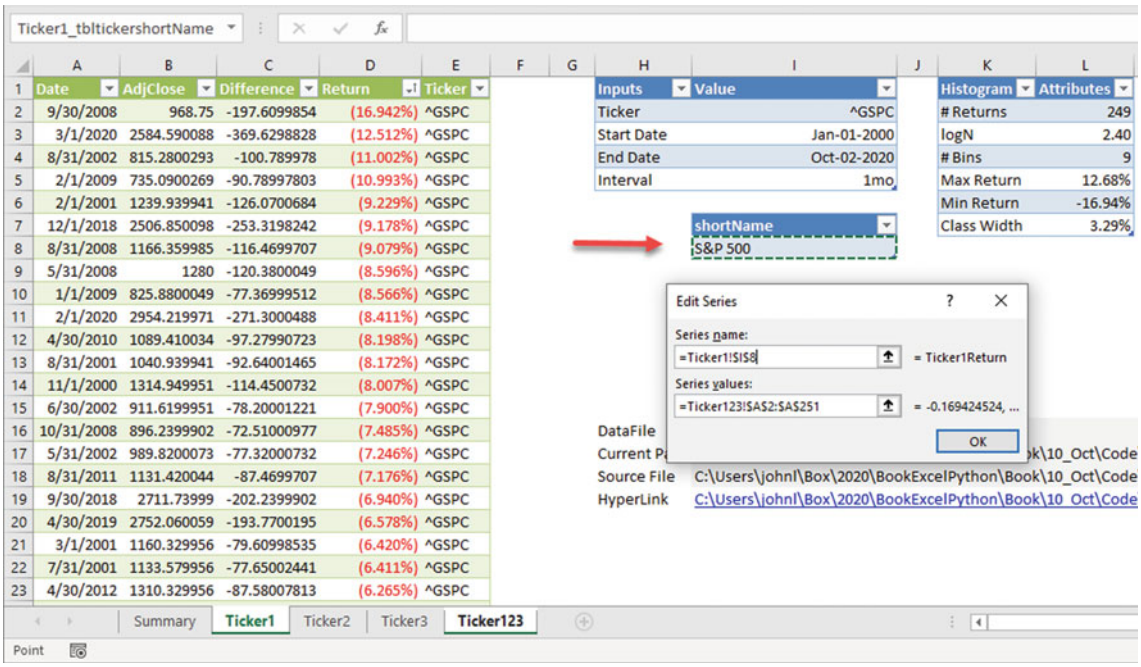


Do the same for the 2nd and 3rd box and whisker charts.
Let's now fix the legend label. Right-mouse click on the first box whisker chart and choose *Select Data.*



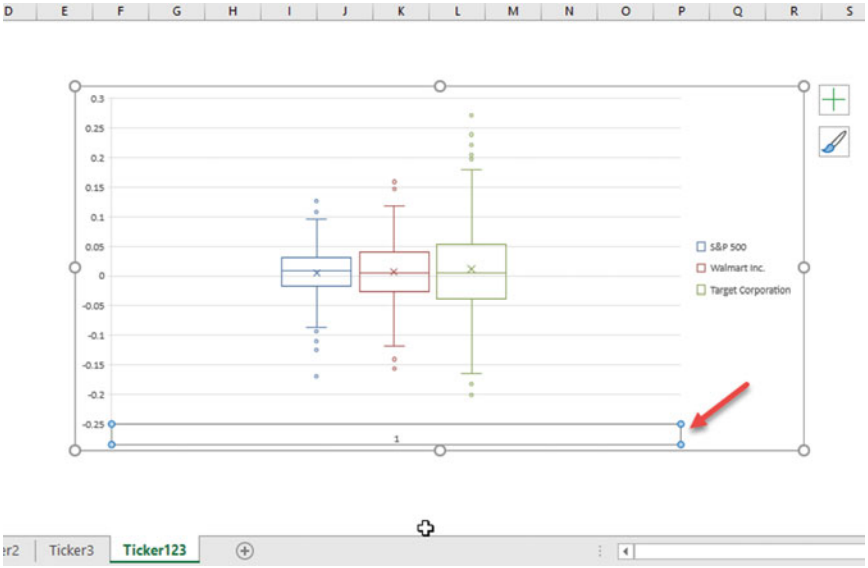
Select *Ticker1Return* and then click on *Edit.*

To make the shortName the *Series name*, indicate cell i8 in the sheet *Ticker1*.



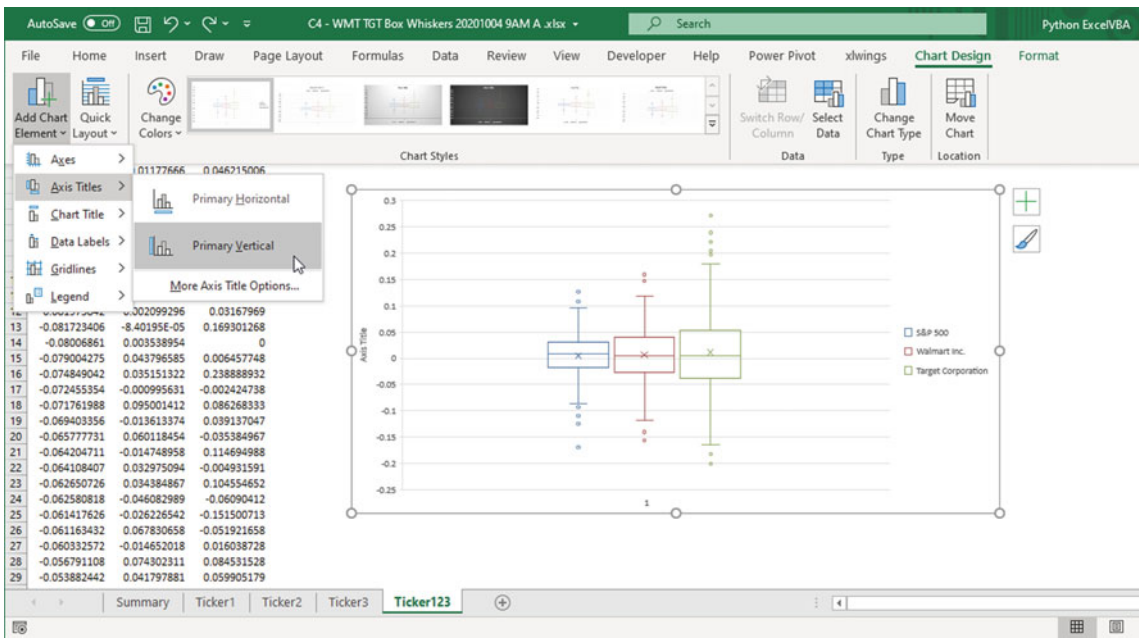
Do the same for ticker 2 and ticker 3 series names.

Highlight the x-label and then press the *delete* key to remove the 1 at the bottom of the chart.

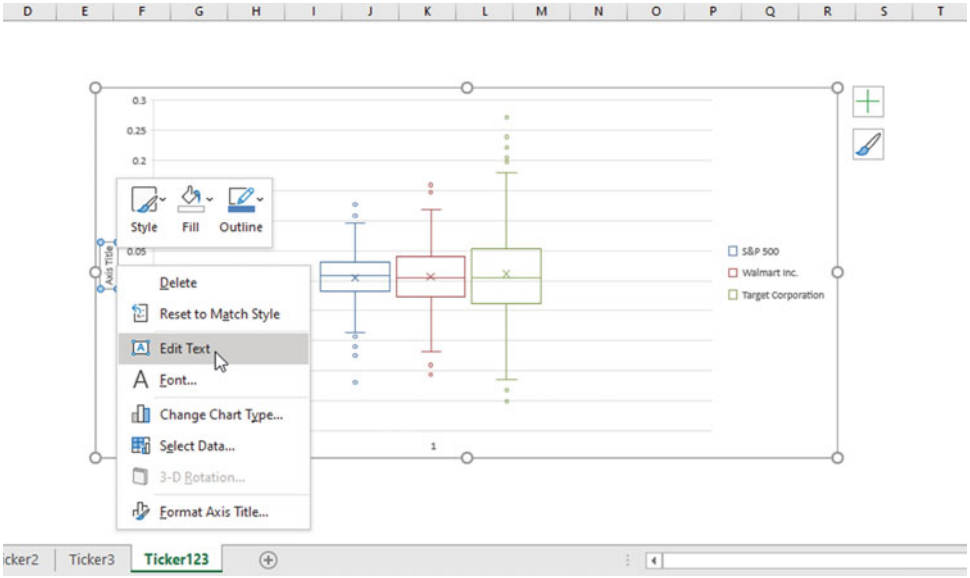


To add the vertical axis title, do the following:

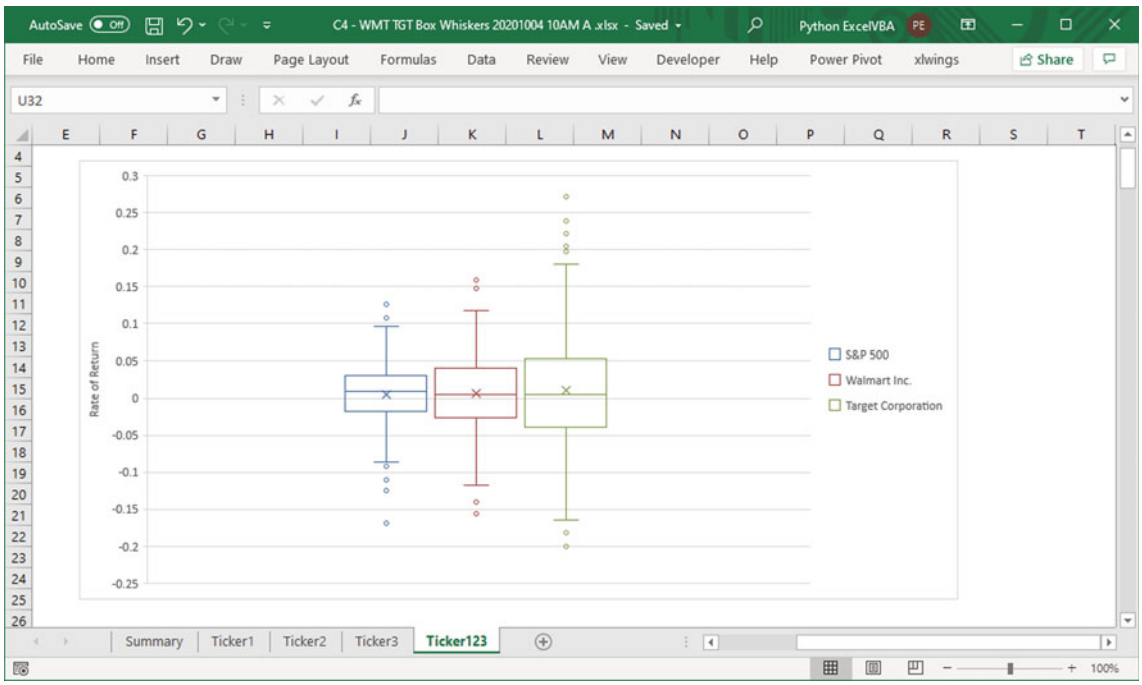
(1) Click on *Primary Vertical* from the *Add Chart Element* button in the *Chart Design* tab.



Click on the *Axis Title* and then choose *Edit Text*. Make the *Axis Title* "Rate of Return."



Below is the final combined box and whisker chart.



4.10 Using Python to Calculate the 5-Year Numerical Measures of the Rate of Return of AAPL, MSFT, and the S&P 500

The following is the Python code to calculate the 5-year numerical measures of the rate of return of AAPL, MSFT, and the S&P 500.

```
[20]  1 pip install yfinance
```

```
[21]  1 #Date: June 8, 2020
      2 import yfinance as yf
      3 import matplotlib.pyplot as plt
      4 import numpy as np
      5 import pandas as pd
      6 from scipy.stats import kurtosis, skew
      7 from tabulate import tabulate
      8 import datetime as dt#get current date
      9 from google.colab import files
```

```
1 class Analyze_Stock:
2
3   def __init__(self, Stock_Name, Start_T, End_T, T_Interval):
4
5     self.__Stock_price = yf.download(Stock_Name,start=Start_T,End=End_T,interval = T_Interval)['Adj Close'].dropna()
6     self.__Ret = self.__Stock_price.pct_change()[1:]
7
8     monthly_data = yf.download(Stock_Name,start=Start_T,End=End_T,interval = '1mo')['Adj Close'].dropna()
9     slice_object = slice(1, len(monthly_data), 12)
10
11    self.__Yr_Ret = monthly_data[slice_object].pct_change()[1:]
12    self.__count = len(self.__Stock_price)
13    self.stock = Stock_Name
14    self.start = Start_T
15    self.end = End_T
16    self.frequency = T_Interval
17  def __helper(self,N):
18    return '{:0.3%}'.format(N)
```

```
19
20  def __helper_2(self,N):
21    return '{:0.2f}'.format(N)
22
23  def __helper_4(self,N):
24    return '{:0.4}'.format(N)
25
26  def Mean_Ret(self):
27    mean = np.mean(self.__Ret)
28    return mean
29
30  def Median_Ret(self):
31    median = np.median(self.__Ret)
32    return median
33
34  def STD_Ret(self):
35    std = np.std(self.__Ret)
36    return std
37
38  def Var_Ret(self):
39    Var = np.var(self.__Ret)
40    return Var
```

```
def Annual_Mean_Ret(self):
  mean = np.mean(self.__Yr_Ret)
  return mean

def Annual_Median_Ret(self):
  median = np.median(self.__Yr_Ret)
  return median

def Annual_Std(self):
  std = np.std(self.__Yr_Ret)
  return std

def Annual_Var(self):
  var = np.var(self.__Yr_Ret)
  return var

def Min_Ret(self):
  Min = min(self.__Yr_Ret)
  return Min
```

```
def Max_Ret(self):
  Max = max(self.__Yr_Ret)
  return Max

def Range_Ret(self):
  Range = max(self.__Yr_Ret)-min(self.__Yr_Ret)
  return Range

def Min_Price(self):
  Min = min(self.__Stock_price)
  return Min

def Max_Price(self):
  Max = max(self.__Stock_price)
  return Max

def Range_Price(self):
  Range = max(self.__Stock_price)-min(self.__Stock_price)
  return Range

def quarter_tile(self):
  quarter = np.quantile(self.__Ret,0.25)
  return quarter

```

```
def half_tile(self):
  half = np.quantile(self.__Ret,0.5)
  return half

def Tquarter_tile(self):
  Tquarter = np.quantile(self.__Ret,0.75)
  return Tquarter

def interquatile(self):
  Int_quarter = np.quantile(self.__Ret,0.75)-np.quantile(self.__Ret,0.25)
  return Int_quarter

def kurt(self):
  Kurt = kurtosis(self.__Stock_price)
  return Kurt

def skewness(self):
  Skew = skew(self.__Stock_price)
  return Skew

def get_count(self):
  return self.__count

def CoV(self):
  cov = np.std(self.__Ret)-np.mean(self.__Ret)
  return cov

```

```
##calculate correlation coefficient
  def Market_corr(self):
    SNP = yf.download('^GSPC',start=self.start,End=self.end,interval = self.frequency)['Adj Close'].dropna()
    corr = np.corrcoef(SNP,self.__Stock_price)
    return corr[0,1]

  def Get_summary(self):
    dict_data = {
    'Measures' :['Start Date','End Date','Frequency','Mean Return', 'Median Return', 'Standard deviation', 'Variance',\
                 'Annual Mean Return','Median Annual Return','Annual Standard Deviation', \
                  'Annual Variance','Minimum Return','Maximum Return','Range Return','Minimum Price',\
                 'Maximum Price','Range Price','Quatile 25%','Quatile 50%','Quatile 75%','Interquartile',\
                  'Kurtosis','Skewness','Count','Coefficient of Variation','Market corr with S&P 500'],
    self.stock : [self.start, self.end,self.frequency, self.__helper(self.Mean_Ret()),self.__helper(self.Median_Ret()),\
                  self.__helper(self.STD_Ret()),self.__helper(self.Var_Ret()),self.__helper(self.Annual_Mean_Ret()),\
                  self.__helper(self.Annual_Median_Ret()),self.__helper(self.Annual_Std()),self.__helper(self.Annual_Var()
                  self.__helper(self.Min_Ret()),self.__helper(self.Max_Ret()),self.__helper(self.Range_Ret()),self.__helpe
                  self.__helper_2(self.Max_Price()),self.__helper_2(self.Range_Price()),self.__helper(self.quarter_tile())
                  self.__helper(self.Tquarter_tile()),self.__helper(self.interquatile()),self.__helper_4(self.kurt()),\
                  self.__helper_4(self.skewness()),self.get_count(),self.__helper_4(self.CoV()),self.__helper_4(self.Marke
    }
    frame_data = pd.DataFrame(dict_data)
    #print(self.__Stock_price)
    #print(self.__Ret)
    return frame_data
```

```
[29]
##allow user to input start&end date and interval
def Usr_input():
  #initialize dates
  previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
  Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
  years_ago = dt.date.today()-dt.timedelta(days=5 * 365)
  Default_start = years_ago.strftime("%Y-%m-%d")

  # request user input
  # set default values if users does not enter a value
  Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
  End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
  T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'

  print("")
  return Start, End, T_interval
```

```
[30]
##allow user input the stock name
def Stock_input():
  Stock = []
  Stock.append(input('Please enter ticker '+str(1)+'(default be AAPL)-->') or 'AAPL') # use or operation the set default input
  Stock.append(input('Please enter ticker '+str(2)+'(default be MSFT)-->') or 'MSFT')
  return Stock

##return the summary of stock report
def get_all_summary():

  stocks=Stock_input()

  (macro_start, macro_end, macro_T_interval) = Usr_input()

##get report from SNP market report
  Market_ind = Analyze_Stock('^GSPC',macro_start,macro_end,macro_T_interval)
  M_summary = Market_ind.Get_summary()

  summary_append = []
  summary_append.append(M_summary)
##append stock report to it
  for i in range(2):
    test = Analyze_Stock(stocks[i],macro_start,macro_end,macro_T_interval)
    summary = test.Get_summary()
    summary_append.append(summary[test.stock])

  summary_append = pd.concat(summary_append,axis=1)
  return summary_append,stocks,macro_start, macro_end, macro_T_interval
```

```
[31] 1 #Generate summary measure report
     2 #ask for user input
     3 (Total_summary,stocks,macro_start, macro_end, macro_T_interval) = get_all_summary()
     4
     5 ##insert sp500 to stocks
     6 stocks.insert(0,'^GSPC')
     7
```

```
Please enter ticker 1(default be AAPL)-->AAPL
Please enter ticker 2(default be MSFT)-->MSFT
Enter start date(Default = 5 Years ago, 2016-04-11)-->2016-04-11
Enter end date(Default = Yesterday, 2021-04-09)-->201-04-09
Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo

[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
```

```
1 print("")
2 print('             Numerical Summary Measures of Rate of Returns')
3 print(' ')
4 print(tabulate(Total_summary, showindex=False, headers=Total_summary.columns,tablefmt="github"))
```

Numerical Summary Measures of Rate of Returns

Measures	^GSPC	AAPL	MSFT
Start Date	2016-04-11	2016-04-11	2016-04-11
End Date	201-04-09	201-04-09	201-04-09
Frequency	1mo	1mo	1mo
Mean Return	1.228%	3.289%	2.917%
Median Return	1.790%	4.403%	2.640%
Standard deviation	4.274%	8.265%	4.757%
Variance	0.183%	0.683%	0.226%
Annual Mean Return	10.310%	44.782%	43.943%
Median Annual Return	10.195%	42.031%	41.964%
Annual Standard Deviation	3.828%	28.872%	6.512%
Annual Variance	0.147%	8.336%	0.424%
Minimum Return	5.389%	8.548%	38.054%
Maximum Return	15.463%	86.517%	53.788%
Range Return	10.074%	77.969%	15.734%
Minimum Price	2096.95	22.36	47.25
Maximum Price	4128.80	133.00	255.85
Range Price	2031.85	110.64	208.60
Quatile 25%	-0.060%	-1.631%	0.195%
Quatile 50%	1.790%	4.403%	2.640%
Quatile 75%	3.453%	9.095%	5.698%
Interquartile	3.513%	10.726%	5.503%
Kurtosis	0.1441	-0.05432	-0.7474
Skewness	0.7535	1.124	0.6674
Count	61	61	61
Coefficient of Variation	0.03047	0.04976	0.0184
Market corr with S&P 500	1.0	0.9411	0.9525

```
[9] 1 #download summary measures as a csv file
    2 Total_summary.to_csv('Chapter4_Summary.csv')
    3 files.download('Chapter4_Summary.csv')
```

4.11 Summary

Numeric measures look at a data set from different viewpoints. The five different views points are given below:

1. Measure of Central Tendency or Measure of Location:
   a. arithmetic mean,
   b. median.

   This group tells us the central location of a data set.

2. Measure of Spread:
   a. standard deviation,
   b. variance.

   This group tells us how to spread out the data set is.

3. Measure of Relative Position:
   a. quartiles,
   b. Z-score.

   This group tells us the relative position of a data value in a data set.

4. Measure of Shape:
   a. Skewness.

   This measure tells us if a data set is symmetrical relative to the mean or median.

5. Measure of Linear Relationship:
   a. Coefficient of Correlation.

   This measure tells us the linear relationship between two variables.

This chapter demonstrated these concepts on the rate of return data of Apple, Microsoft, Target, Walmart, and the S&P 500. The data source for the rate of return data was Yahoo! Finance.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.,” Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.,” BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.,” O'Reilly Media, 2017.

Probability Concepts and Their Analysis

5

5.1 Introduction

In Sect. 5.2 of this chapter, we discuss data presentation. Section 5.3 discusses probability, and Sect. 5.4 talks about combinations. Permutations are discussed in Sect. 5.5 and, finally, we summarize the chapter in Sect. 5.6.

5.2 Data Presentation

In the previous chapters, we studied how to analyze and extract information from a data set. It was implied that we had all the data items that we need. We will now deal with situations where this is not the case. We will be unable to get all the data items we are interested in analyzing because of prohibitive costs. The cost could be time, money, or effort.

Because we had all the data items that we need, we could make definitive statements like the data set's mean is 50. But now we will have to change our definitive statement to some like there is a 90% chance that the mean of the data set is 50. Interestingly enough, it is not necessary to have a definitive answer before making an informed educational decision.

What we will be doing is, in a systematic way, obtain a partial set of the data set of interest and then, from the partial data set, make statements about the complete data set. Loosely, we can call the partial data set of interest the *sample data*. Also, we can loosely call the complete data set the *population data*. The study on how to do this is called *inferential statistics*.

The first thing that we will need to study is to study *probability*. Interesting probability is the opposite of inferential statistics. In probability, we have the population data. We are interested in knowing what the probability is in getting specific sample data.

So inferential statistics is basically

$$\text{Sample Data} \rightarrow \text{Population data.}$$

And, the probability is basically

$$\text{Population} \rightarrow \text{Sample Data.}$$

5.3 Probability

One of the most fundamental concepts in probability is that the probability of the total number of possible outcomes equals 100% (many times referred to as 1). The probability of an item is between 0 and 100% (i.e., between 0 and 1).

As a corollary, the sum of the probability of all the items of the total number of possible outcomes equals one.

Suppose there are only two items, A and B. Then

$$\text{probability (A)} + \text{probability (B)} = 1$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_5

Therefore, if we know the probability of B, we should know the probability of A by rearranging the above formula to

$$\text{probability (A)} = 1 - \text{ probability (B)}$$

An underlying issue is to find out the total number of possible outcomes and know what items we are interested in and fin the percentage of these items compared to the total number of possible outcomes.

$$\text{Probability of an event} = \frac{\text{items of interest}}{\text{total number of possible outcomes}}$$

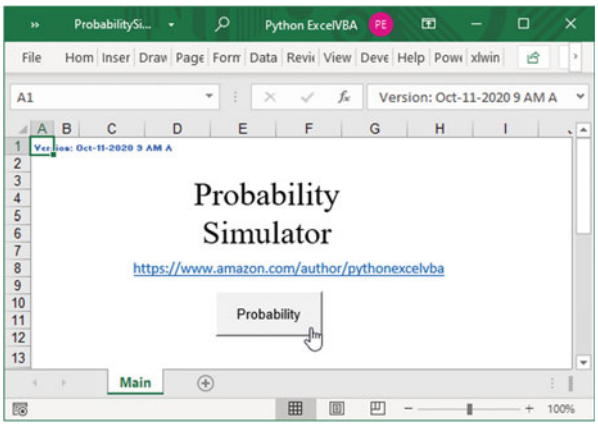
For example, the total number of possible outcomes of tossing a die is six because there are only six possible numbers. The probability of tossing the number three, therefore, would be 1/6 = 0.17. If we change to two dice, then the total number of possible outcomes would be 12. If we ask what the chance of getting a three when we throw the two dice is, the probability would be 2/12 = 0.17.

What does a probability of an event mean? There is a good chance if we roll a dice ten times, the number three never shows up, or the number three shows up five times. This is not in line with our expectation that the probability of number three from a dice is 0.17.

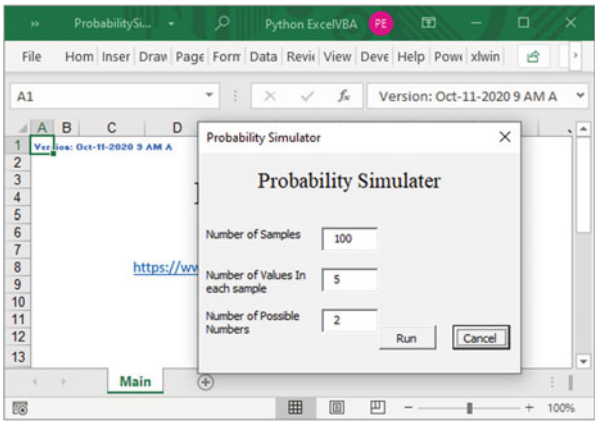
What is important to note is that the number three will approach 0.17 as we increase the number of times we roll the dice. A computer simulation best shows this. Below is the computer simulation in Excel.

5.3.1 Probability Simulation with Excel VBA

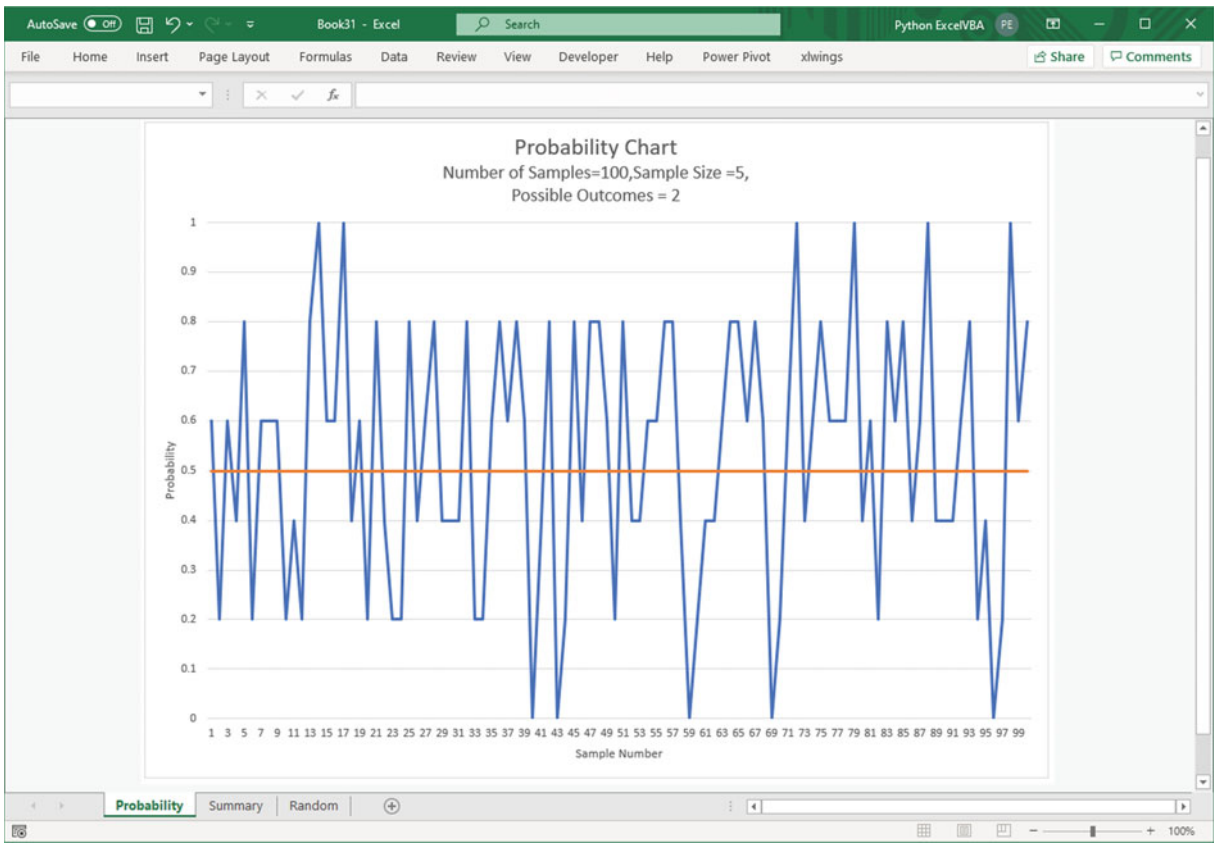
We will do some computer simulations to illustrate the probability concept. For example, if we were to flip a coin 5, 10, 50, 100, 500, and 5000 times, will we see 50% heads for each try? We will create an Excel Visual Basic for Application (VBA) macro to simulate the flipping of coins and chart the results.



Pressing the *Probability* button will show the *Probability Simulator* dialog box.



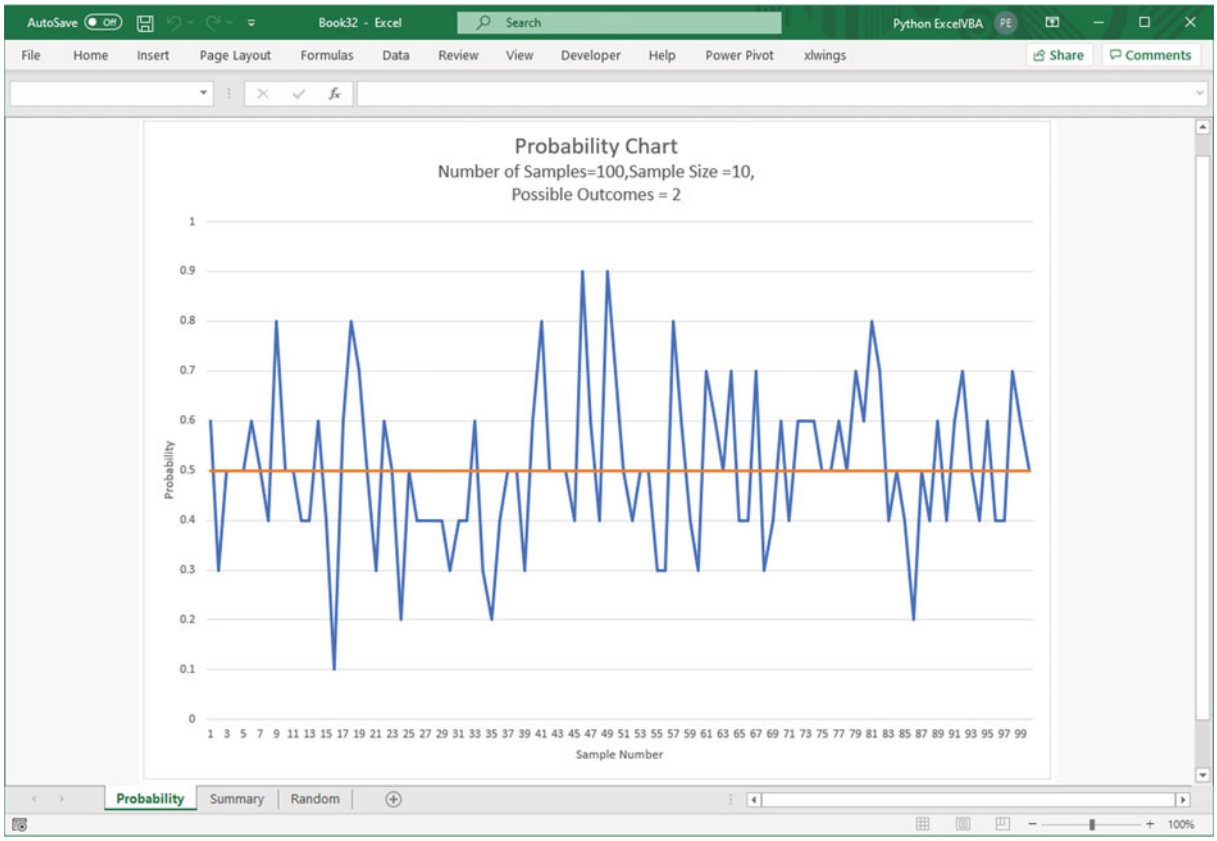
We will first generate 100 samples and, in each sample, have 5 values. To do this, fill out the *Probability Simulator* dialog box as shown above and then press the *Run* button to get a chart similar to the following.



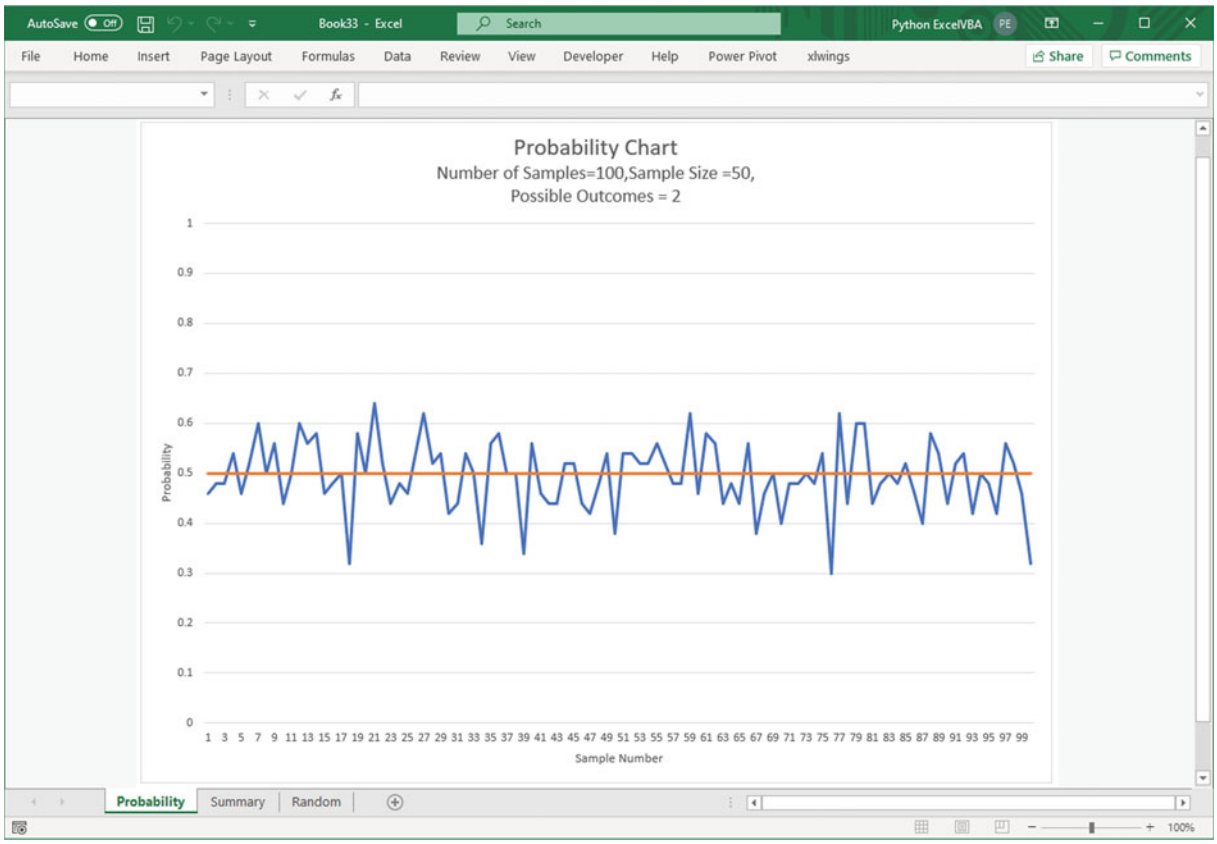
In the above chart, there is a line across the value 0.5 because that is the theoretical probability when there are two possible numbers with an equal probability of occurring.

We will now demonstrate what happens when we increase the sample size of each sample. We will do this by showing the chart with the sample size 10, then a sample size of 50, then a sample size of 100, and a sample size of 500.

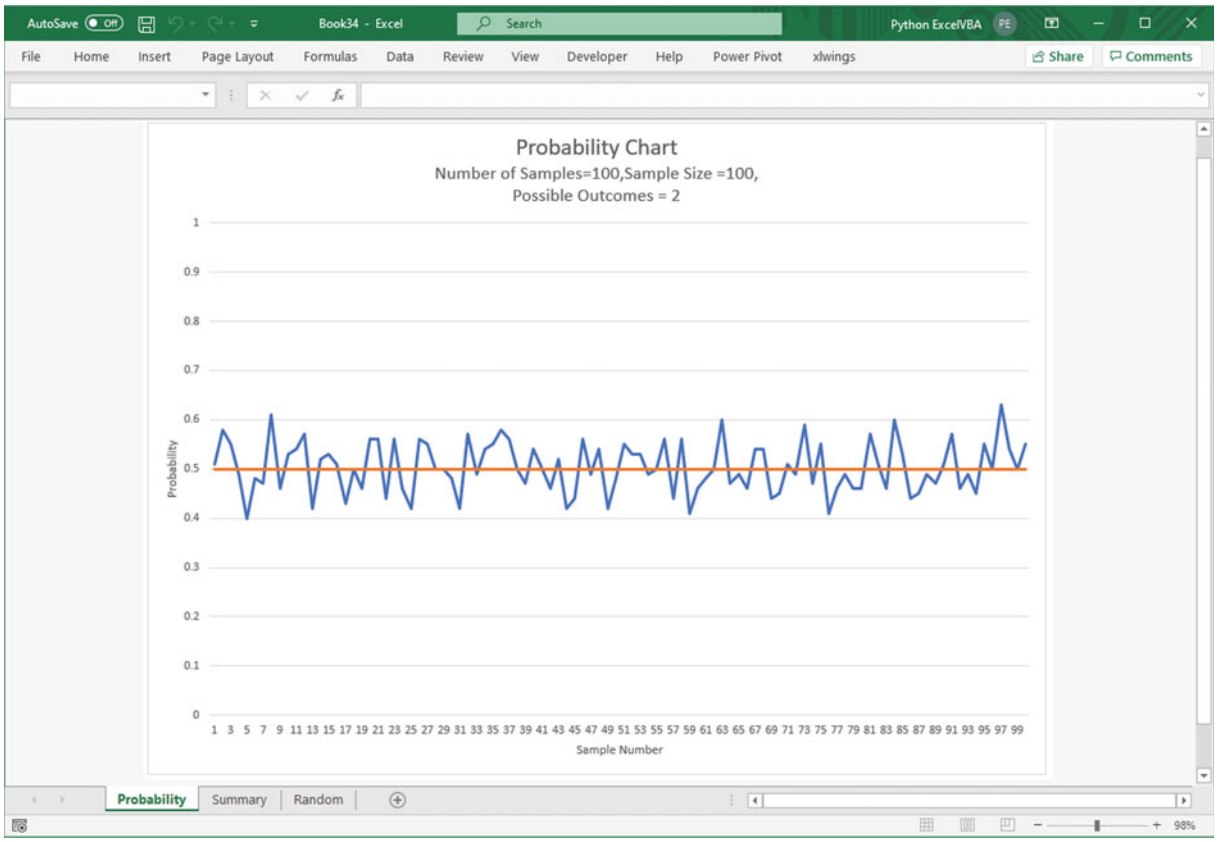
The following chart graphs the result of simulating the flipping of a coin ten times in 100 tries.



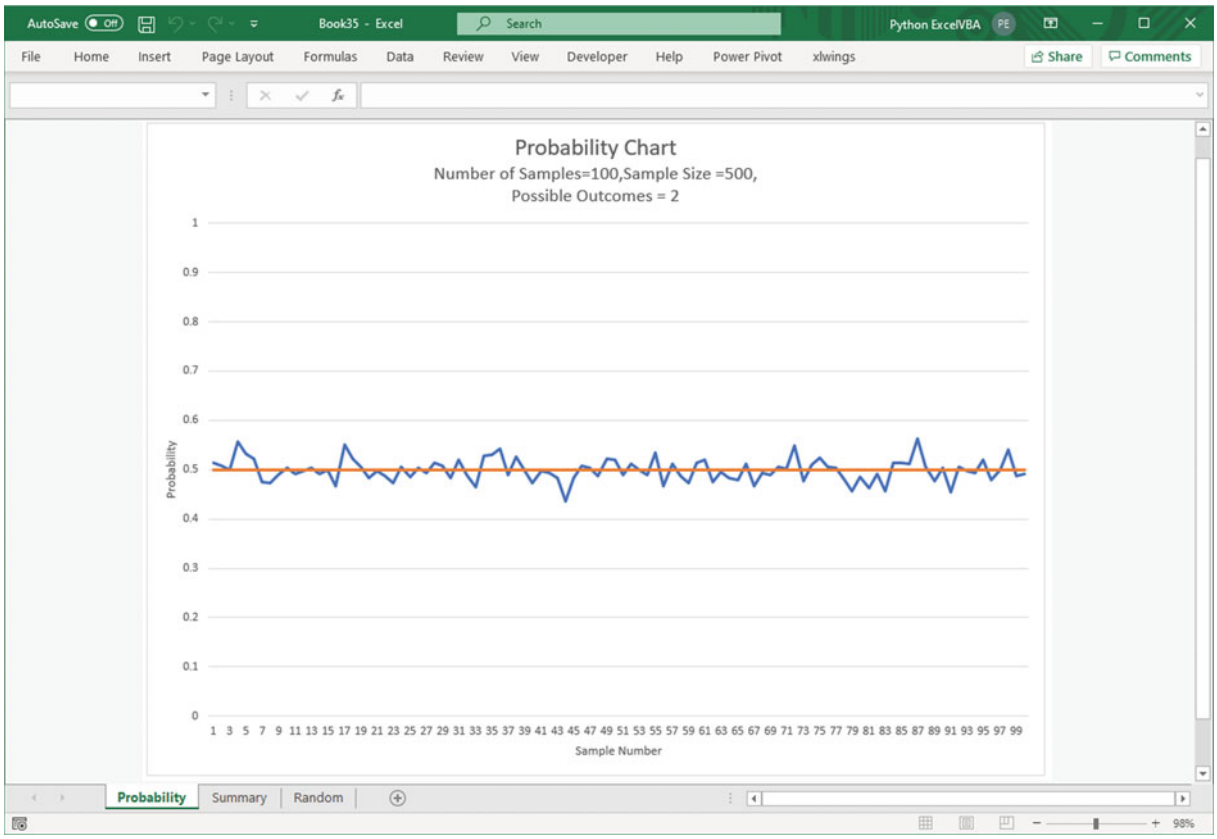
The following chart graphs the result of simulating the flipping of a coin 50 times in 100 tries.



The following chart graphs the result of simulating the flipping of a coin 100 times in 100 tries.

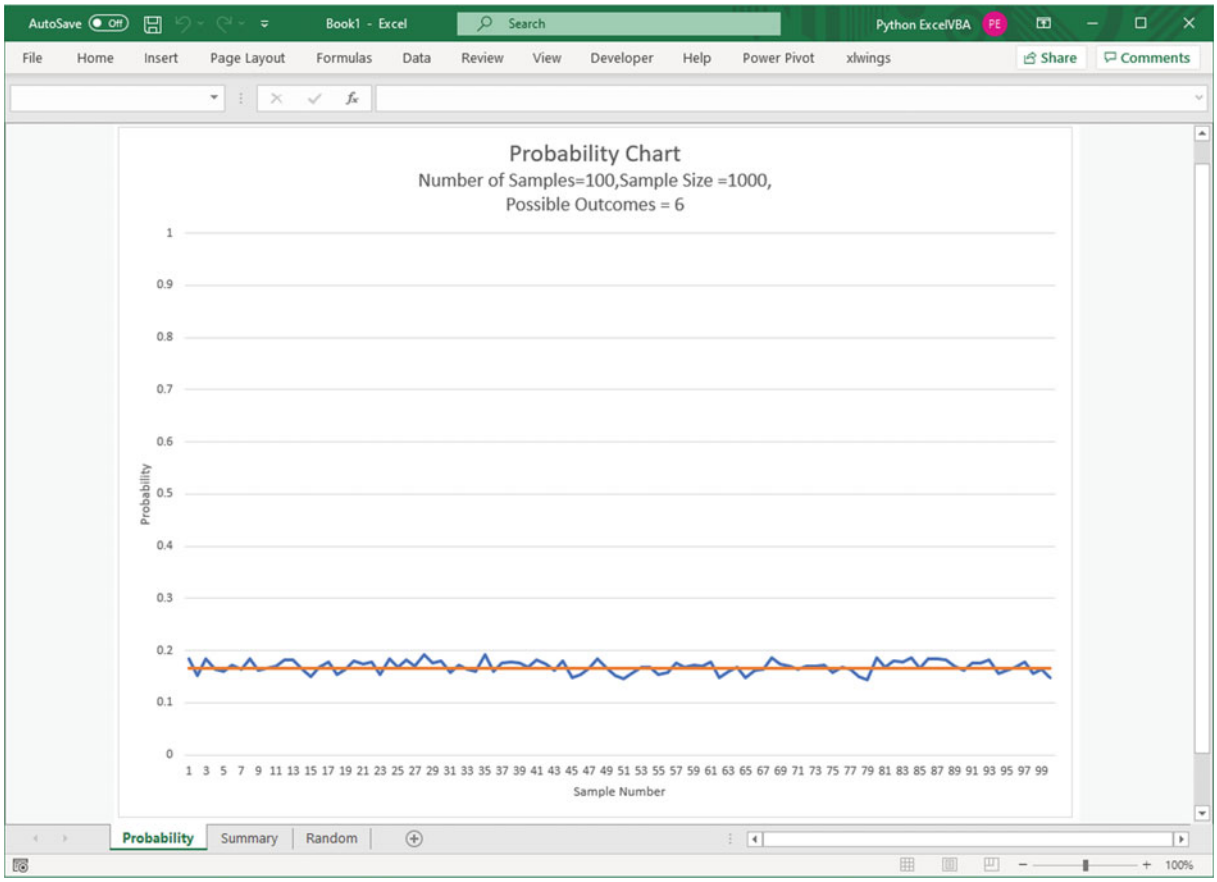


The following chart graphs the result of simulating the flipping of a coin 500 times in 100 tries.



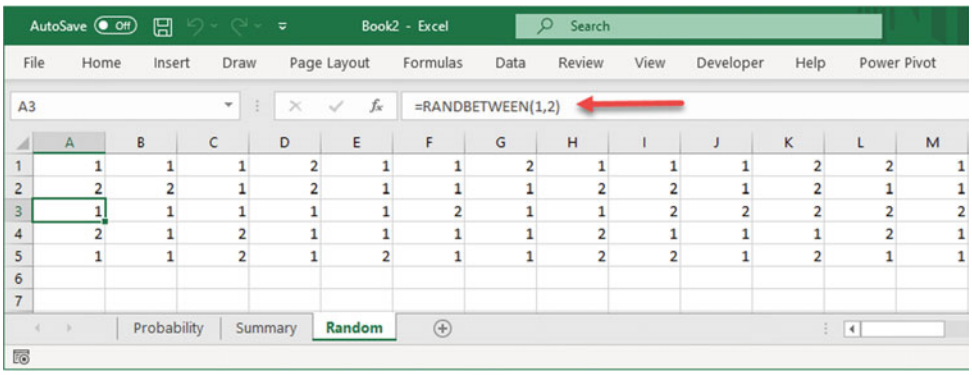
Notice that as we increased the number of times that we simulate the flipping of a coin, the closer a sample hovers around the theoretical value of 1.5 (1/2).

The following chart graphs the result of simulating the rolling of a die that has six sides. The die is rolled 1,000 times in 100 tries. Each side of the die has an equal chance of showing. Notice that the chart data hovers around the theoretical value of 1.6667 (1/6).



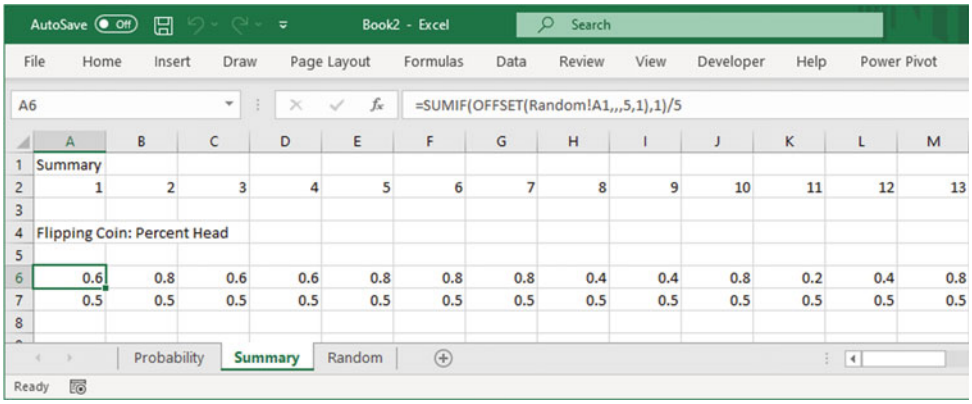
5.3.1.1 Probability Simulation Excel Workbook

The Excel probability macro creates a new workbook with three sheets. The first sheet is the *Random* sheet. Each column represents a sample. The default value in the simulator dialog box is 100 samples; this will result in the *Random* worksheet having 100 columns of data. The default value in the simulator dialog box is five values in a sample; this will result in each column having five values.

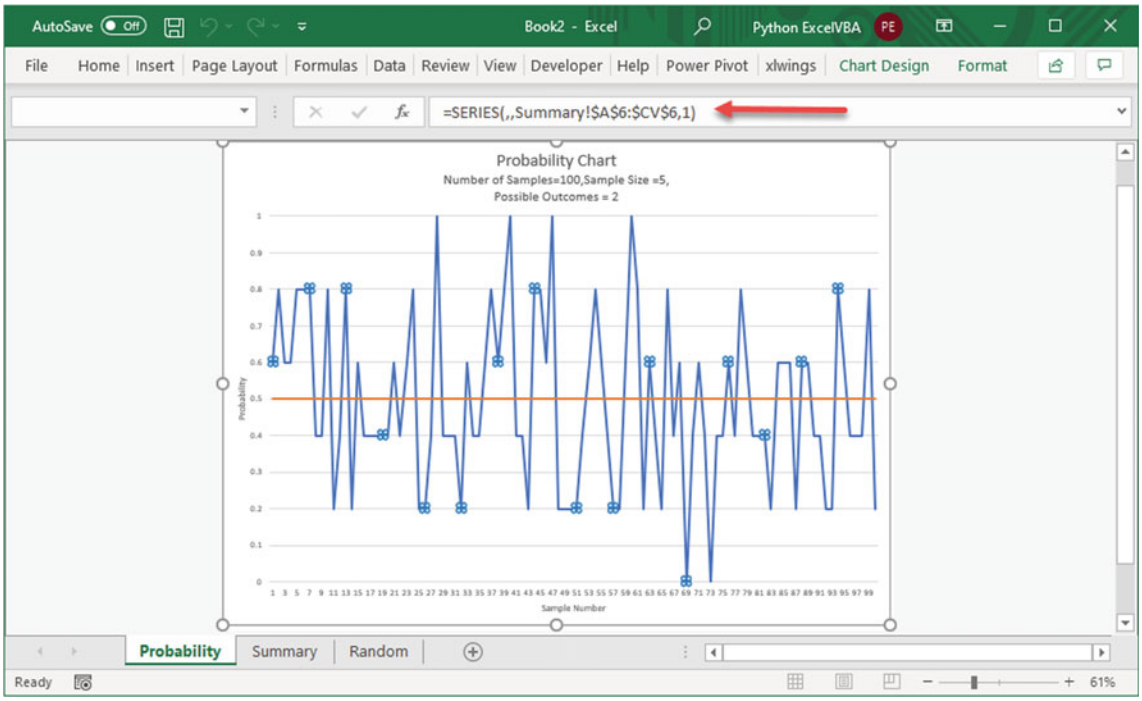


Each cell uses Excel's *RANDBETWEEN* function to generate random numbers.

The second sheet is the *Summary* sheet. Row 6 and row 7 are data rows for the probability chart. Cell a6 shows the percentage of cells in column A in the *Random* sheet that have the value of 1. Cell b7 shows the percentage of cells in column B in the *Random* sheets with the value of 1. The Excel *SUMIF* and Excel *OFFSET* functions are used to show the percentage of 1 in a column.



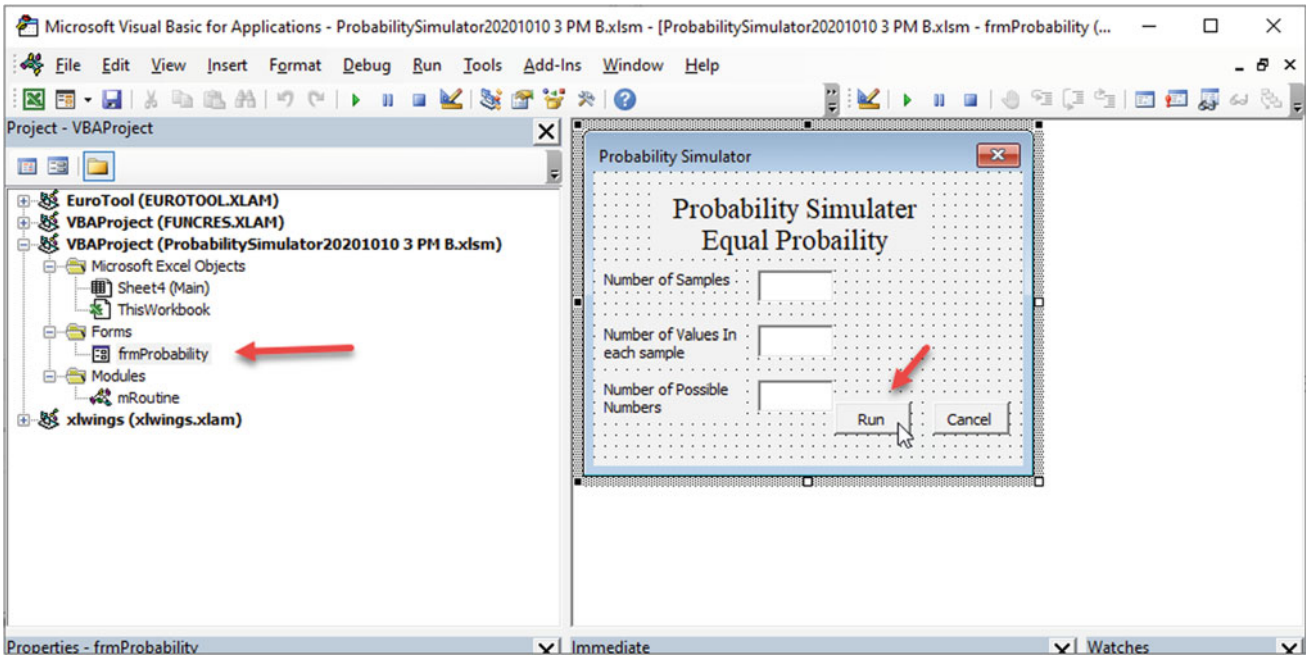
Row 7 shows the theoretical value that we expect.



The third sheet is the probability chart. Clicking on the blue line, we can see that this line's data source is row 6 of the "Summary" sheet. Clicking on the red line will show that the line's data source is row 7 of the "Summary" sheet.

5.3.1.2 Excel VBA Code

With the probability simulator workbook active, press the keys *Alt-F11*; this will show Excel's integrated development environment or IDE. The VBA code is part of the form *frmProbability*.



To see the code,

(1) Double-click on *frmProbability*.
(2) Double-click on the *Run* button.

The following is the VBA code for the probability simulator:

```
Option Explicit
'/*************************************************************************
'/            https://www.amazon.com/author/pythonexcelvba
'/*************************************************************************

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdRun_Click()
    If Me.txtSamples < 1 Or Me.txtSamples > 500 Then
        MsgBox "Please enter a number for the Samples between 1 and 256", _
         vbExclamation, "Probability Simulator"
         Exit Sub
    ElseIf Me.txtValues < 1 Or Me.txtValues > 65536 Then
        MsgBox "Please enter a number for the Values between 1 and 65536", _
         vbExclamation, "Probability Simulator"
         Exit Sub
    ElseIf Me.txtPossibleNum < 0 Then
        MsgBox "Please enter a number greater than 0 for the PossibleNum.", _
         vbExclamation, "Probability Simulator"
        Exit Sub
    End If
    Me.Hide
    ProbabilitySimulation
    Application.StatusBar = False
    Unload Me
End Sub
```

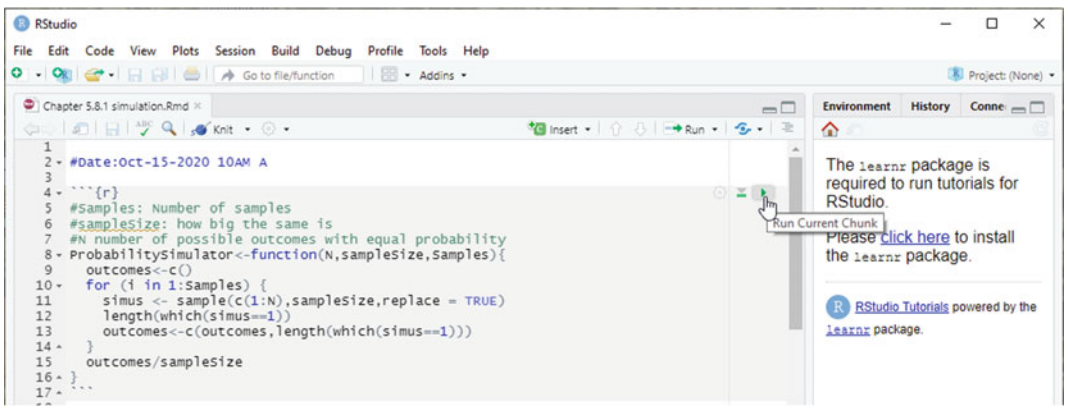
```
'/**********************************************************************
'/Purpose:This procedure randomlly chooses between 0 and 1 and then
'/         graphs the percentage of 1s in each sample
'/Date: Oct-10-2020
'/**********************************************************************
Sub ProbabilitySimulation()

    Dim lSamples As Long
    Dim lValues As Long
    Dim lBegin As Long
    Dim lEnd As Long
    Dim wbProbability As Workbook
    Dim wsRandom As Worksheet
    Dim wsSummary As Worksheet
    Dim rCell As Range
    Dim rRandom As Range
    Dim lRow As Long
    Dim lColumn As Long
    Dim vProb As Variant
    Dim dtStart As Date

    lSamples = Me.txtSamples
    lValues = Me.txtValues
    lBegin = 1
    lEnd = Me.txtPossibleNum

    Set wbProbability = Workbooks.Add
    Set wsRandom = wbProbability.Worksheets.Add
    Set wsSummary = wbProbability.Worksheets.Add

    wsRandom.Name = "Random"
    wsSummary.Name = "Summary"
    Application.Calculation = xlCalculationManual
    With wsRandom
        .Name = "Random"
        Set rRandom = .Range("a1").Resize(lValues, lSamples)
    End With

    dtStart = VBA.Now()
    ReDim vProb(1 To lValues, 1 To lSamples)

    For lColumn = 1 To lSamples

        wsRandom.Cells(1, 1).Offset(0, lColumn - 1).Resize(lValues, 1).FormulaR1C1 = _
                "=RANDBETWEEN(" & lBegin & "," & lEnd & ")"
    Next

    'change formulas to values
    wsRandom.Cells(1, 1).CurrentRegion.Value = wsRandom.Cells(1, 1).CurrentRegion.Value

    With wsSummary
            .Range("a1").Value = "Summary"
            .Range("a2").Value = 1
            .Range("B2").Value = 2
            .Range("a2:b2").AutoFill Destination:=.Range("a2").Resize(1, lSamples), Type:=xlFillDefault
            .Range("A4").Value = "Flipping Coin: Percent Head"
    End With

   ' Set rRandom = wbProbability.Worksheets("Random").Range("a1").CurrentRegion
    With wsSummary.Range("a6").Resize(1, lSamples).Cells
            .FormulaR1C1 = "=SUMIF(OFFSET(Random!R[-5]C,,," & lValues & ",1),1)/" & lValues
            .Offset(1, 0).Value = (1 / Me.txtPossibleNum)
    End With

    'Charting with Shape object instead of  Chart object
    Dim chrt As Shape
```

```
    Set chrt = wsSummary.Shapes.AddChart2(227, xlLine)
    With chrt.Chart
        .Axes(xlValue).MaximumScale = 1
        .SetSourceData Source:=wsSummary.Range("c6").CurrentRegion
        .SetElement (msoElementPrimaryCategoryAxisTitleAdjacentToAxis)
        .Axes(xlCategory, xlPrimary).AxisTitle.Text = "Sample Number"
        .SetElement (msoElementPrimaryValueAxisTitleAdjacentToAxis)
        .Axes(xlValue).HasTitle = True
        .Axes(xlValue).AxisTitle.Text = "Probability"
        .ChartTitle.Text = "Probability Chart" & Chr(10) _
                                & "Number of Samples=" & lSamples & "," _
                                & "Sample Size =" & lValues & "," & Chr(10) _
                                & "Possible Outcomes = " & Me.txtPossibleNum
        .ChartTitle.Characters(Start:=1, Length:=17).Font.Size = 18
        .Legend.Delete
        .Axes(xlCategory).TickLabelSpacing = lSamples / 50
        .Location Where:=xlLocationAsNewSheet, Name:="Probability"

    End With


    On Error Resume Next
    Application.DisplayAlerts = False
    wbProbability.Worksheets("Sheet1").Delete
    Application.DisplayAlerts = True
    Application.Calculation = xlCalculationAutomatic

End Sub

Private Sub UserForm_Initialize()
    'Default dialog box values
    With Me
        .txtSamples = 100
        .txtValues = 5
        .txtPossibleNum = 2
    End With
End Sub
```

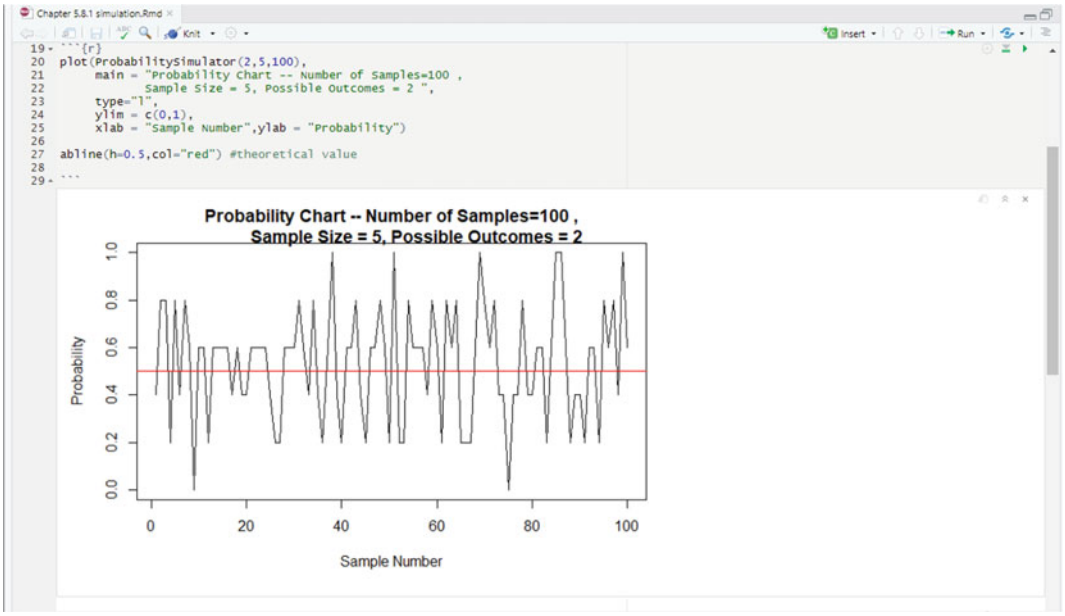
5.3.2 Probability Simulation in R

In this section, we will do the probability simulations in the R statistical package.

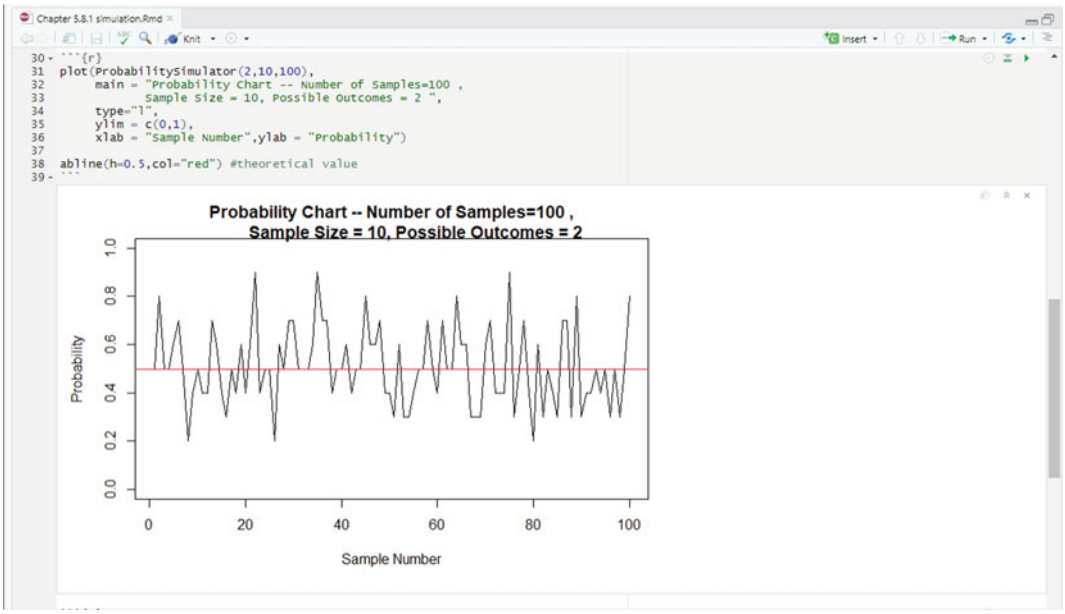
The *Probabilitysimulator* function is a custom function that will generate the probability simulation. This function has three parameters. The first parameter, *N*, tells the function how many possible outcomes are possible in the probability simulation. All outcomes have the same probability of occurring. The parameter, *Samples*, indicates how many samples to generate. The parameter, *samplesize*, indicates how many values are in the sample size.



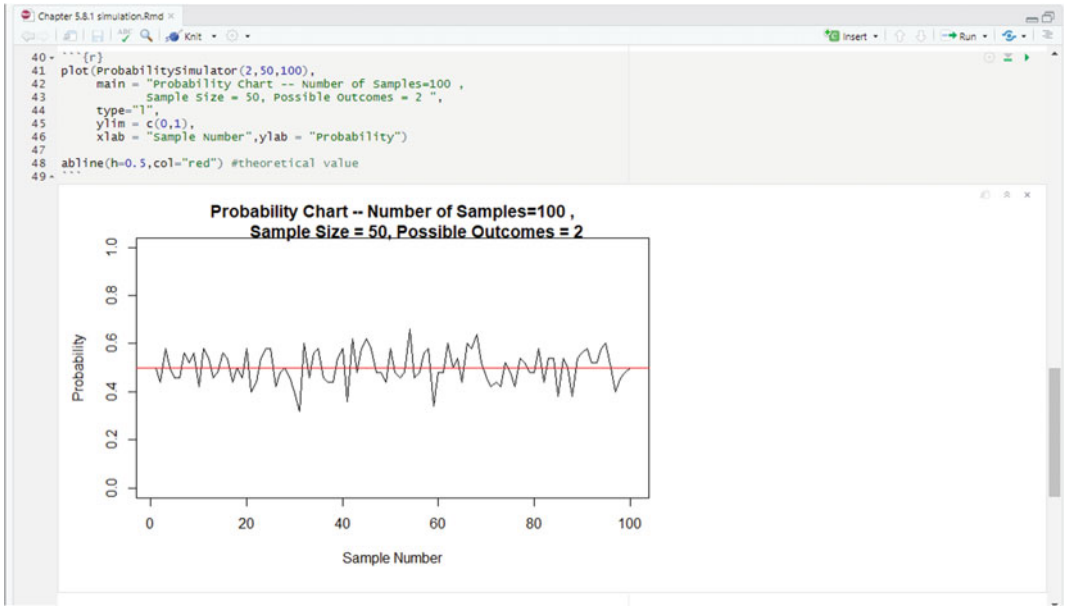
The following chart graphs the result of simulating the flipping of a coin five times in 100 tries.



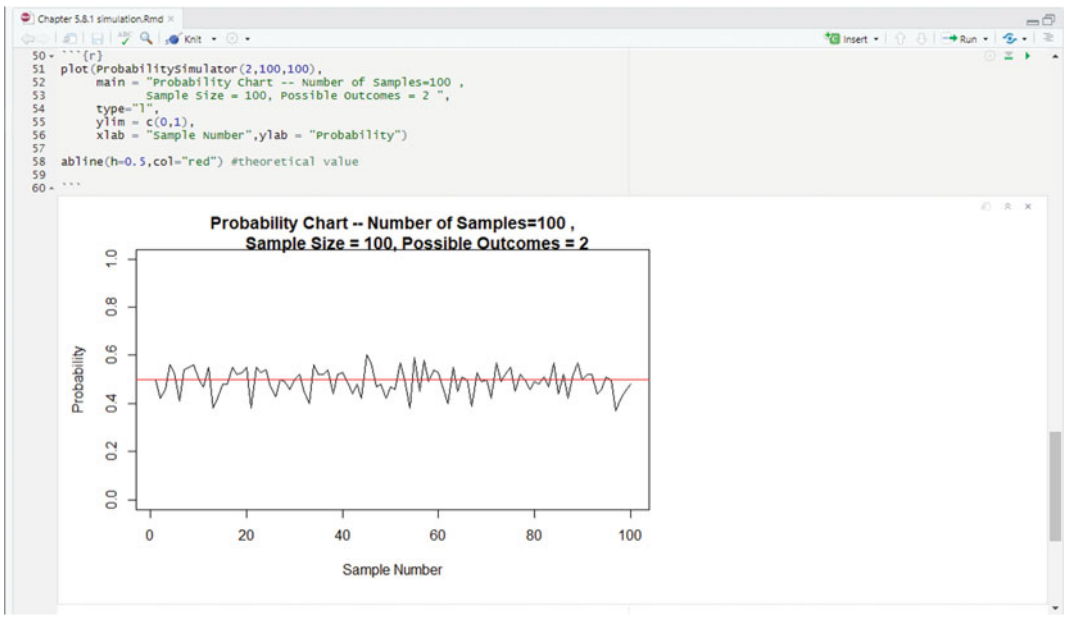
The following chart graphs the result of simulating the flipping of a coin ten times in 100 tries.



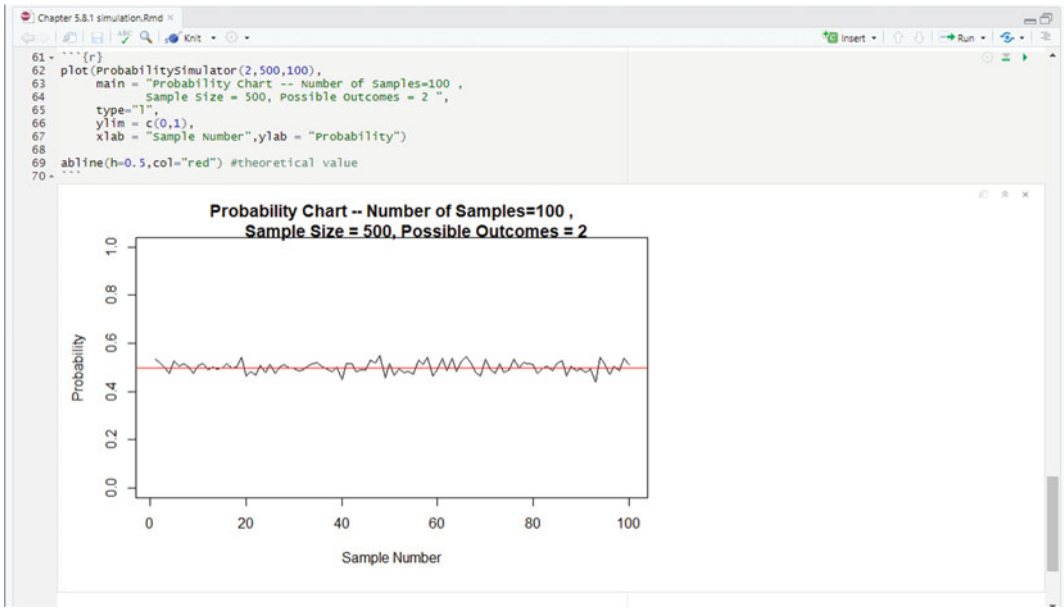
The following chart graphs the result of simulating the flipping of a coin 50 times in 100 tries.



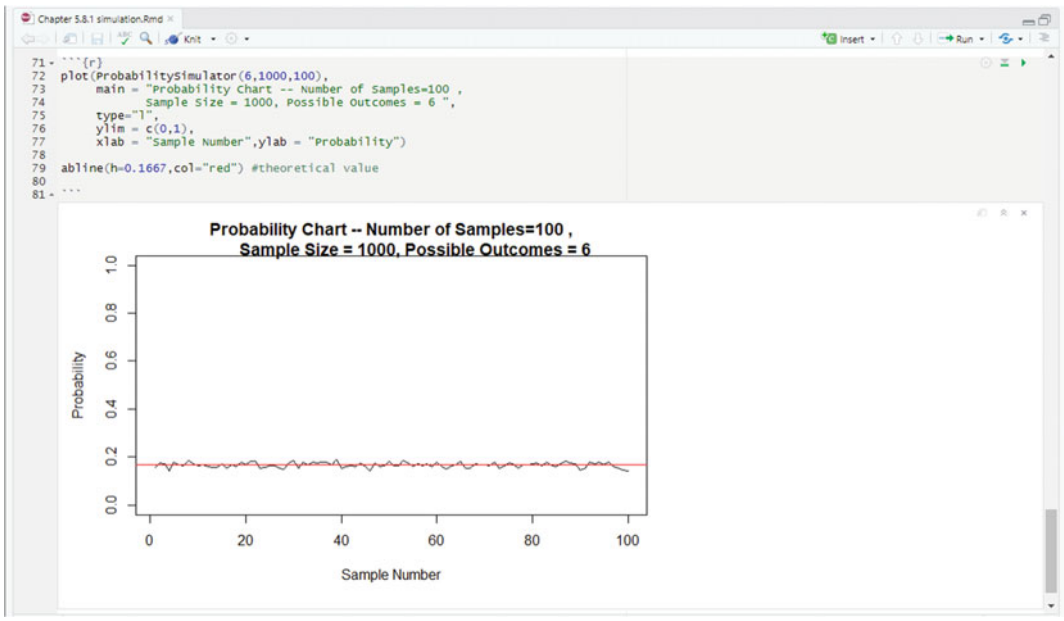
The following chart graphs the result of simulating the flipping of a coin 100 times in 100 tries.



The following chart graphs the result of simulating the flipping of a coin 500 times in 100 tries.



The following chart graphs the result of simulating the rolling of a die 1000 times in 100 tries.



5.3.3 Probability Simulation in Python

5.3.3.1 Code

Chapter5_Probability_Simulation.py

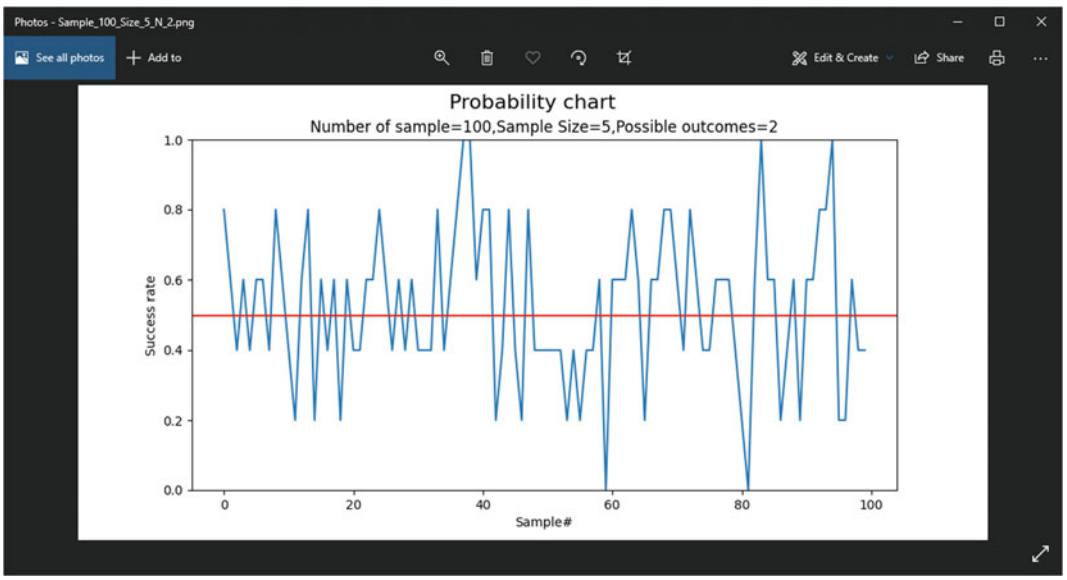
```
#Oct-16-2020 9AM A
import numpy as np # library to generate random numbers
import os # library to find current directory
import matplotlib.pyplot as plt

#generate ranodm numbers
#N is possible outcomes
#size is the size of the sample
#number_of_sample is the number of samples to be generated
def PsudoRand(N,size,number_of_sample):
    psudo = list(range(int(N)))#generate N unidform random number from uniform distribution
    choice = np.random.choice(psudo, (int(size),int(number_of_sample)))#choose N number from Rsudo without replacement
    return choice

#Crate Probability charts
#N is possible outcomes
#size is the size of the sample
#number_of_sample is the number of samples to be generated
def Plot(N,size,number_of_sample):
    Sample = PsudoRand(N,size,number_of_sample)
    num_of_success = np.count_nonzero(Sample==0,axis=0)/int(size)
    plt.figure(figsize=(10,5))
    plt.plot(num_of_success)
    plt.axhline(y=1.0/N, color='r', linestyle='-')
    plt.title('Number of sample='+str(number_of_sample)+',Sample Size='+str(size)+',Possible outcomes='+str(N),fontsize=12)
    plt.suptitle('Probability chart',fontsize=16)
    plt.ylim(0,1)
    plt.xlabel('Sample#')
    plt.ylabel('Success rate')

    filename = '\\Sample_' + str(number_of_sample) + '_Size_' + str(size) + '_N_' + str(N) + '.png'
    plt.savefig(os.getcwd() + filename) # save graph in current working directory

##################################### Main
print("Creating graph for 2 equal outcomes, size 5, 100 samples in the folder " + os.getcwd())
Plot(2,5,100)
print("Creating graph for 2 equal outcomes, size 10, 100 samples in the folder " + os.getcwd())
Plot(2,10,100)
print("Creating graph for 2 equal outcomes, size 50, 100 samples in the folder " + os.getcwd())
Plot(2,50,100)
print("Creating graph for 2 equal outcomes, size 100, 100 samples in the folder " + os.getcwd())
Plot(2,100,100)
print("Creating graph for 2 equal outcomes, size 500, 100 samples in the folder " + os.getcwd())
Plot(2,500,100)
print("Creating graph for 3 equal outcomes, size 1000, 100 samples in the folder " + os.getcwd())
Plot(3,1000,100)
```

5.3.3.2 Output

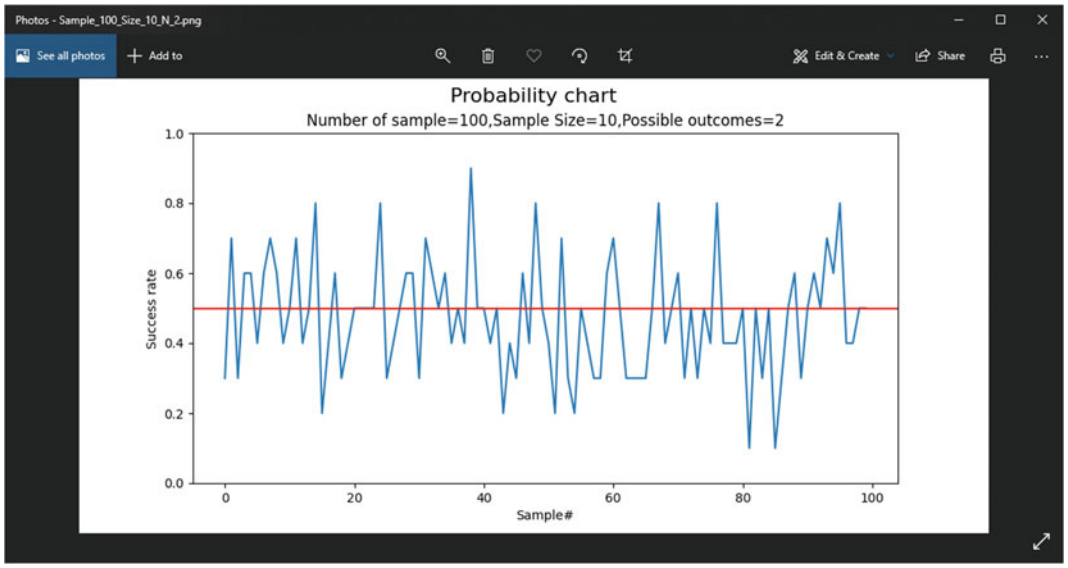
Start reading the Python code on line 35. The probability simulation graphs are saved in the same folder as the *Python* code. Line 32 saves the probability simulation graphs. Below is the output of the *Python* code.

```
Chapter5_Probability_Simulation.py ×
  1  #Oct-16-2020 9AM A
  2  import numpy as np # library to generate random numbers

OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS                                  1:

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Demo\Chpater5> & 'C:\Python\python.exe' 'c:\Users\johnl\.vscode\extensions\ms-python.python-2020.9.114305\
406' '--' 'c:\Demo\Chpater5\Chapter5_Probability_Simulation.py'
Creating graph for  2 equal outcomes, size 5, 100 samples in the folder C:\Demo\Chpater5
Creating graph for  2 equal outcomes, size 10, 100 samples in the folder C:\Demo\Chpater5
Creating graph for  2 equal outcomes, size 50, 100 samples in the folder C:\Demo\Chpater5
Creating graph for  2 equal outcomes, size 100, 100 samples in the folder C:\Demo\Chpater5
Creating graph for  2 equal outcomes, size 500, 100 samples in the folder C:\Demo\Chpater5
Creating graph for  3 equal outcomes, size 1000, 100 samples in the folder C:\Demo\Chpater5
PS C:\Demo\Chpater5> dir


    Directory: C:\Demo\Chpater5


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/16/2020   9:34 AM           2100 Chapter5_Probability_Simulation.py
-a----        10/16/2020   9:58 AM          36373 Sample_100_Size_1000_N_3.png
-a----        10/16/2020   9:58 AM          46640 Sample_100_Size_100_N_2.png
-a----        10/16/2020   9:58 AM          65603 Sample_100_Size_10_N_2.png
-a----        10/16/2020   9:58 AM          41244 Sample_100_Size_500_N_2.png
-a----        10/16/2020   9:58 AM          52823 Sample_100_Size_50_N_2.png
-a----        10/16/2020   9:58 AM          68455 Sample_100_Size_5_N_2.png
```

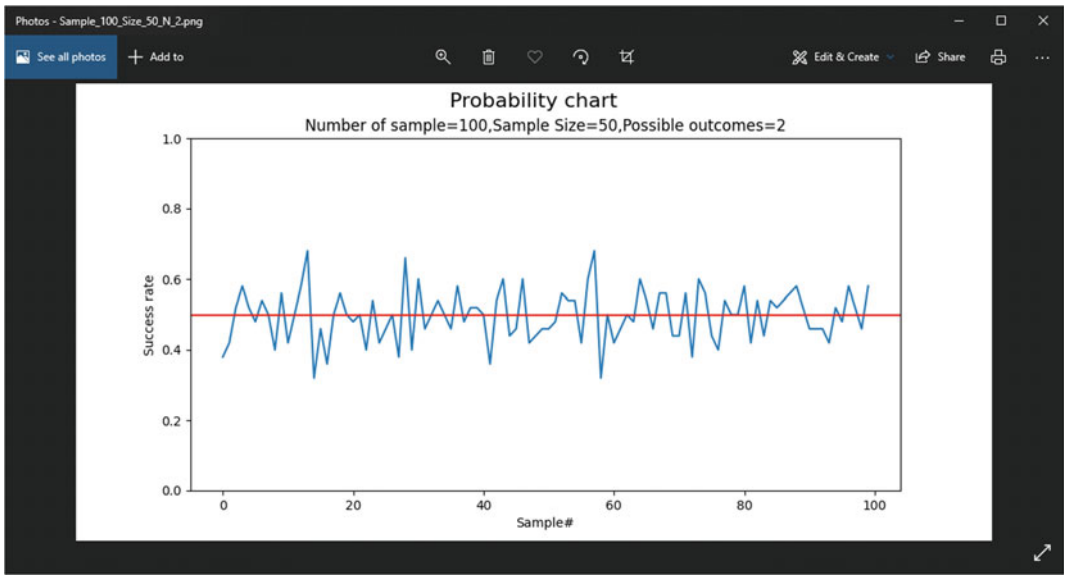
The following chart graphs the result of simulating the flipping of a coin five times in 100 tries.



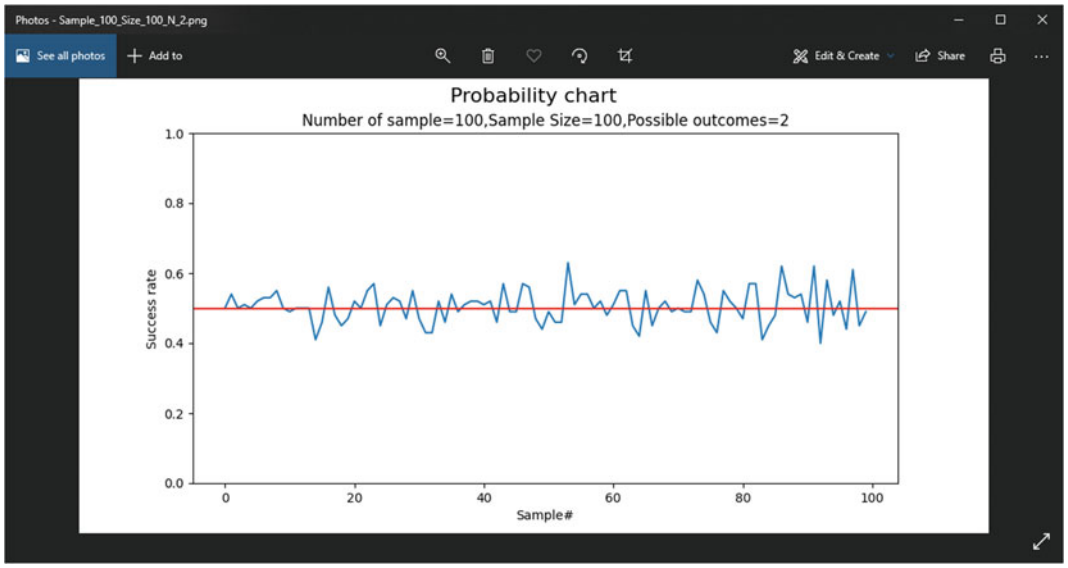
The following chart graphs the result of simulating the flipping of a coin ten times in 100 tries.



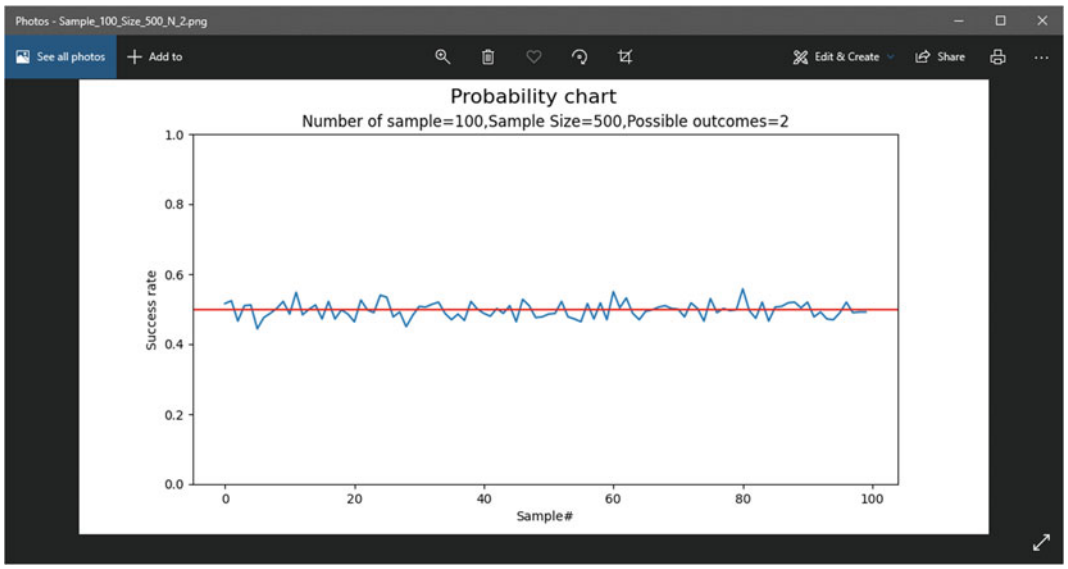
The following chart graphs the result of simulating the flipping of a coin 50 times in 100 tries.



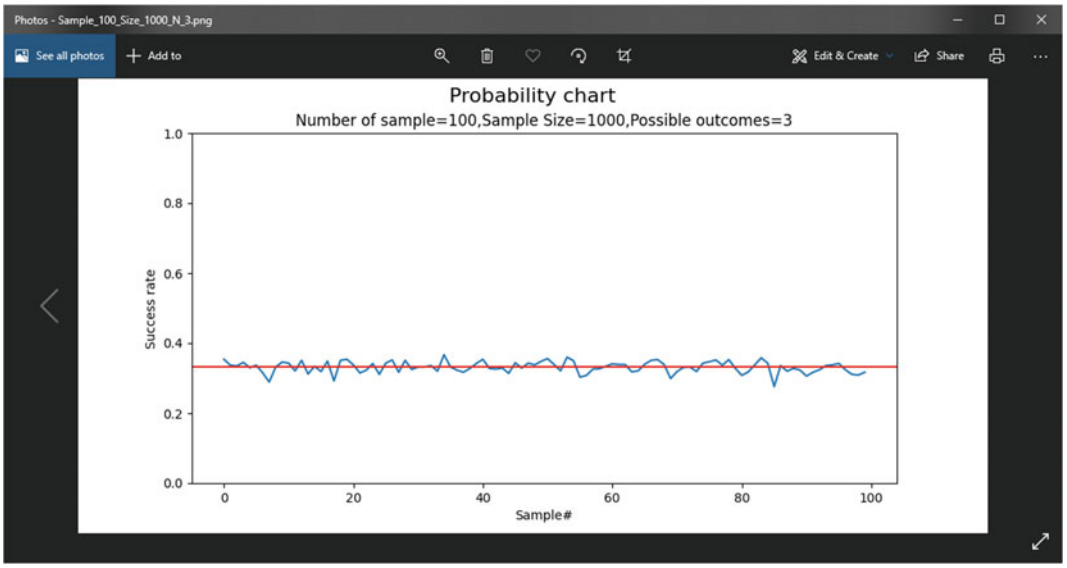
The following chart graphs the result of simulating the flipping of a coin 100 times in 100 tries.



The following chart graphs the result of simulating the flipping of a coin 500 times in 100 tries.



The following chart graphs the result of simulating the rolling of a die 1000 times in 100 tries.



5.4 Combinations

Many times when we are trying to figure out the total number of outcomes, we need to figure out the *combination* of a set of values. Suppose we are interested in the number of possible arrangements of the letters A, B, and C, and order is not important. That is, AB and BA are considered the same arrangements.

The mathematical formula for a combination is as follows:

$$n^C r = \frac{n!}{r!(n-r)!}$$

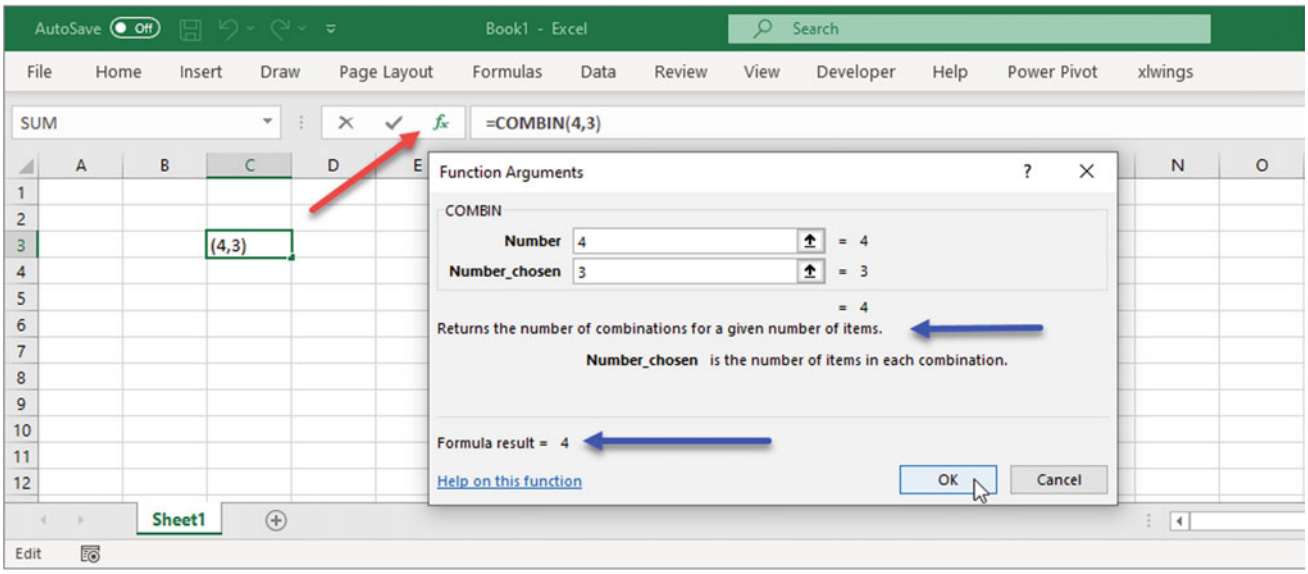
where:

n = total number of elements in the list.

r = the number of items chosen for the combination.

Suppose we are interested in finding out how many combinations of three names are from a list of four names.

Excel has a built-in function called *COMBIN* to calculate the number of combinations from a list of values; this is shown below.

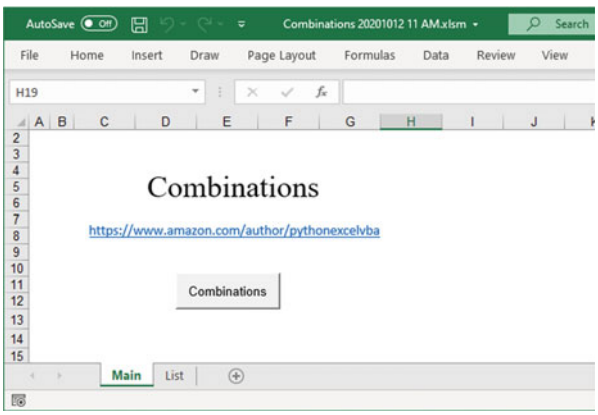


Clicking the *Insert Function* icon brings up the *Function Arguments* dialog box. This dialog box shows that the *COMBIN* function has two arguments, *Number* and *Number_chosen*. The *Function Arguments* dialog box indicates the *COMBIN* function "Returns the number of combinations for a given number of items."

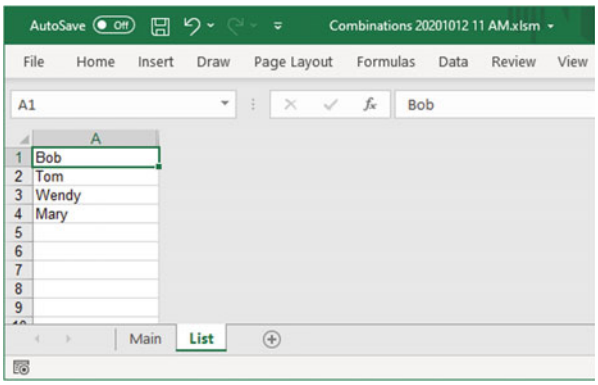
Excel calculates that there are four possible combinations of three names from a list of four names. Many times, we are interested in the list of combinations. In the next section, we use an Excel macro to list the combination of interests.

5.4.1 Combination List with Excel VBA

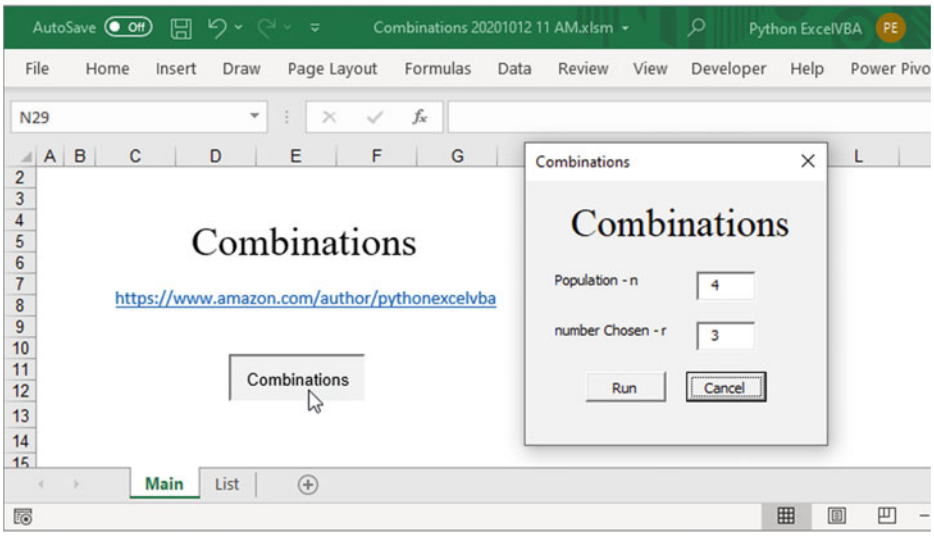
In this section, we will look at an Excel VBA macro to find all the combinations of three names from a list of four names.



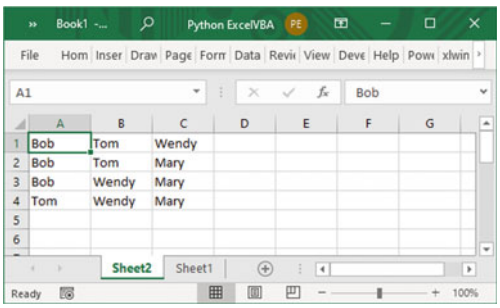
The worksheet *List* contains a list of four names.



To run the program, we need to push the *Combinations* button on the worksheet *Main*.

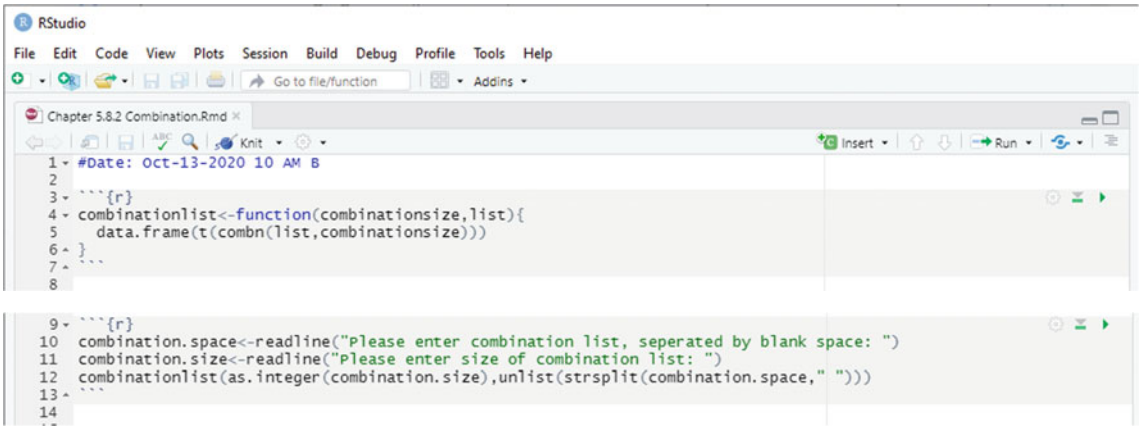


Pressing the *Run* button results in four combination rows from a combination of three names from a list of four names.

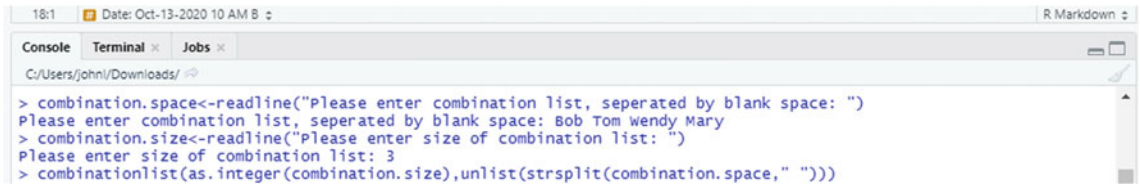


5.4.2 Combination List with R

The *R* function generates a combination list based on the parameters *combinationsize* and *list*.



The *R* code in lines 10 and 11 generates the following in the *Console*.



The *R* code in line 12 generates the following output, which agrees with the Excel output.

5.5 Permutations

Many times, when we are trying to figure out the total number of outcomes, we need to figure out the *permutation* of a set of values. Suppose we are interested in the number of possible arrangements of the letters A, B, and C, and order is important. That is, AB and BA are two different arrangements. The concept of permutations can help us answer our question.

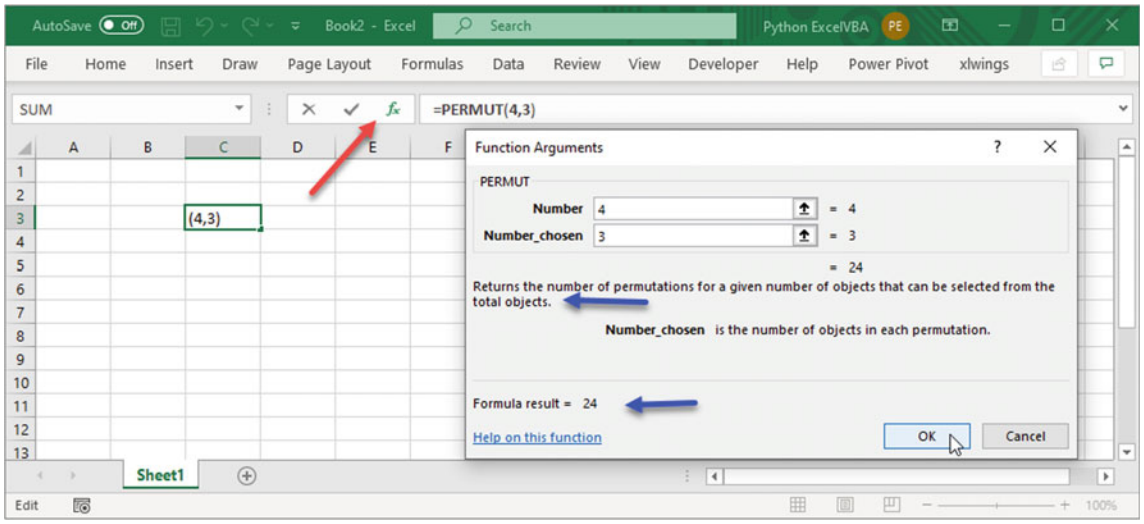
The mathematical formula for a combination is as follows:

where:

n = total number of elements in the list.

r = the number of items chosen for the combination.

Excel has a built-in function called *PERMUT* to calculate the number of permutations from a list of values; this is shown below.

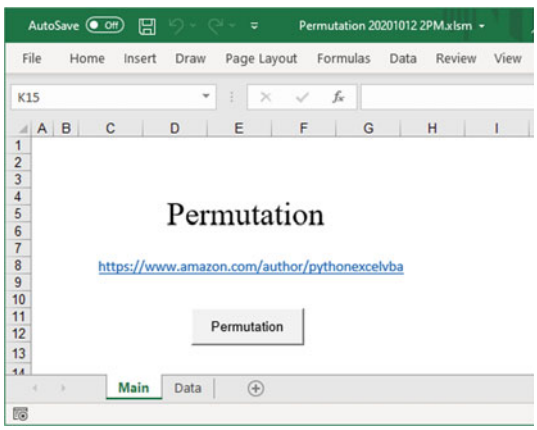


Clicking the *Insert Function* icon brings up the *Function Arguments* dialog box. This dialog box shows that the *PERMT* function has two arguments, *Number* and *Number_chosen*. The *Function Arguments* dialog box indicates the *PERMUT* function "Returns the number of permutations for a give number of objects that can be selected from the total objects."

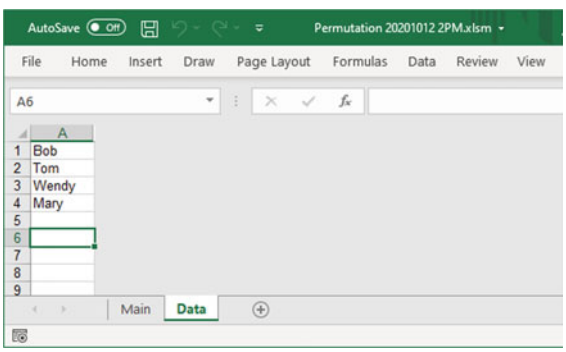
Excel calculates that there are twenty-four possible permutations from three names from a list of four names. Many times, we are interested in the list of permutations. In the next section, we use an Excel macro to list all the permutations.

5.5.1 Permutation List with Excel VBA

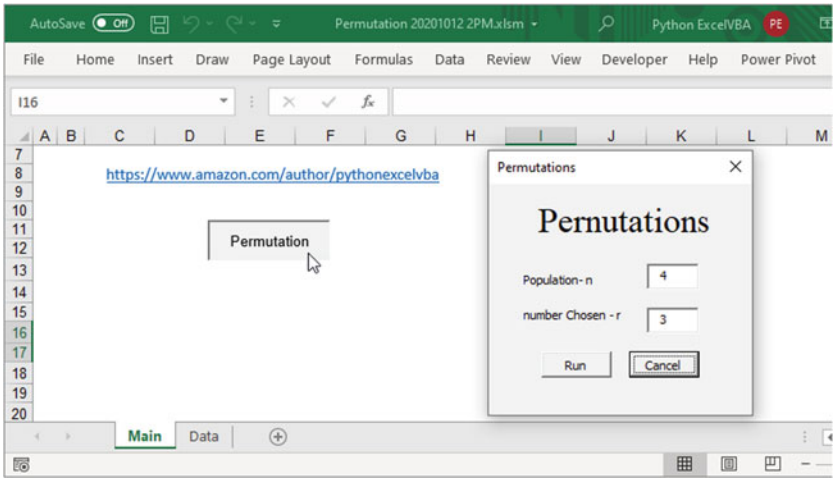
In this section, we will look at an Excel VBA macro to find all the permutations of three names from a list of four names.



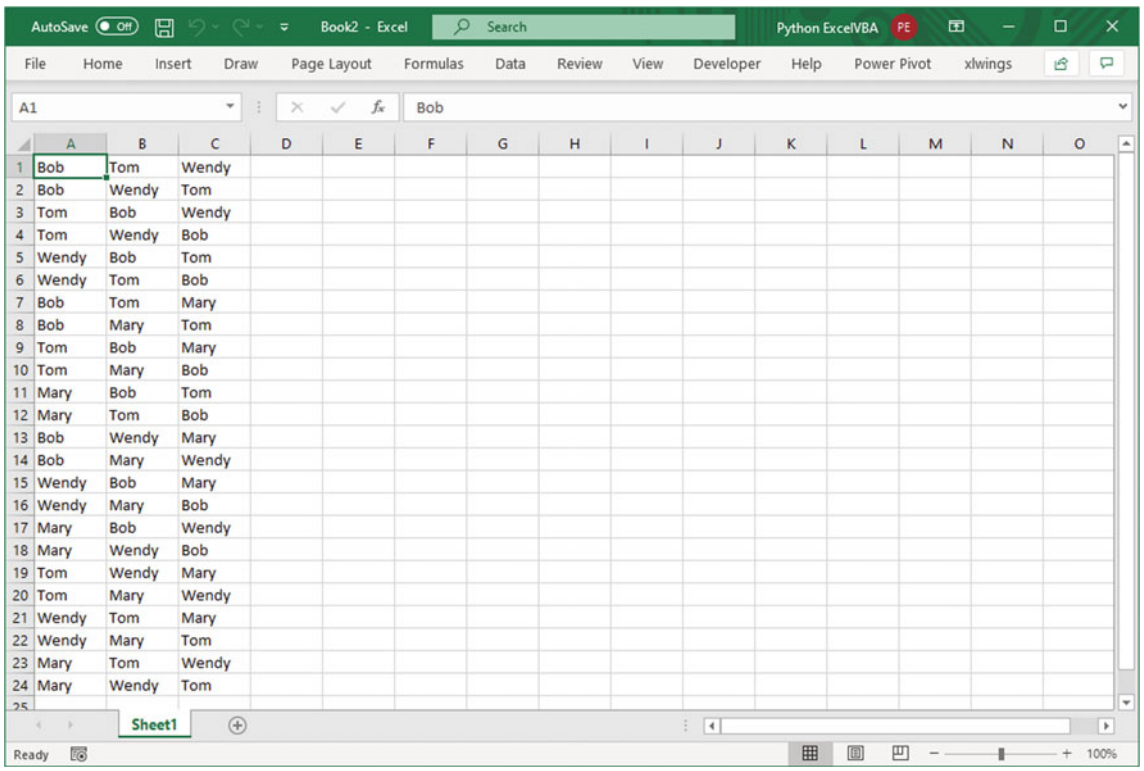
The worksheet *Data* contains a list of four names.



To run the program, we need to push the *Permutation* button on the worksheet *Main*.



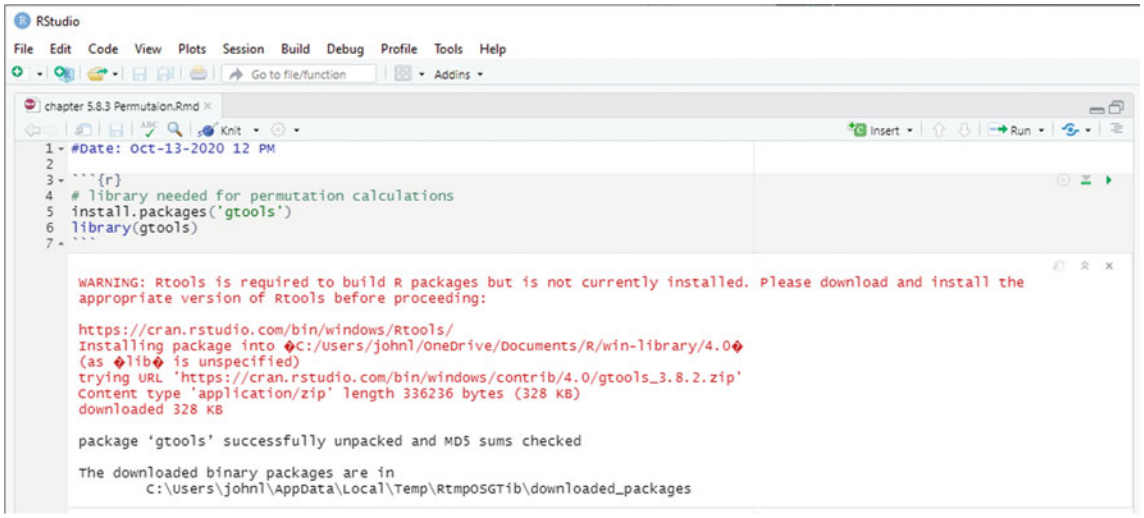
Below is shown all the possible permutations of three names from a four-name list after pressing the *Permutation* button on the *Main* worksheet.

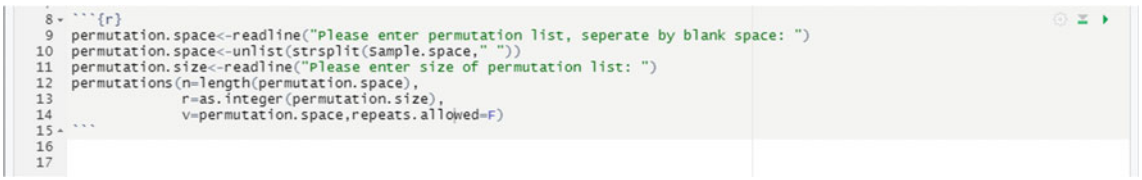


	A	B	C
1	Bob	Tom	Wendy
2	Bob	Wendy	Tom
3	Tom	Bob	Wendy
4	Tom	Wendy	Bob
5	Wendy	Bob	Tom
6	Wendy	Tom	Bob
7	Bob	Tom	Mary
8	Bob	Mary	Tom
9	Tom	Bob	Mary
10	Tom	Mary	Bob
11	Mary	Bob	Tom
12	Mary	Tom	Bob
13	Bob	Wendy	Mary
14	Bob	Mary	Wendy
15	Wendy	Bob	Mary
16	Wendy	Mary	Bob
17	Mary	Bob	Wendy
18	Mary	Wendy	Bob
19	Tom	Wendy	Mary
20	Tom	Mary	Wendy
21	Wendy	Tom	Mary
22	Wendy	Mary	Tom
23	Mary	Tom	Wendy
24	Mary	Wendy	Tom

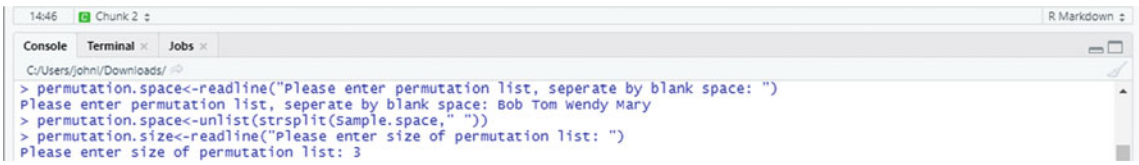
5.5.2 Permutation List with R

The *R* library *gtools* is required to create the permutation list.

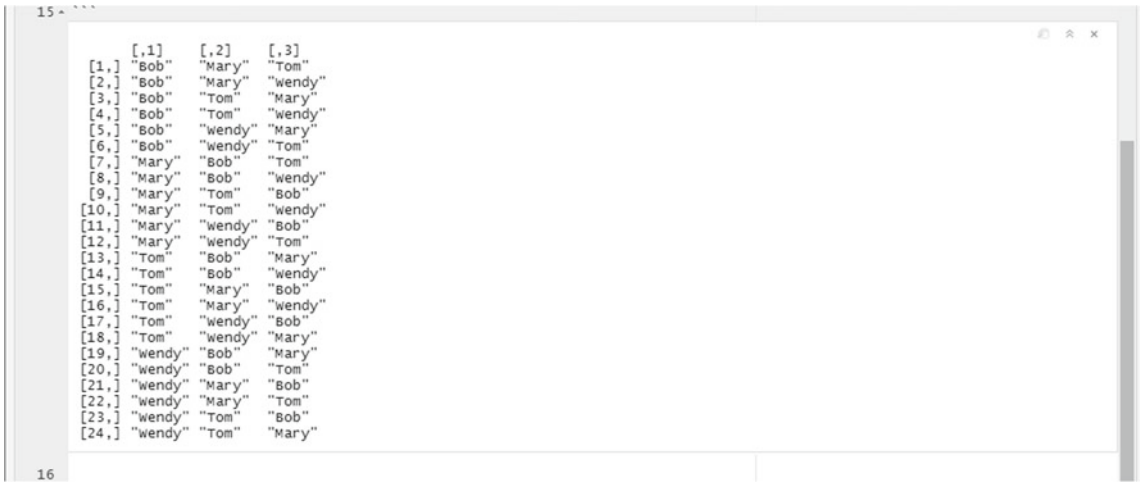




Lines 9, 10, and 11 create the following output in the console.



Line 12 generates the following permutation list, which agrees with the Excel output.



5.6 Summary

This chapter begins our study of inferential statistics. We first look at the very basic concepts of probability because inferential statistics need to use probability to make inferences about a population from sample data.

We are studying inferential statistics to do the following:

$$\text{Sample Data} \rightarrow \text{Population Data}$$

We study probability to do the following:

$$\text{Population Data} \rightarrow \text{Sample Data}$$

In this chapter, we did a probability simulation in Excel, Python, and R to illustrate the concept of probability. We simulated the chance of getting flipping a coin and getting a "head." Logically, we concluded that the probability should be 50%. We saw in the simulation that the more times we flipped in a trial, the closer the result was to 50%.

We also used Excel and R to list the combination and permutations of a list.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With VBA.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Discrete Random Variables and Probability Distributions

6

6.1 Introduction and Probability Distribution

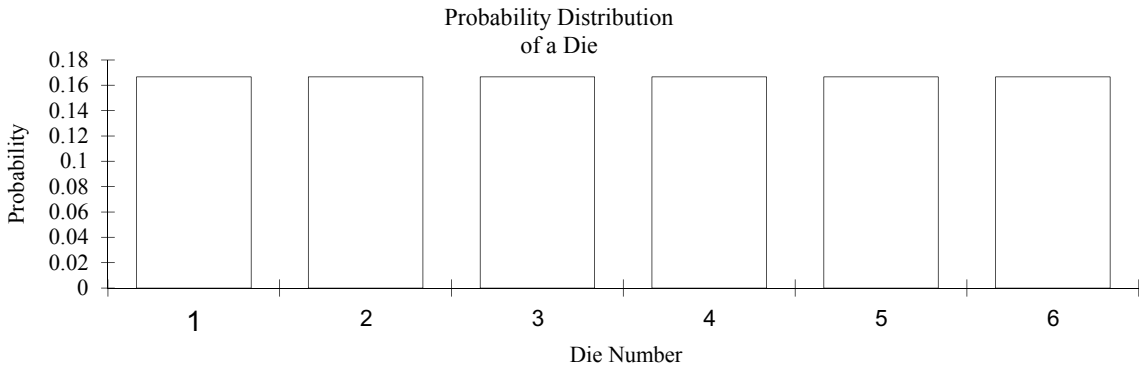
In this chapter, and the next three chapters, we will begin to study specific data sets of importance in statistics. With all of these data sets, we are interested in how the data items are distributed and the likelihood of getting a specific data item from a data set.

For example, let's show a distribution of a die. There are six possible numbers in a die: 1, 2, 3, 4, 5, and 6. We can think of these possible numbers as the population. If we roll a die, we have a 16.66% chance of getting the number 3. We also have a 16.66% chance of getting the numbers 1, 4, 5, or 6. Now let's list out the probability of getting each specific data item as shown below.

Probability Distribution of a Die	
#	probability
1	16.667%
2	16.667%
3	16.667%
4	16.667%
5	16.667%
6	16.667%
Total Prob	100% *

* number rounded

The list of all the probability distributions of each data item is called a *probability distribution*. Frequently, we graph the probability distribution of a data set. Below is the graph of the probability distribution of a data set.



© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_6

So far, we have been looking at data sets that have countable data items. In this chapter, we will look at countable data sets of interest in statistics. Statistics model each data set of interest with a mathematical formula. The mathematical formula for our die would be

$$F(x) = .1666, \text{ where } 1 \leq x < \leq 6$$

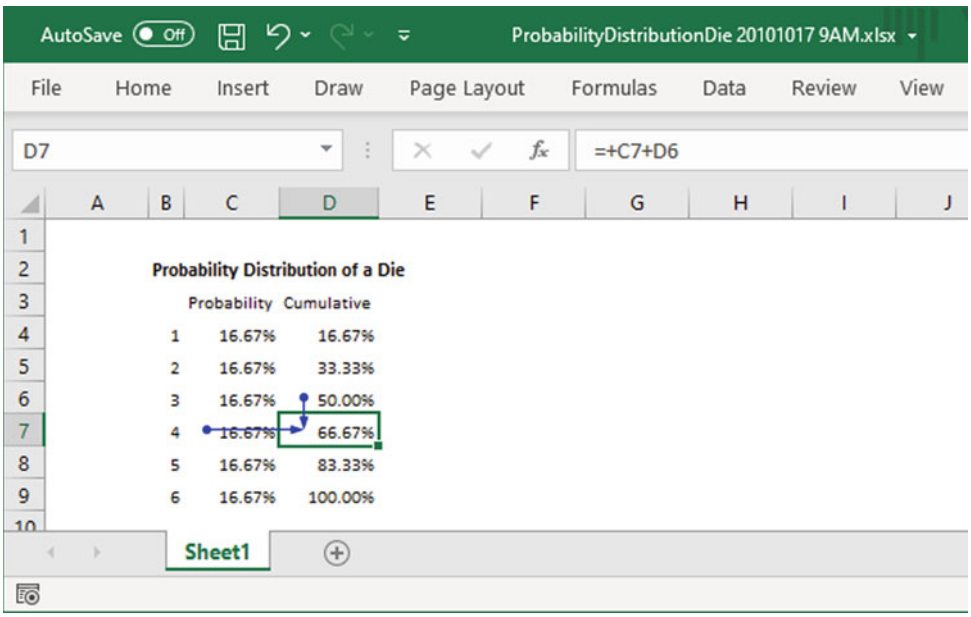
In this function, the x variable can be any value between 1 and 6 inclusive. Statistics call this type of x variable a *discrete random* variable.

The chapter is broken down as follows. In Sect. 2, we discuss cumulative probability distribution, and in Sect. 3, we talk about binomial distribution. Poisson random variable is examined in Sect. 4. Excel 4.0 macro functions and Excel names are discussed in Sect. 5. Stephen Bullen's charting method is explained in Sect. 6. Finally, in Sect. 7, we summarize the chapter.

6.2 Cumulative Probability Distribution

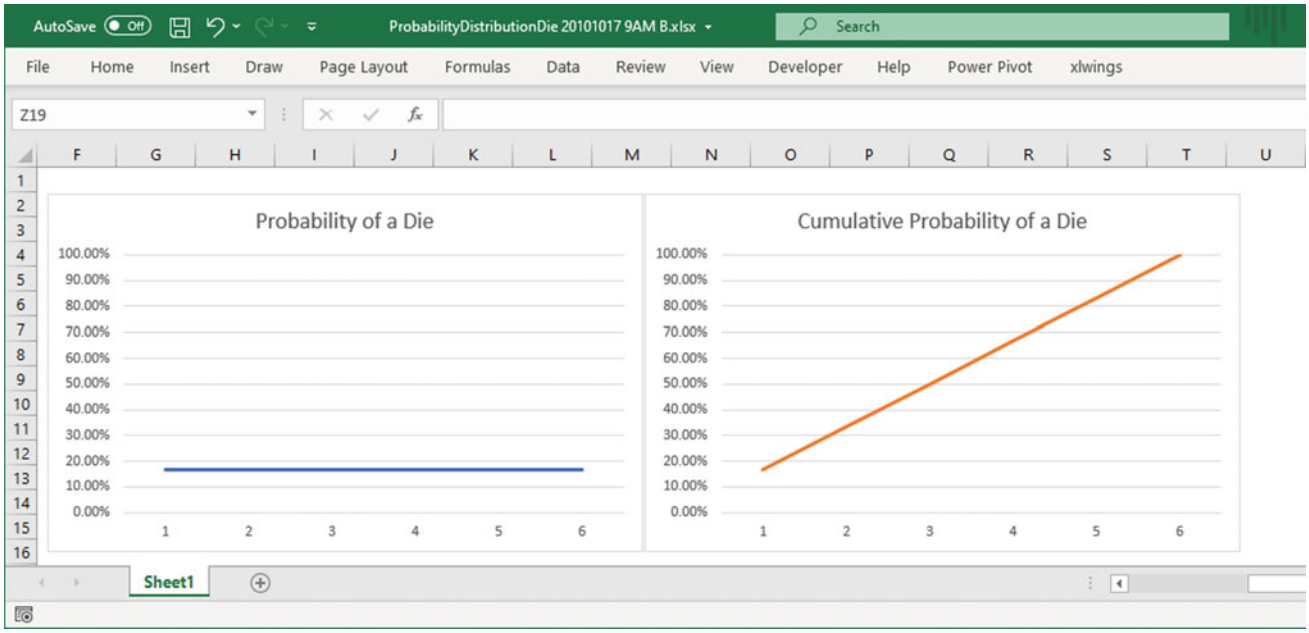
In the previous section, we were interested in getting the probability of getting a specific data item. In statistics, we are also interested in getting the probability of getting more than one specific data item; more specifically, in statistics, we are interested in cumulative probabilities. For our die, we would be interested in the following questions. What is the probability of getting either a 1 or 2? What is the probability of getting a 1 or 2 or 3? What is the probability of getting either a 1 or 2 or 3 or 4? What is the probability of getting a 1 or 2 or 3 or 4 or 5? What is the probability of getting a 1 or 2 or 3 or 4 or 5 or 6? The function that models these questions is

$$F(x) = .1666x, \text{ where } 1 \leq x \leq 6.$$



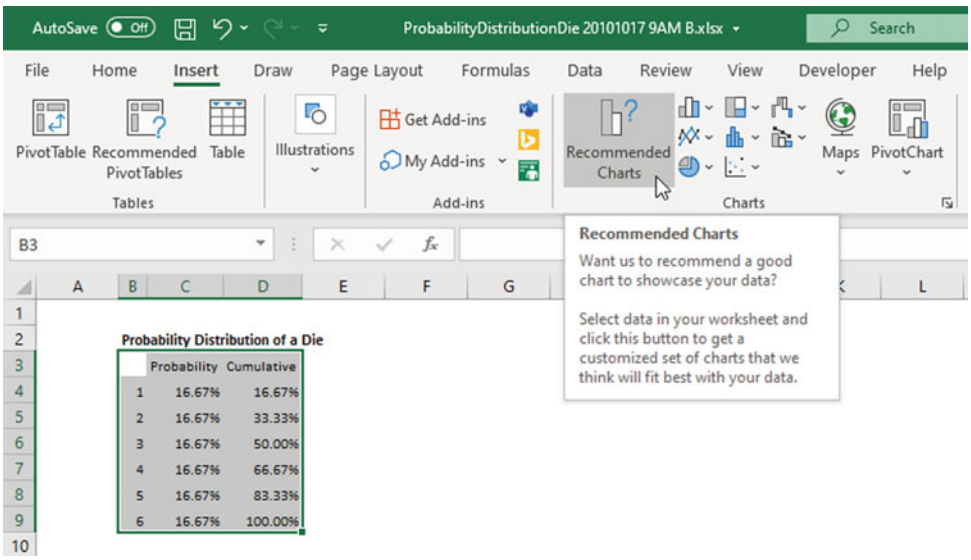
The above table indicates the probability of getting a 1 or 2 or 3 or 4 from rolling a die is 66.67%.

In statistics, we like to produce two graphs for every distribution. The first graph is the probability distribution of a graph. The second graph is the cumulative distribution of a graph.

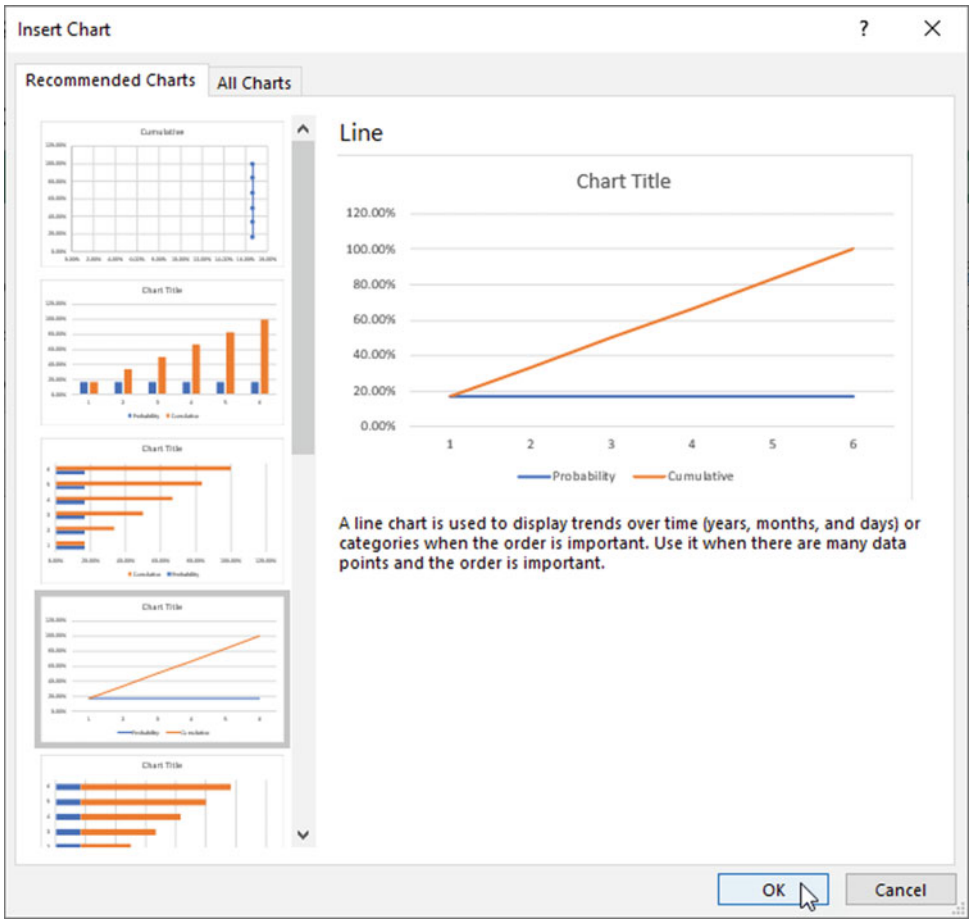


The following are steps to create the probability distribution graph of a die:

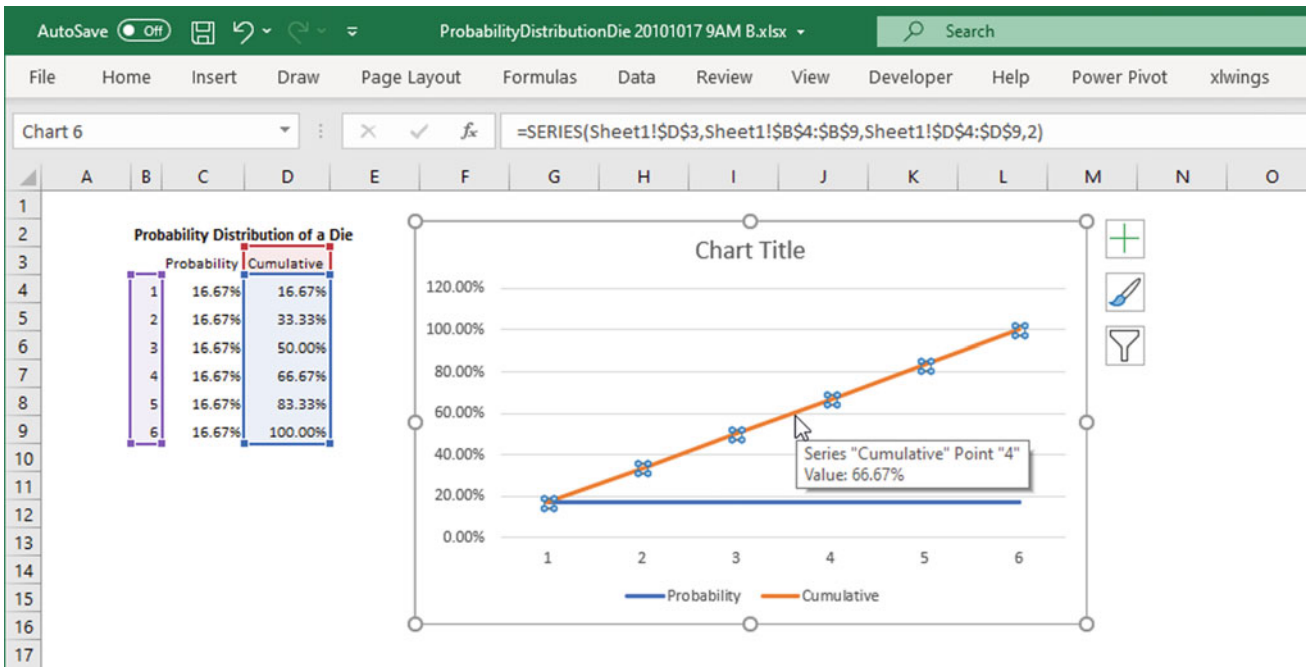
(1) Select cells b3:d9, then click on *Recommend Charts* on the *Insert* tab.



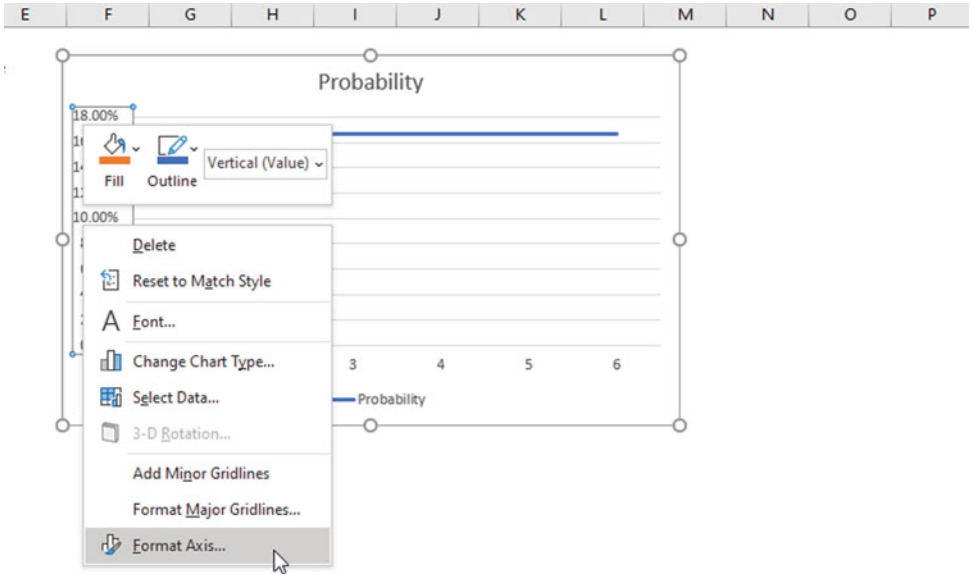
(2) Select *Line* chart.



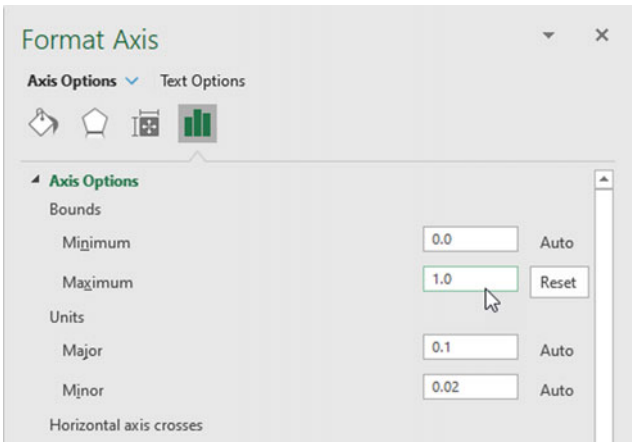
(3) Click on the *Cumulative* line and then press the *delete* key.



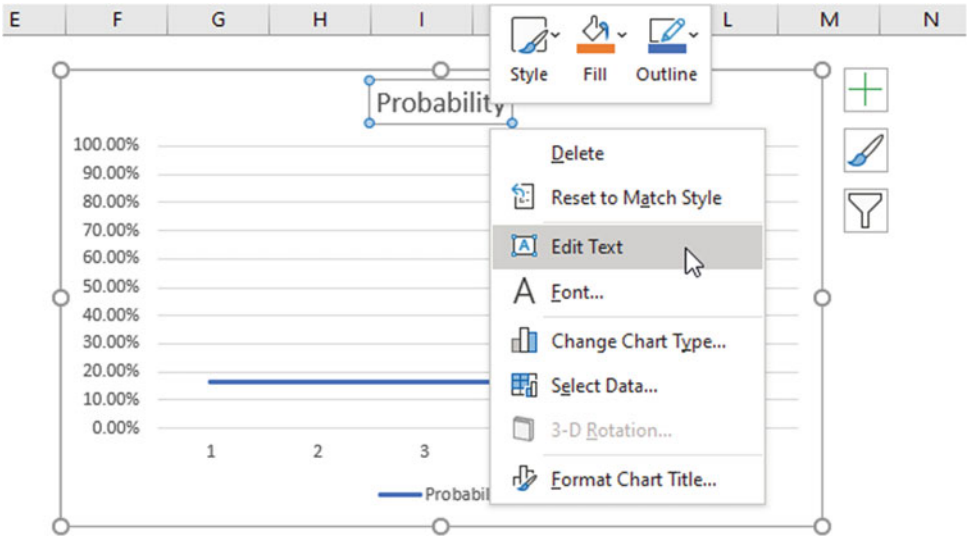
(4) Right-mouse click on the y-axis and choose *Format Axis.*



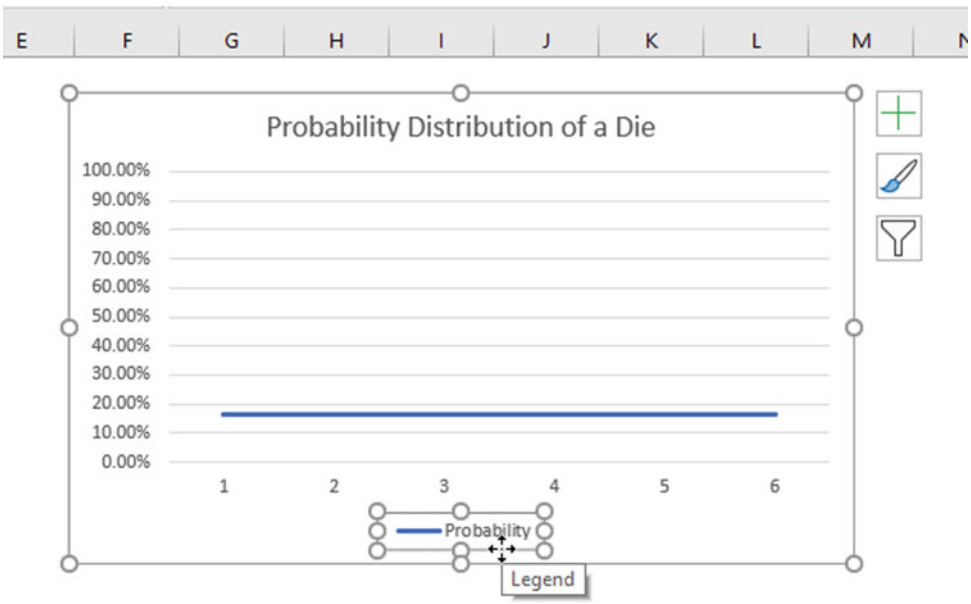
(5) Set the maximum bound to 1.



(6) Right-mouse click on chart and *Edit Text.*



(7) Click on legend and then press the *Delete* key.



The steps to create the cumulative probability distribution graph of a die are the same as creating the probability distribution graph of a die.

6.3 Binomial Distribution

The first important distribution that we will look at is the *binomial distribution.* There are two important characteristics of the binomial distribution. The first important characteristic is that the probability of something happening is independent of what happened previously. For example, if I flip a coin, there is a 50% chance that I will get a Head. If I flip it one more time, there is still a 50% chance that I will get a Head. Now suppose I randomly put in ten numbers into a hat. The first time I pull out a

number, each number has a 10% chance of being chosen. The second time I pull out a number, the probability of each of the remaining numbers being chosen is not 10% anymore. It is now 1/9 or 11%.

The second important characteristic of the binomial distribution is there are only two possible values possible. These values could represent True, False, or Success, Failure, or 0, 1.

The binomial distribution is mathematically defined as the following:

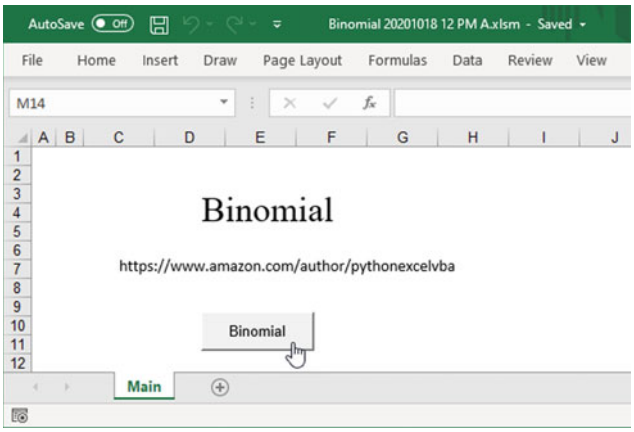
$$\frac{n!}{r!(n-r)!}p^r q^{n-r}$$

In the formula, n is how many times we flip the coin, r is how many times we want the outcome to be tails, p is the probability of success, and q is the probability of failure.

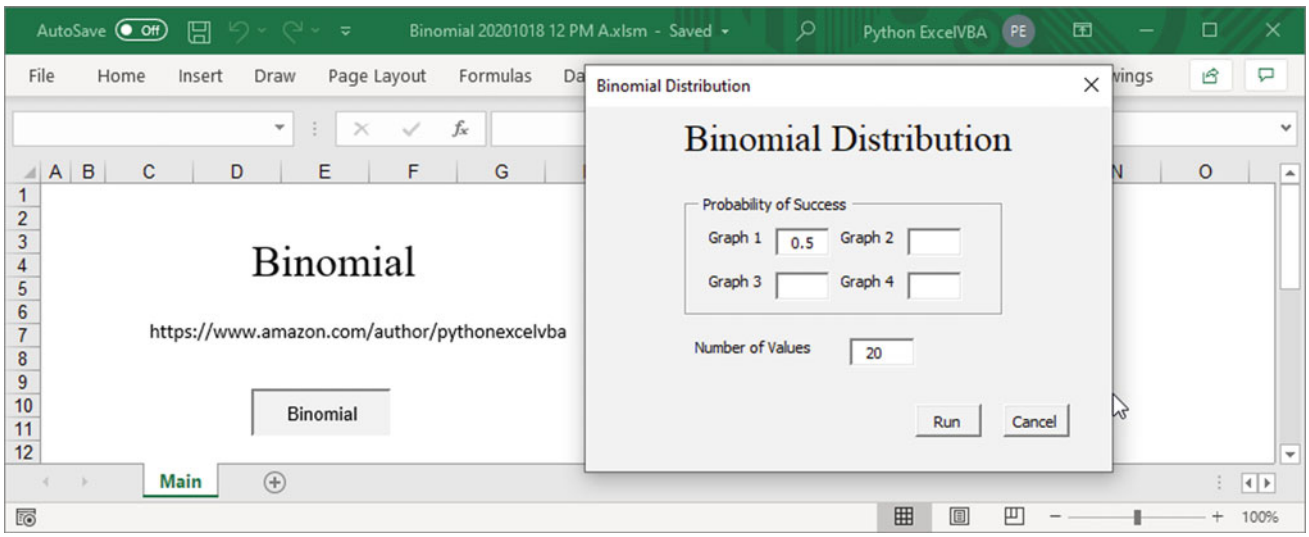
As an example, suppose that we are playing in a friendly betting game. The game consists of betting how many times we will get heads when we flip a coin ten times. If we think about this game carefully, it follows a binomial distribution. In this game, we will have to pay $1 to play it, and we will receive $2 if we guess how many times the coin will turn up heads. What number should we choose? Should we choose 0, 3, 5, 7, or another number? We can randomly choose one of the six numbers (0–5) or make an educated bet.

6.3.1 Binomial Distribution in Excel

To make an educated bet, let's create a binomial distribution table for the ten-coin flips and also graph the binomial distribution for ten-coin flips. At the same time, we will also create a table and graph the cumulative binomial distribution.

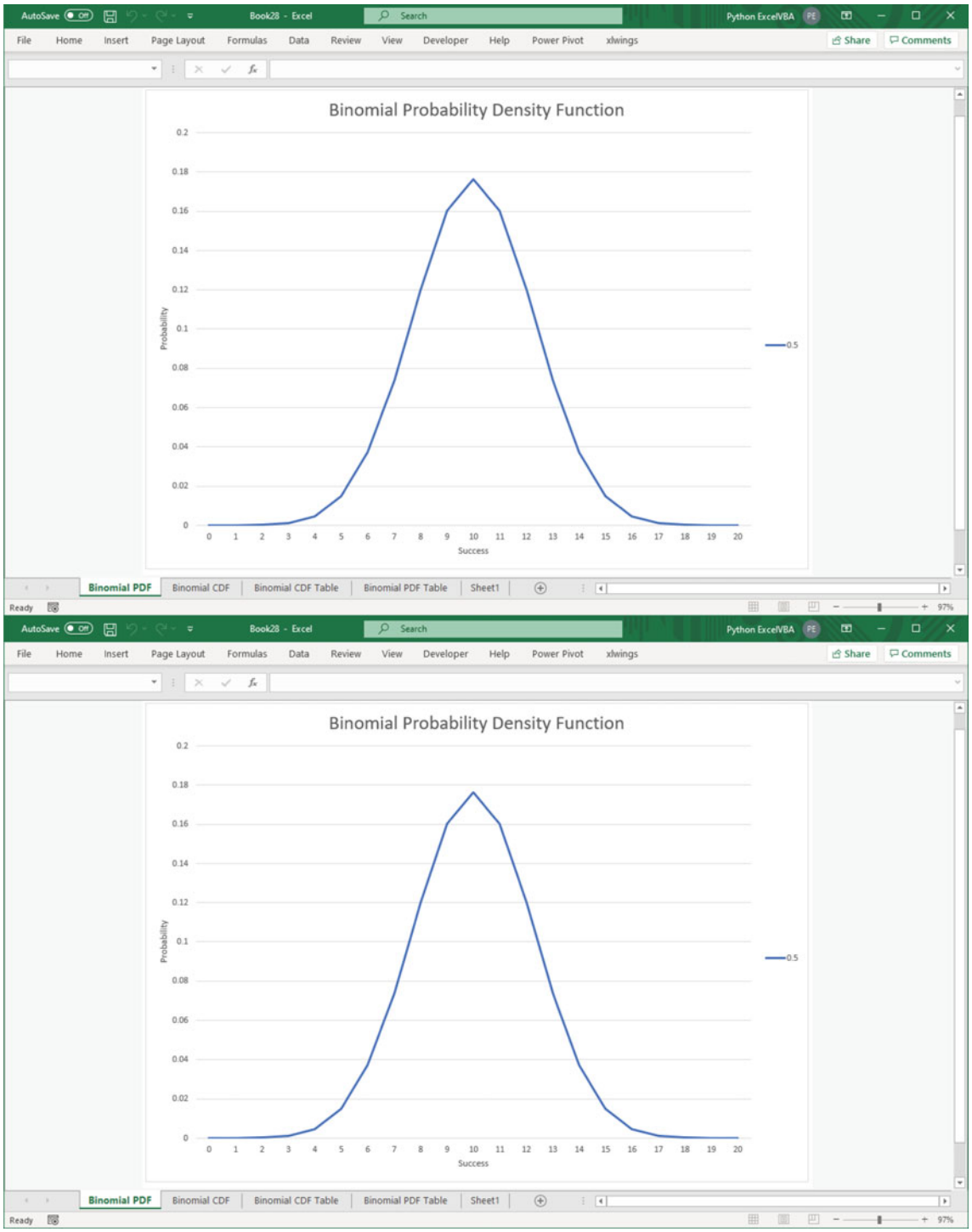


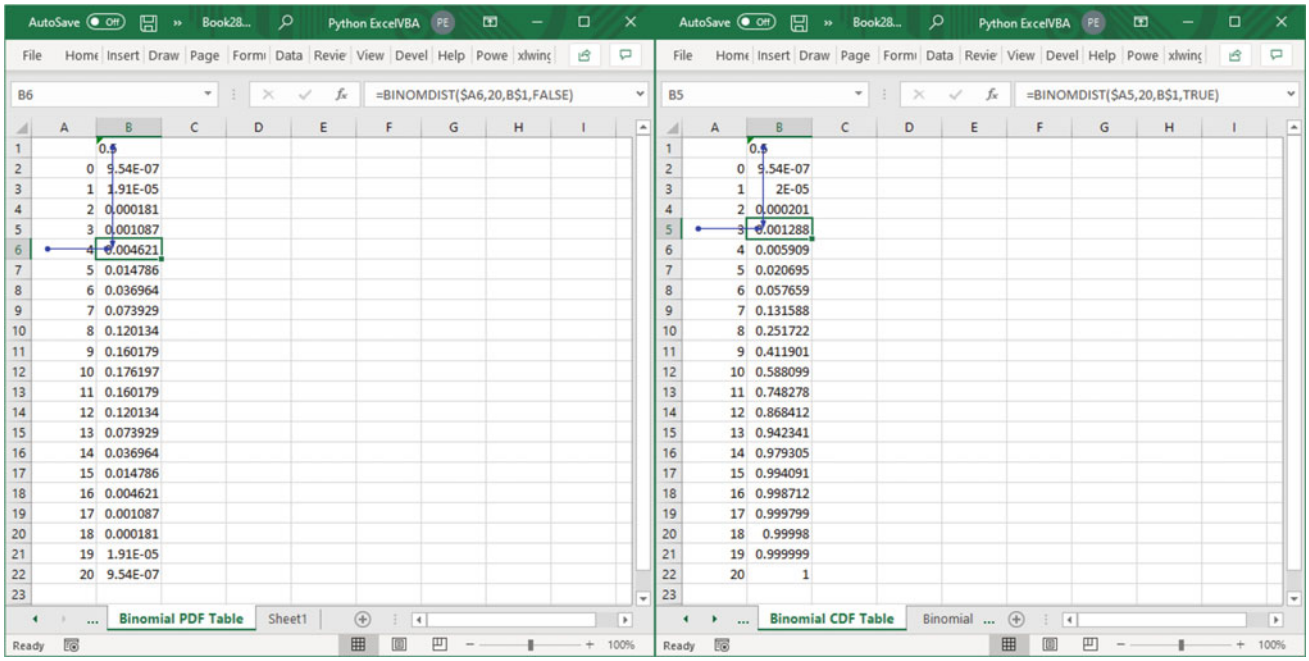
Pushing the *Binomial* button will produce the following dialog box.



Clicking on the *Run* button produces the following four things:

1. Binomial PDF graph.
2. Binomial CDF graph.
3. Binomial PDF table.
4. Binomial CDF table.



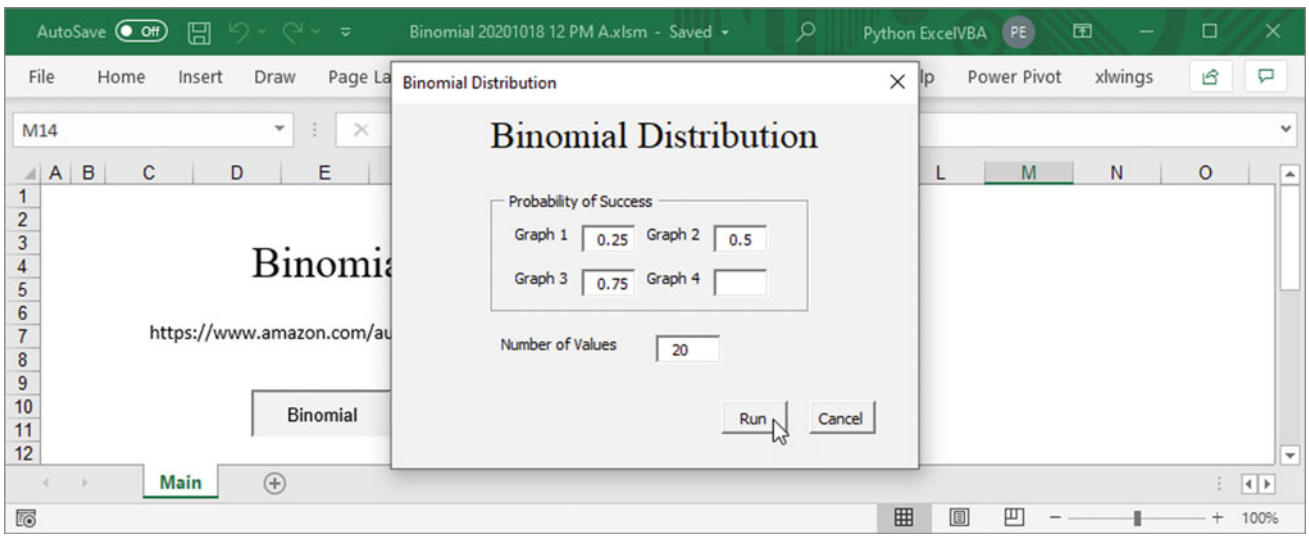


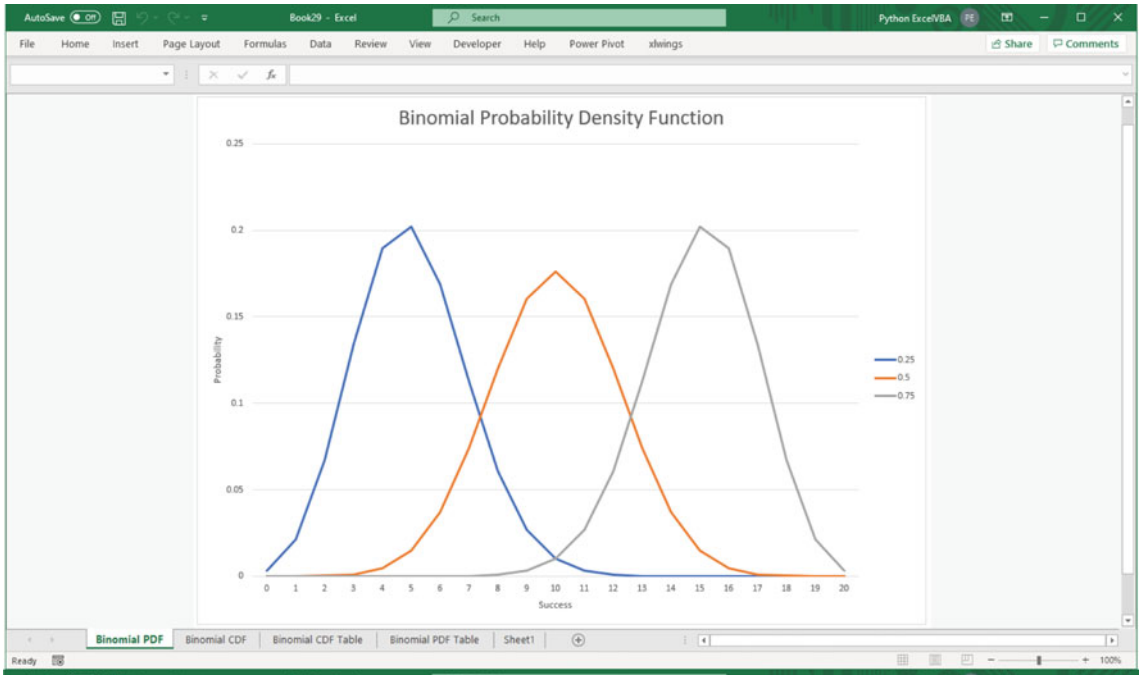
We used the *BINOMDIST* Excel function to create the binomial PDF and CDF distribution. The first parameter is the number of successes. The second parameter is the number of independent trials. The third parameter is the probability of success on each trial. The fourth parameter indicates if it's a cumulative binomial distribution.

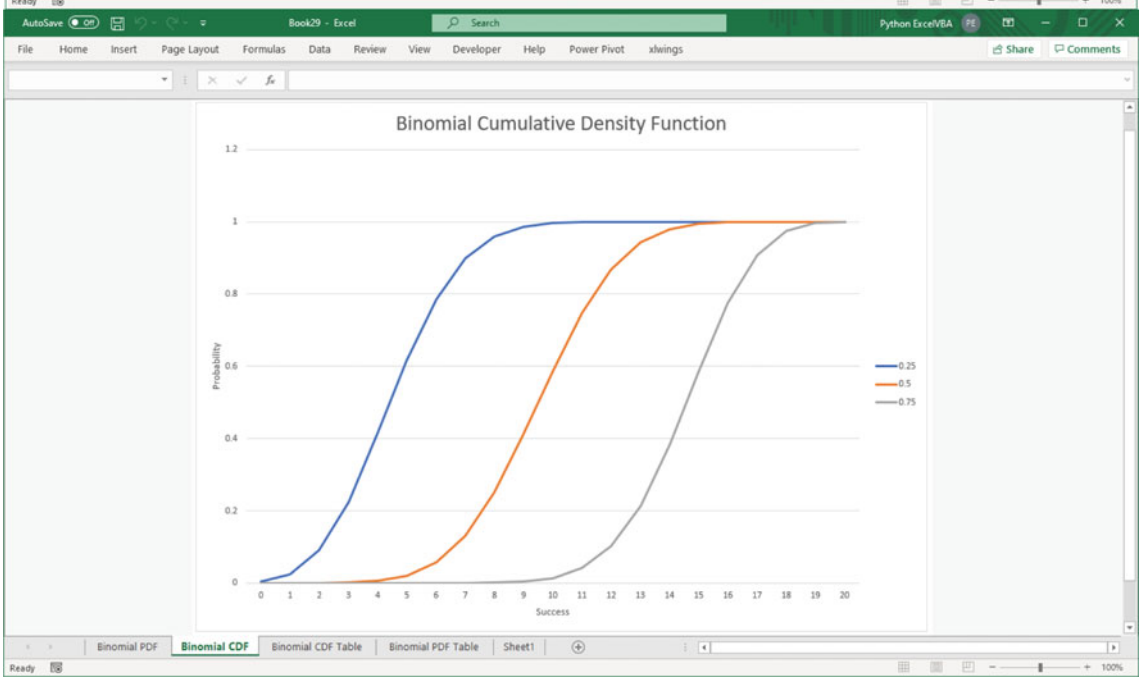
From the binomial distribution, we can see that the best choice would be betting that we would see five heads when flipping a coin ten times. There will be a 24.61% chance that we will see five heads after flipping a coin ten times.

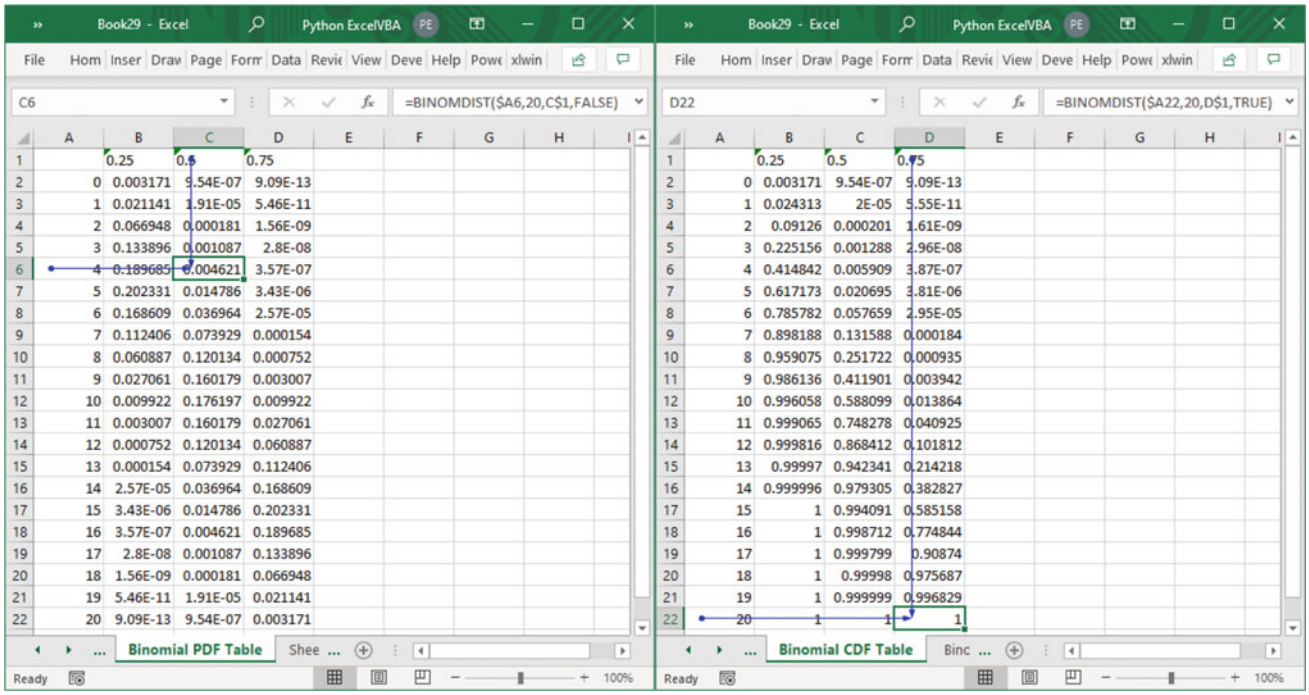
Now suppose in our friendly betting game that there is another game where if we bet $5, we can choose three numbers, and if we are correct, we will win $10. Which three numbers should we choose? The most logical choice would be 3, 4, or 5. We would have a 20 + 24 + 20 = 64% chance of winning.

Let's now create three binomial distributions of ten coin flips. The first distribution will have it, so there is only a 25% chance of getting a Head in each flip. The second binomial distribution will have it so that there would be a 50% chance of getting a Head. The third binomial distribution will have it so that there would be a 75% chance of getting a Head. How will these three distributions compare with each other? Below are the tables and graphs for these three distributions.




Binomial Probability Density Function
Probability
Success
0.25
0.5
0.75
Binomial PDF
Binomial CDF
Binomial CDF Table
Binomial PDF Table
Sheet1


Binomial Cumulative Density Function
Probability
Success
0.25
0.5
0.75
Binomial PDF
Binomial CDF
Binomial CDF Table
Binomial PDF Table
Sheet1



6.3.2 Binomial Distribution in R

The following *R* code populates the variable "binomialData" with binomial distributions with a 25%, 50%, and 75% chance of success.

Below are the first ten rows of the binomial probability density distribution data. Click on the two at the bottom right to see the next ten rows of binomial probability density distribution data.

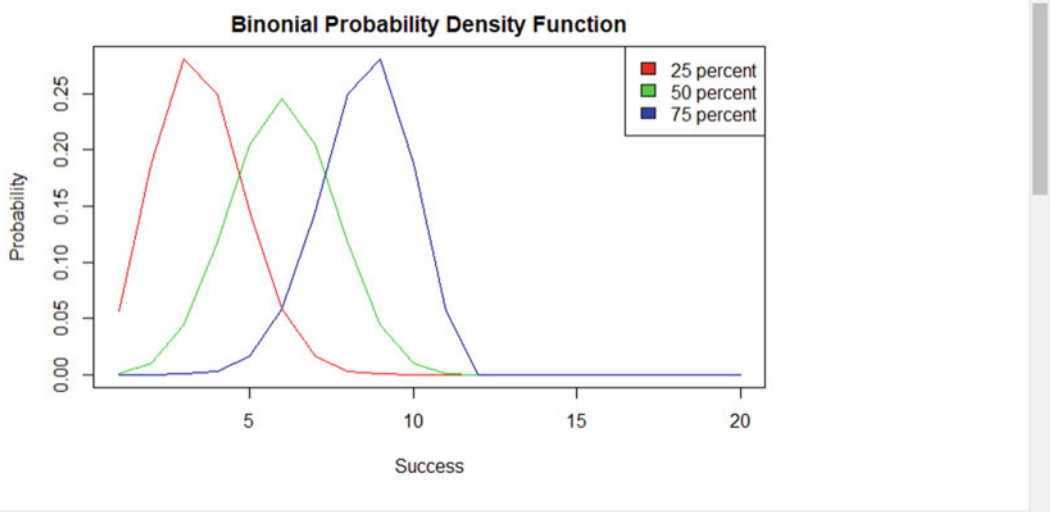
```r
colnames(binomialData)<-c("Success","25 percent","50 percent","75 percent")
binomialData
```

Success <int>	25 percent <dbl>	50 percent <dbl>	75 percent <dbl>
0	5.631351e-02	0.0009765625	9.536743e-07
1	1.877117e-01	0.0097656250	2.861023e-05
2	2.815676e-01	0.0439453125	3.862381e-04
3	2.502823e-01	0.1171875000	3.089905e-03
4	1.459980e-01	0.2050781250	1.622200e-02
5	5.839920e-02	0.2460937500	5.839920e-02
6	1.622200e-02	0.2050781250	1.459980e-01
7	3.089905e-03	0.1171875000	2.502823e-01
8	3.862381e-04	0.0439453125	2.815676e-01
9	2.861023e-05	0.0097656250	1.877117e-01

1-10 of 20 rows Previous 1 2 Next

The R code below graphs three binomial probability density functions. The first probability density function has a 25% chance of success. The second probability density function has a 50% chance of success. The third probability density function has a 75% chance of success. Notice that a binomial probability density function shifts to the right as the percentage of success increases.

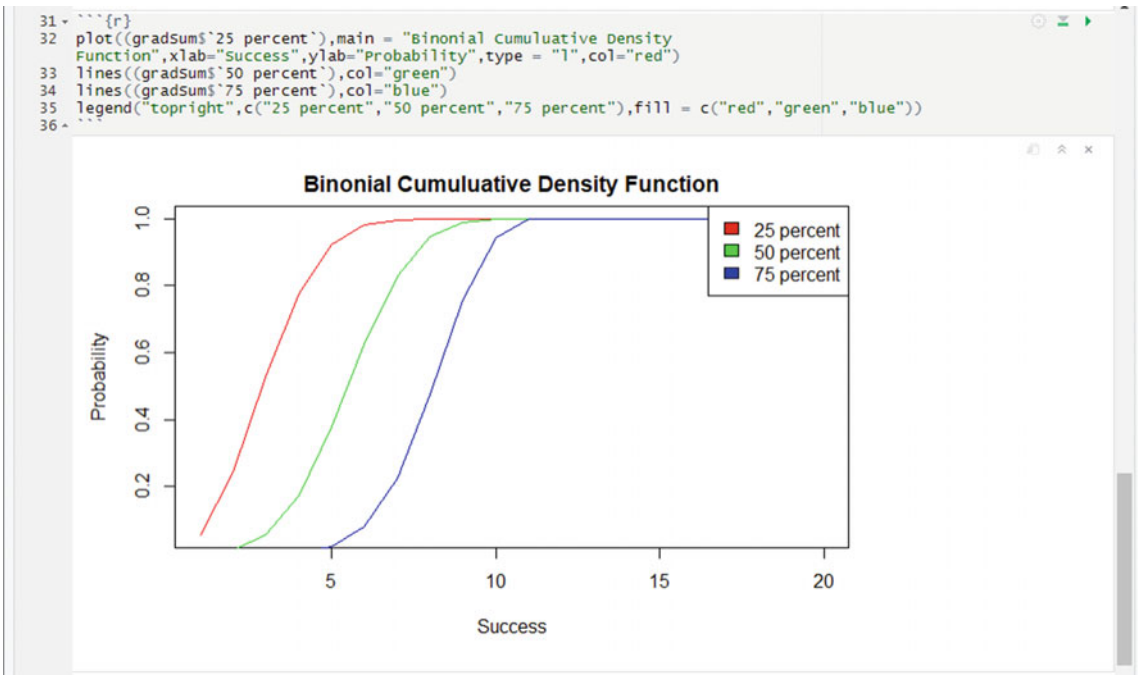
```r
plot((binomialData$`25 percent`),main = "Binonial Probability Density
Function",xlab="Success",ylab="Probability",type = "l",col="red")
lines((binomialData$`50 percent`),col="green")
lines((binomialData$`75 percent`),col="blue")
legend("topright",c("25 percent","50 percent","75 percent"),fill = c("red","green","blue"))
```



Below are the first ten rows of binomial cumulative density distribution data. Click on the two at the bottom right to see the next ten rows of binomial cumulative density distribution data.

C6.5 BinomialRmd

```
```{r}
gradSum = cumsum(binomialData)
gradSum$Success <-c(0:19) # R is zero based
gradSum
```
```

Success <int>	25 percent <dbl>	50 percent <dbl>	75 percent <dbl>
0	0.05631351	0.0009765625	9.536743e-07
1	0.24402523	0.0107421875	2.956390e-05
2	0.52559280	0.0546875000	4.158020e-04
3	0.77587509	0.1718750000	3.505707e-03
4	0.92187309	0.3769531250	1.972771e-02
5	0.98027229	0.6230468750	7.812691e-02
6	0.99649429	0.8281250000	2.241249e-01
7	0.99958420	0.9453125000	4.744072e-01
8	0.99997044	0.9892578125	7.559748e-01
9	0.99999905	0.9990234375	9.436865e-01

1-10 of 20 rows Previous 1 2 Next

The *R* code below graph three binomial cumulative density functions. The first binomial cumulative density function has a 25% chance of success. The second binomial cumulative density function has a 50% chance of success. The third binomial cumulative density has a 75% chance of success. Notice that a binomial probability density function shifts to the right as the percentage of success increases.

```
```{r}
plot((gradSum$`25 percent`),main = "Binonial Cumuluative Density
Function",xlab="Success",ylab="Probability",type = "l",col="red")
lines((gradSum$`50 percent`),col="green")
lines((gradSum$`75 percent`),col="blue")
legend("topright",c("25 percent","50 percent","75 percent"),fill = c("red","green","blue"))
```
```



6.3.3 Binomial Distribution in Python

The following is the Python code to create and graph the binomial distribution:

```
+ Code + Text

[18] #Apr-04-2021
     import numpy as np
     import pandas as pd
     import operator as op
     from functools import reduce
     import matplotlib.pyplot as plt

[19] ##define the function of n choose k
     def ncr(n, r):
         r = min(r, n-r)
         numer = reduce(op.mul, range(n, n-r, -1), 1)
         denom = reduce(op.mul, range(1, r+1), 1)
         return numer / denom
     ##define the calculation

     def calculate_pdf (n, k, p):
         return ncr(n, k)*pow(p,k)*pow((1-p),n-k)

[20] probability = np.zeros((31,3))##set the dataframe for probability
     p = [0.25,0.5,0.75]
     for i in range(3):
         for j in range(31):
             probability[j,i] = calculate_pdf(30,j,p[i])

     prob = pd.DataFrame(probability)
     prob.columns = ['25 percent', '50 percent','75 percent']

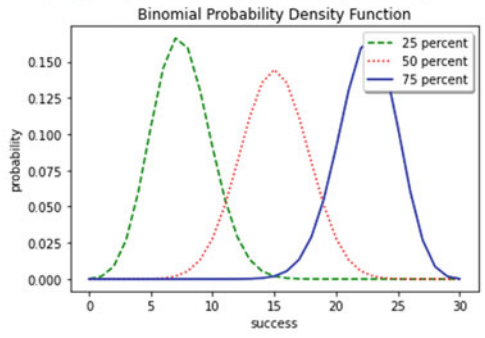
     #compute culmulated probability density
     cdf = prob.cumsum()
```

Below graphs the binomial probability density distributions where the chances of success are 25%, 50%, and 75%.

```
fig, ax = plt.subplots()
ax.plot(prob['25 percent'],'g--', label='25 percent')
ax.plot(prob['50 percent'],'r:', label='50 percent')
ax.plot(prob['75 percent'],'b', label='75 percent')

legend = ax.legend(loc='upper right', shadow=True)
plt.xlabel('success')
plt.ylabel('probability')
plt.title('Binomial Probability Density Function')

Text(0.5, 1.0, 'Binomial Probability Density Function')
```

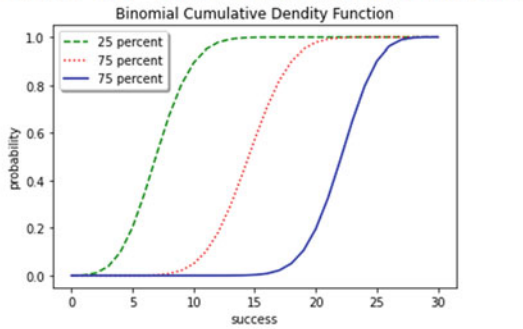


Below shows the binomial cumulative probability density distributions where the chances of success are 25%, 50%, and 75%.

```
[7] fig2, bx = plt.subplots()
    bx.plot(cdf['25 percent'],'g--', label='25 percent')
    bx.plot(cdf['50 percent'],'r:', label='75 percent')
    bx.plot(cdf['75 percent'],'b', label='75 percent')

    legend = bx.legend(loc=0, shadow=True)
    plt.xlabel('success')
    plt.ylabel('probability')
    plt.title('Binomial Cumulative Dendity Function')
```

```
Text(0.5, 1.0, 'Binomial Cumulative Dendity Function')
```



6.4 Poisson Random Variable

In the previous section, the random variable was defined in terms of the number of successes. These successes were *obtained within a fixed number of trials* of some random experiment. In this section, we will discuss a distribution called the *Poisson distribution.* This distribution can be used to deal with a single type of outcome or "event."

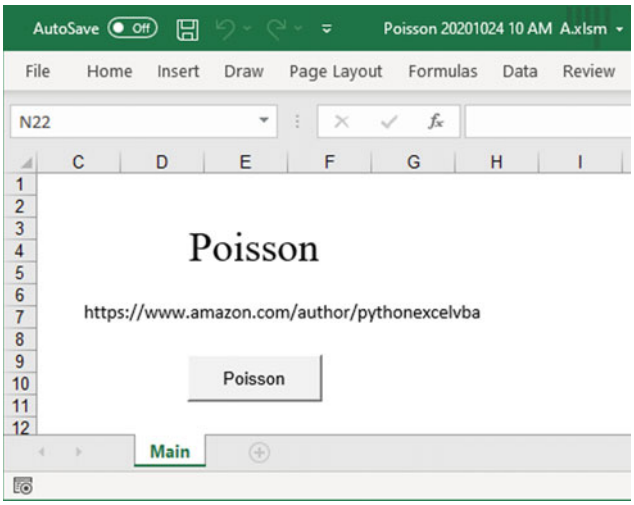
The formula for the density function of the Poisson distribution is

$$e^{-\lambda}\lambda^{x}/x! \text{ for x } = 0, 1, 2, 3, \ldots \text{ and } \lambda > 0$$

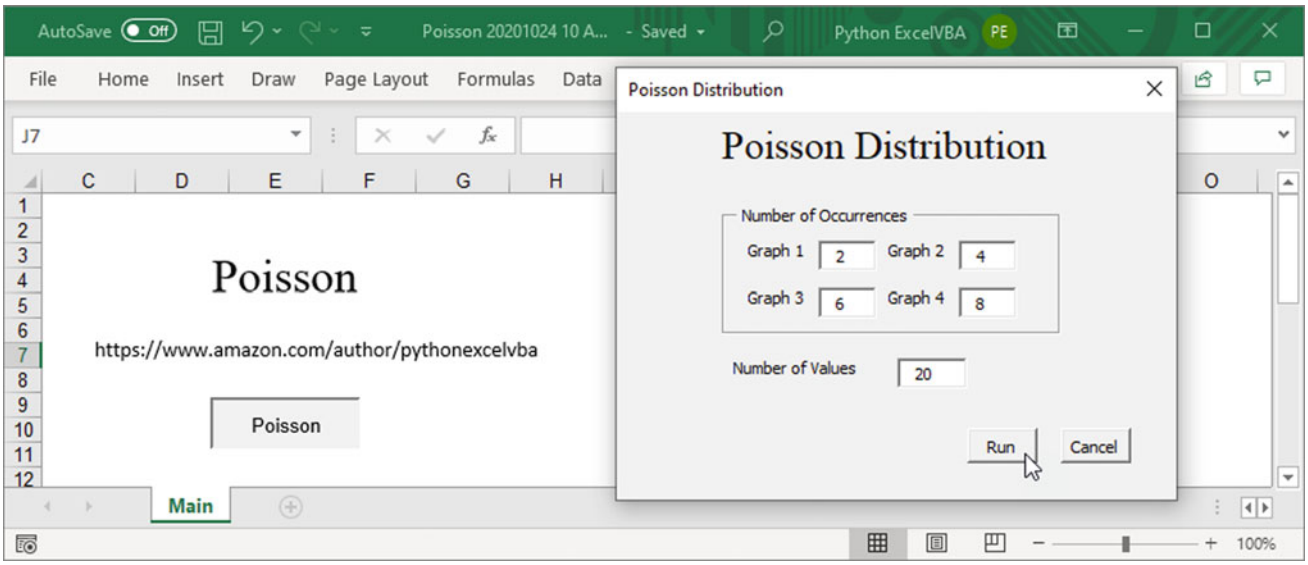
where x represents the number of rare events in a unit of time, space, or volume; λ is the mean value of x; e is the base of natural logarithms and is approximately equal to 2.71828; and ! is the factorial symbol.

6.4.1 Poisson Distribution in Excel

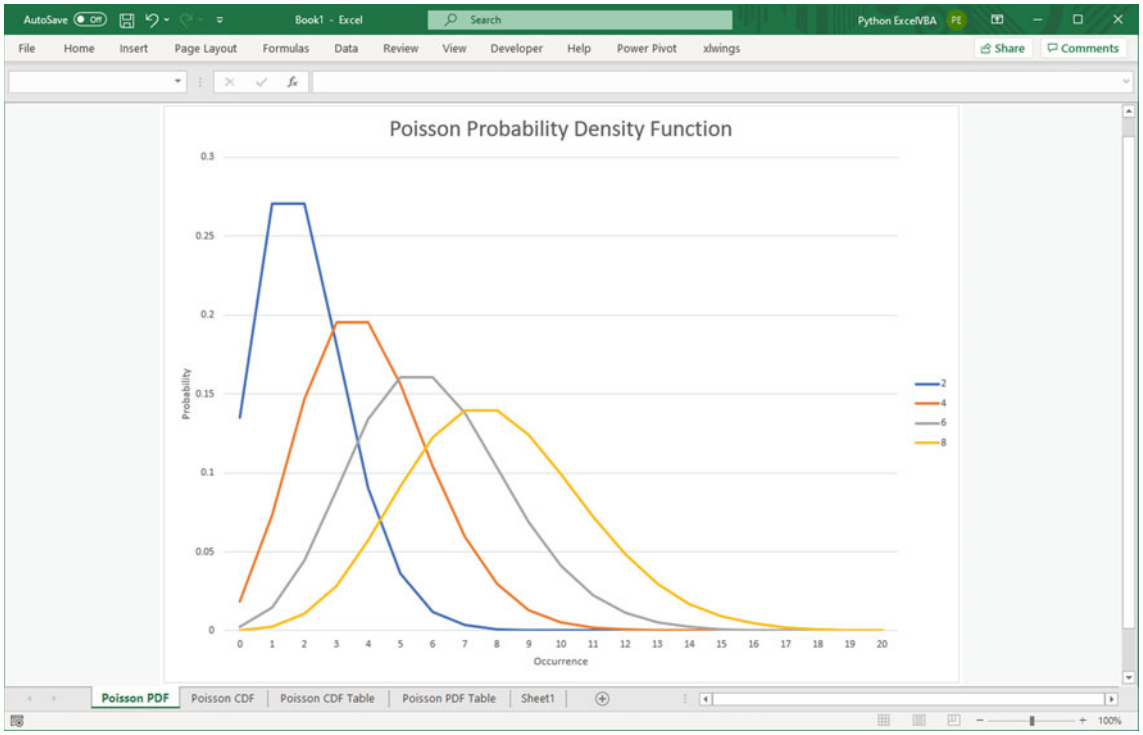
To analyze the Poisson distribution, we will look at the workbook *Poisson.xlsm*.



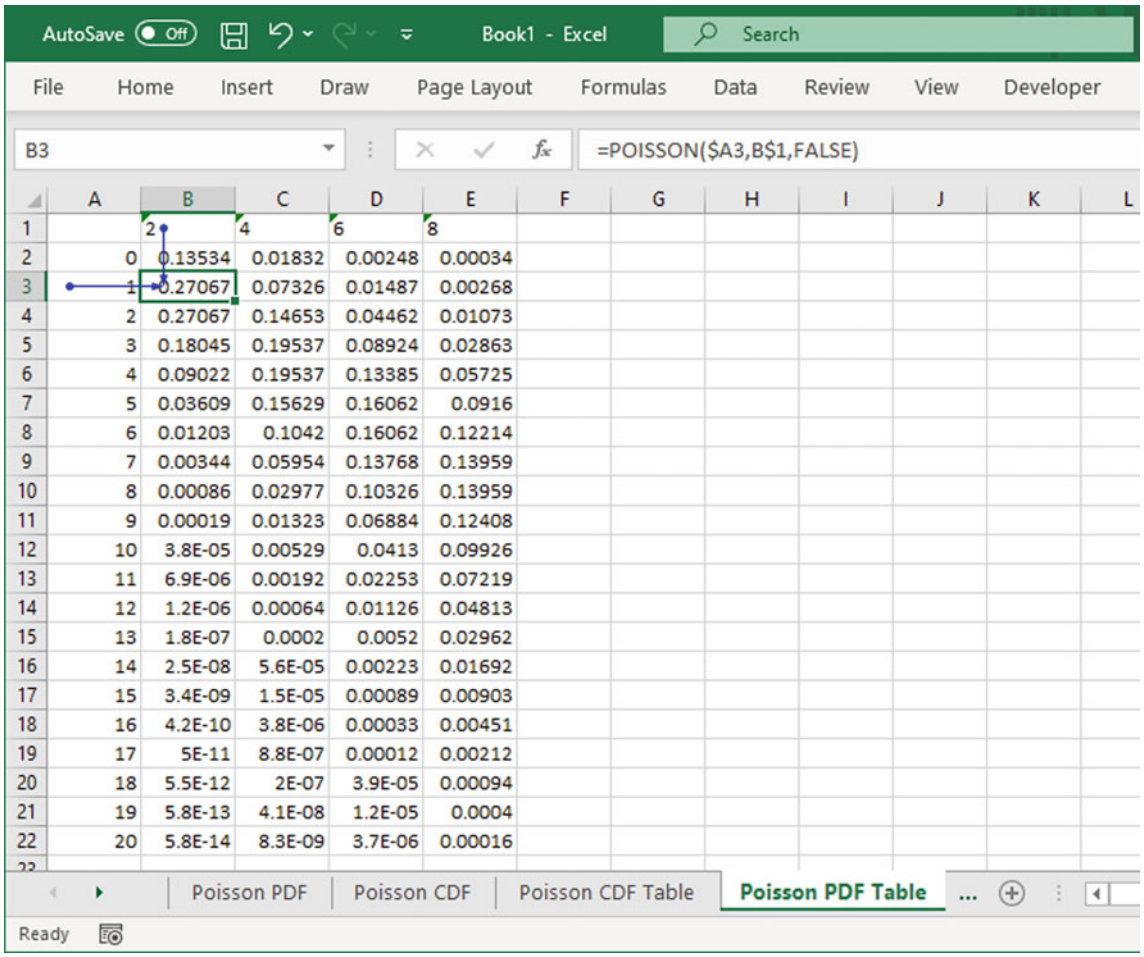
We will now analyze four different Poisson distributions with 4, 6, 8, and 10 occurrences. Press the *Poisson* button above to create these four Poisson distributions. Pressing the *Poisson* button will bring the *Poisson Distribution* dialog box shown below.



Fill out the dialog box above and then press the *Run* button to the four Poisson distributions. The dialog box below shows the four PDF Poisson distributions.



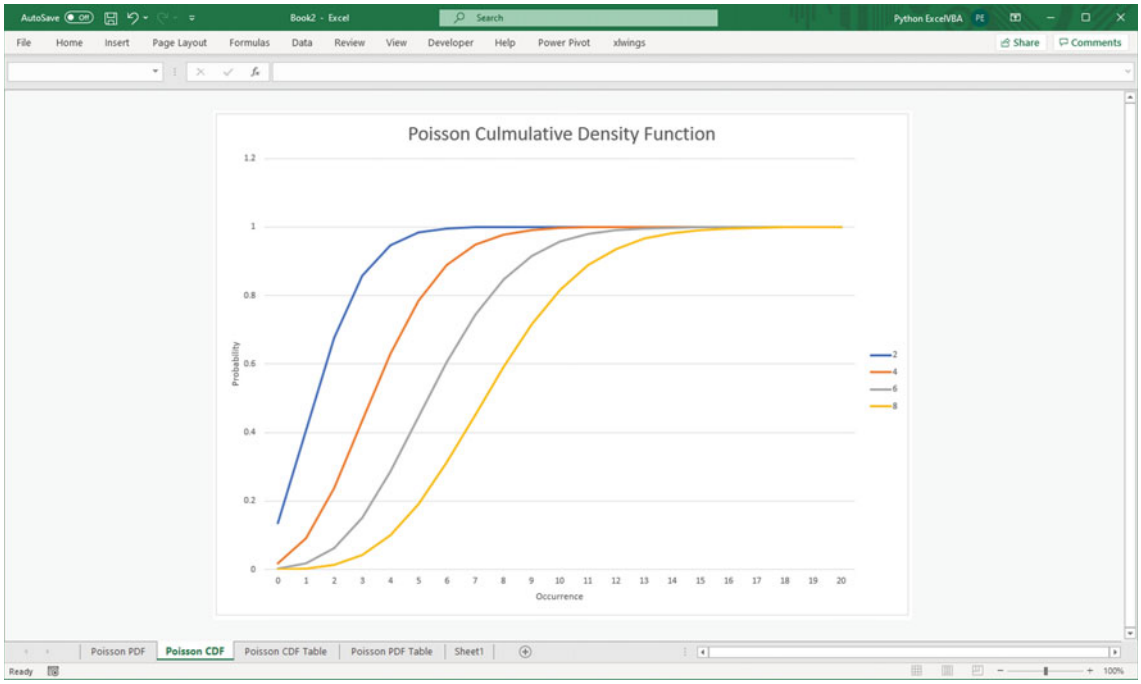
The worksheet below shows a Poisson PDF table for the four Poisson probability density distributions.



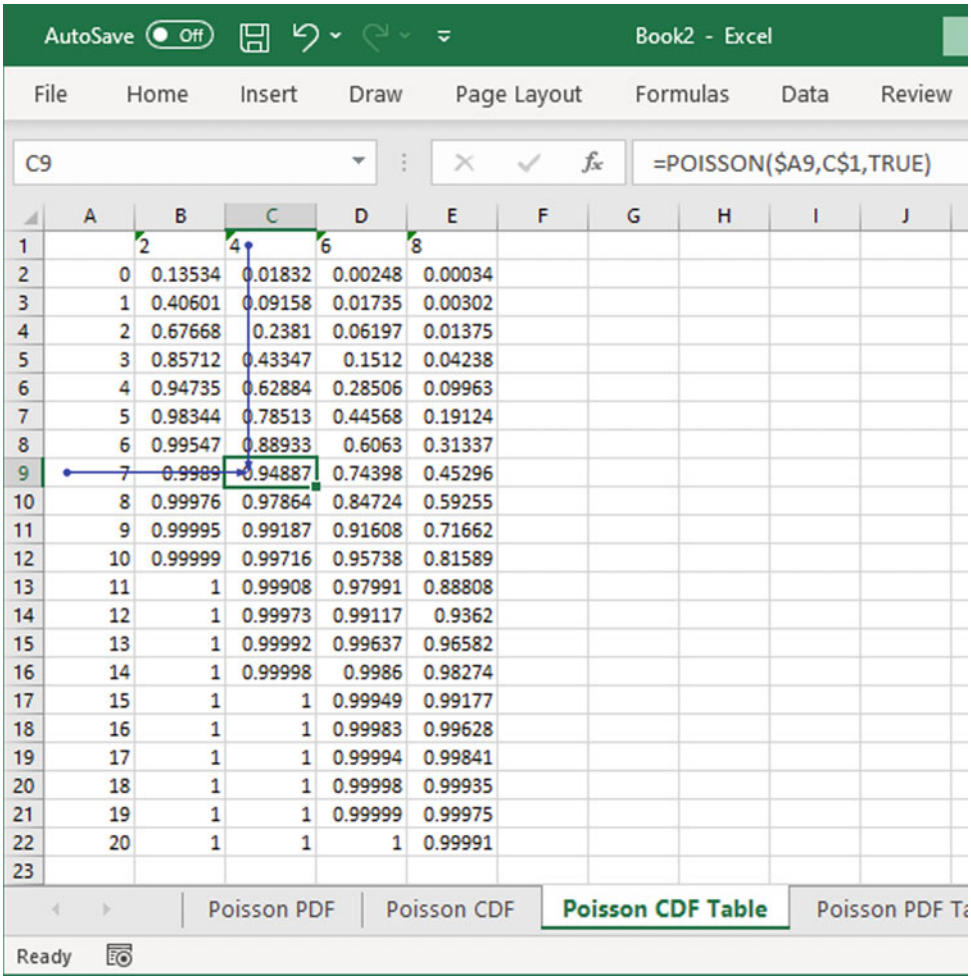
	A	B	C	D	E
1		2	4	6	8
2	0	0.13534	0.01832	0.00248	0.00034
3	1	0.27067	0.07326	0.01487	0.00268
4	2	0.27067	0.14653	0.04462	0.01073
5	3	0.18045	0.19537	0.08924	0.02863
6	4	0.09022	0.19537	0.13385	0.05725
7	5	0.03609	0.15629	0.16062	0.0916
8	6	0.01203	0.1042	0.16062	0.12214
9	7	0.00344	0.05954	0.13768	0.13959
10	8	0.00086	0.02977	0.10326	0.13959
11	9	0.00019	0.01323	0.06884	0.12408
12	10	3.8E-05	0.00529	0.0413	0.09926
13	11	6.9E-06	0.00192	0.02253	0.07219
14	12	1.2E-06	0.00064	0.01126	0.04813
15	13	1.8E-07	0.0002	0.0052	0.02962
16	14	2.5E-08	5.6E-05	0.00223	0.01692
17	15	3.4E-09	1.5E-05	0.00089	0.00903
18	16	4.2E-10	3.8E-06	0.00033	0.00451
19	17	5E-11	8.8E-07	0.00012	0.00212
20	18	5.5E-12	2E-07	3.9E-05	0.00094
21	19	5.8E-13	4.1E-08	1.2E-05	0.0004
22	20	5.8E-14	8.3E-09	3.7E-06	0.00016

The above table uses the Excel *Poisson* function to create the Poisson distribution. The *Poisson* function has three arguments. The first argument is the number of events. The second argument is the expected numeric value. The third argument indicates if the cumulative Poisson distribution should be calculated or not.

The chart below shows the four CDF Poisson distributions.



The worksheet below shows a Poisson CDF table for the four Poisson cumulative density functions.



	A	B	C	D	E
1		2	4	6	8
2	0	0.13534	0.01832	0.00248	0.00034
3	1	0.40601	0.09158	0.01735	0.00302
4	2	0.67668	0.2381	0.06197	0.01375
5	3	0.85712	0.43347	0.1512	0.04238
6	4	0.94735	0.62884	0.28506	0.09963
7	5	0.98344	0.78513	0.44568	0.19124
8	6	0.99547	0.88933	0.6063	0.31337
9	7	0.9989	0.94887	0.74398	0.45296
10	8	0.99976	0.97864	0.84724	0.59255
11	9	0.99995	0.99187	0.91608	0.71662
12	10	0.99999	0.99716	0.95738	0.81589
13	11	1	0.99908	0.97991	0.88808
14	12	1	0.99973	0.99117	0.9362
15	13	1	0.99992	0.99637	0.96582
16	14	1	0.99998	0.9986	0.98274
17	15	1	1	0.99949	0.99177
18	16	1	1	0.99983	0.99628
19	17	1	1	0.99994	0.99841
20	18	1	1	0.99998	0.99935
21	19	1	1	0.99999	0.99975
22	20	1	1	1	0.99991
23					

Note that the third parameter in the *POISSON* function is set to true to indicate that cumulative Poisson distribution should be calculated.

6.4.2 Poisson Distribution in R

The following *R* code populates the variable "PoissonData" with Poisson distributions with 4, 6, 8, and 10 occurrences:

Below are the first ten rows of the Poisson probability density distribution data. Click on the two at the bottom right to see the next ten rows of Poisson probability density distribution data.

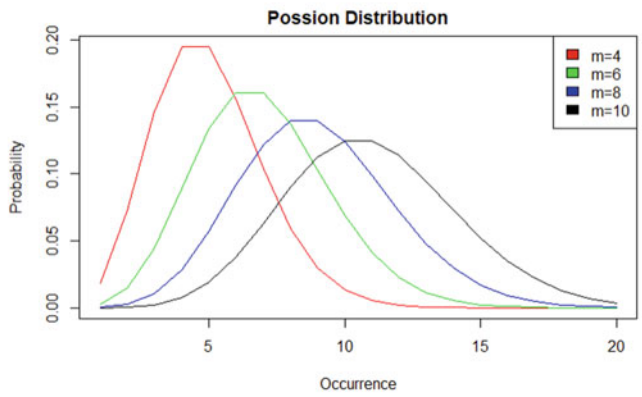
```r
colnames(PoissonData)<-c("Occurrence","m=4","m=6","m=8","m=10")
PoissonData
```

Occurrence <int>	m=4 <dbl>	m=6 <dbl>	m=8 <dbl>	m=10 <dbl>
0	1.831564e-02	2.478752e-03	0.0003354626	4.539993e-05
1	7.326256e-02	1.487251e-02	0.0026837010	4.539993e-04
2	1.465251e-01	4.461754e-02	0.0107348041	2.269996e-03
3	1.953668e-01	8.923508e-02	0.0286261442	7.566655e-03
4	1.953668e-01	1.338526e-01	0.0572522885	1.891664e-02
5	1.562935e-01	1.606231e-01	0.0916036616	3.783327e-02
6	1.041956e-01	1.606231e-01	0.1221382155	6.305546e-02
7	5.954036e-02	1.376770e-01	0.1395865320	9.007923e-02
8	2.977018e-02	1.032577e-01	0.1395865320	1.125990e-01
9	1.323119e-02	6.883849e-02	0.1240769173	1.251100e-01

1-10 of 20 rows Previous 1 2 Next

The graph below shows four Poisson probability density functions.

```r
plot((PoissonData$`m=4`),main = "Possion Distribution",xlab="Occurrence",ylab="Probability",type = "l",col="red")
lines((PoissonData$`m=6`),col="green")
lines((PoissonData$`m=8`),col="blue")
lines((PoissonData$`m=10`),col="black")
legend("topright",c("m=4","m=6","m=8","m=10"),fill = c("red","green","blue","black"))
```



Below are the first ten rows of the Poisson cumulative density distribution data. Click on the two at the bottom right to see the Poisson cumulative density distribution data's next ten rows.

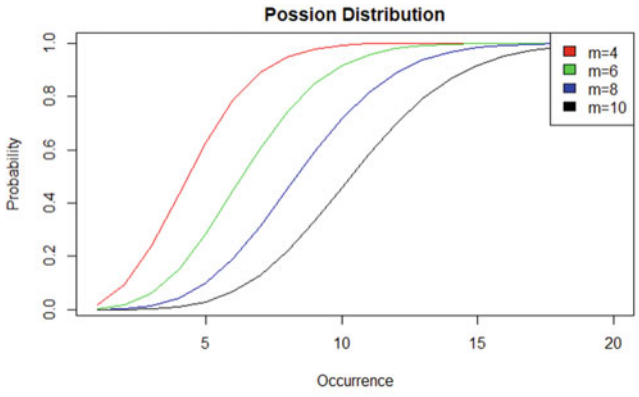
```r
PoissonData_Culmulative = cumsum(PoissonData)
PoissonData_Culmulative$Occurrence <-c(0:19)
PoissonData_Culmulative
```

Occurrence <int>	m=4 <dbl>	m=6 <dbl>	m=8 <dbl>	m=10 <dbl>
0	0.01831564	0.002478752	0.0003354626	4.539993e-05
1	0.09157819	0.017351265	0.0030191637	4.993992e-04
2	0.23810331	0.061968804	0.0137539677	2.769396e-03
3	0.43347012	0.151203883	0.0423801120	1.033605e-02
4	0.62883694	0.285056500	0.0996324005	2.925269e-02
5	0.78513039	0.445679641	0.1912360621	6.708596e-02
6	0.88932602	0.606302782	0.3133742775	1.301414e-01
7	0.94886638	0.743979760	0.4529608095	2.202206e-01
8	0.97863657	0.847237494	0.5925473414	3.328197e-01
9	0.99186776	0.916075983	0.7166242587	4.579297e-01

1-10 of 20 rows Previous 1 2 Next

The graph below shows four Poisson cumulative density functions.

```r
plot((PoissonData_Culmulative$`m=4`),main = "Possion Distribution",xlab="Occurrence",ylab="Probability",type = "l",col="red")
lines((PoissonData_Culmulative$`m=6`),col="green")
lines((PoissonData_Culmulative$`m=8`),col="blue")
lines((PoissonData_Culmulative$`m=10`),col="black")
legend("topright",c("m=4","m=6","m=8","m=10"),fill = c("red","green","blue","black"))
```



6.4.3 Poisson Distribution in Python

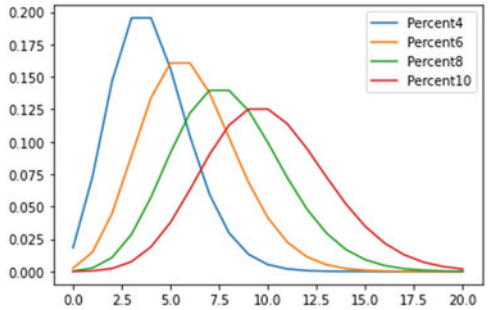
The following Python code populates the variable "PoissonData" with Poisson distributions with 4, 6, 8, and 10 occurrences:

```
+ Code + Text

[2]  #Date: Apr-04-2021 9 AM
     import numpy as np
     import math
     import pandas as pd

[3]  def possion_pdf(x, mu):
       """
       calculate possion pdf
       :param x: list, a list of success count
       :param mu: int or float,  the mean of possion
       """
       return [(np.e ** -mu) * (mu ** i) / math.factorial(i) for i in x]

[4]  Percent4 = possion_pdf(list(range(21)), 4)
     Percent6 = possion_pdf(list(range(21)), 6)
     Percent8 = possion_pdf(list(range(21)), 8)
     Percent10 = possion_pdf(list(range(21)), 10)
     # preparing dataframe
     pre_df = {'Percent4':Percent4,
               'Percent6':Percent6,
               'Percent8':Percent8,
               'Percent10':Percent10}

     df = pd.DataFrame(pre_df)
```

The following Python code graphs the Poisson probability density distribution with 4, 6, 8, and 10 occurrences:

```
# the pdf plot
df.plot()
```

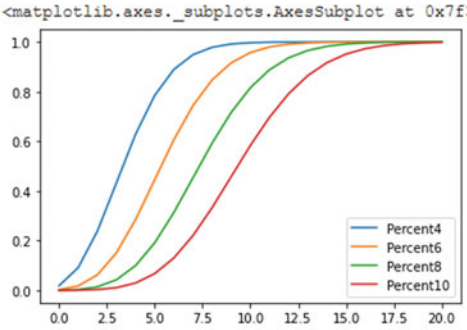
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6300e84310>
```



The following Python code graphs the cumulative Poisson probability density distribution with 4, 6, 8, and 10 occurrences:

```
[ ] # cdf plot
df.cumsum().plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3fe61fb588>
```

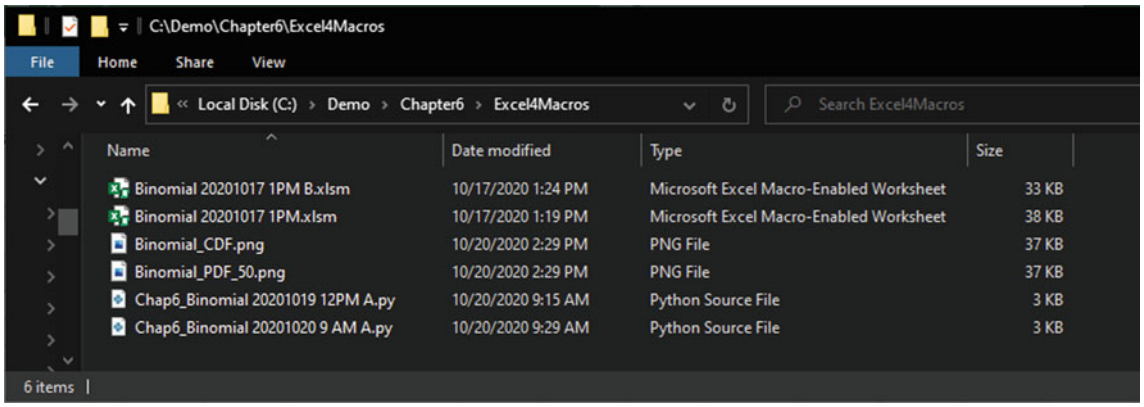


6.5 Excel 4.0 Macro Functions and Excel Names

Excel *Names* have many beneficial uses. The most popular use of Excel names is to point to cell range. Other uses of Excel name are to define an Excel name as a constant or a formula. Most people do not realize that you can use Excel 4.0 macro functions in Excel names; this allows you to do things that would require VBA. Excel 4.0 macro functions that returned values can be used in Excel names. The URL https://exceloffthegrid.com/using-excel-4-macro-functions/ has a PDF that lists Excel 4.0 macro functions.

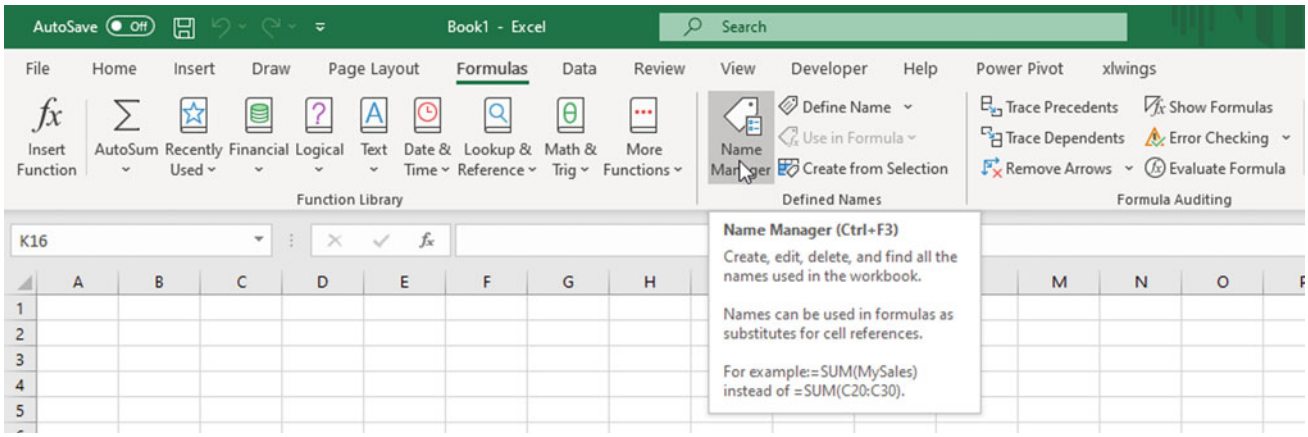
6.6 Examples

6.6.1 Files Function

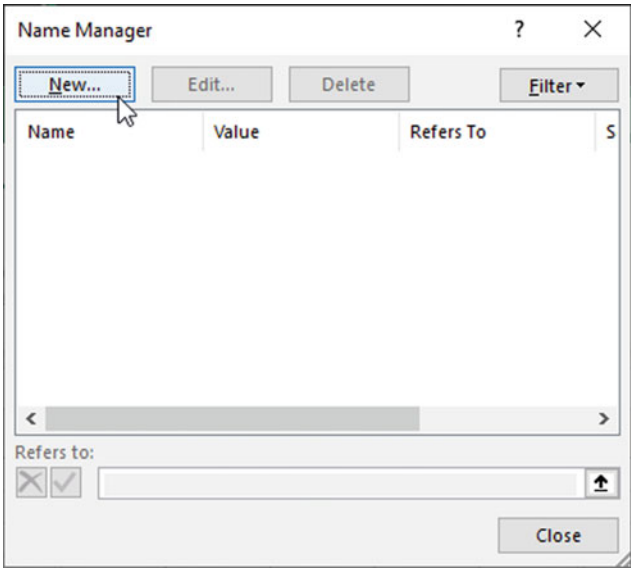


This section demonstrates using the *Files* Excel 4.0 macro function to retrieve the list of names in the directory"C:\Demo \Chapter6\Excel4Macro."

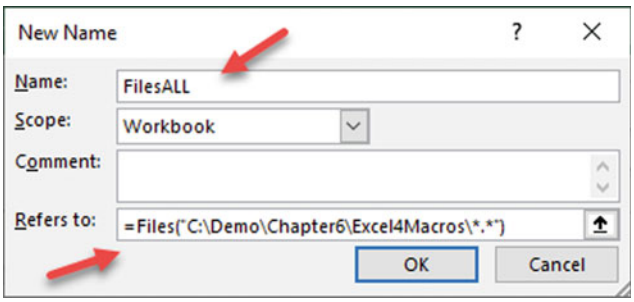
(1) Click on *Name Manger* in the *Formulas* tab.



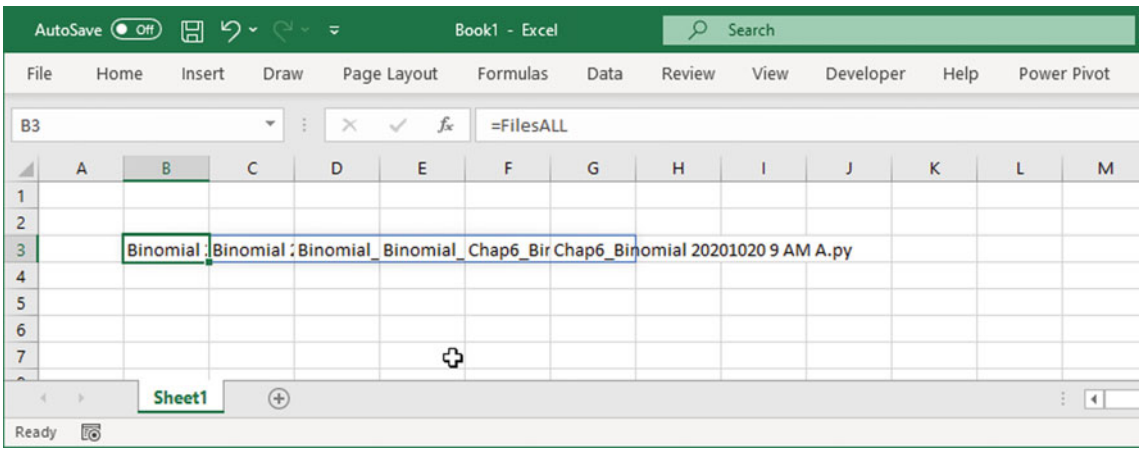
(2) Click on the *New* button.



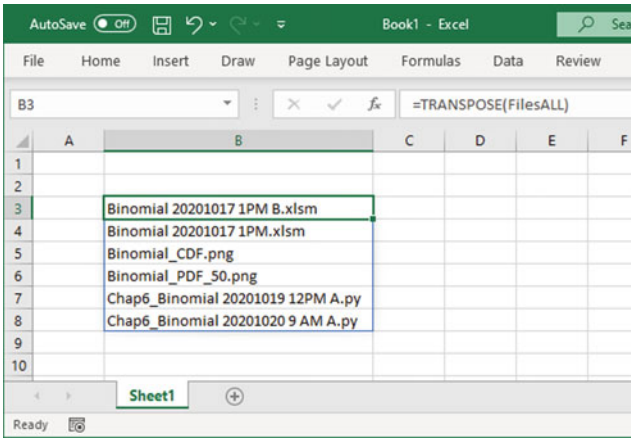
(3) Create a name called *FilesAll*. This name should refer to = *Files("C:\Demo\Chapter6\Excel4Macros*.*")*



4) In cell b3, enter " = FilesAll" The files are returned in a horizontal list.



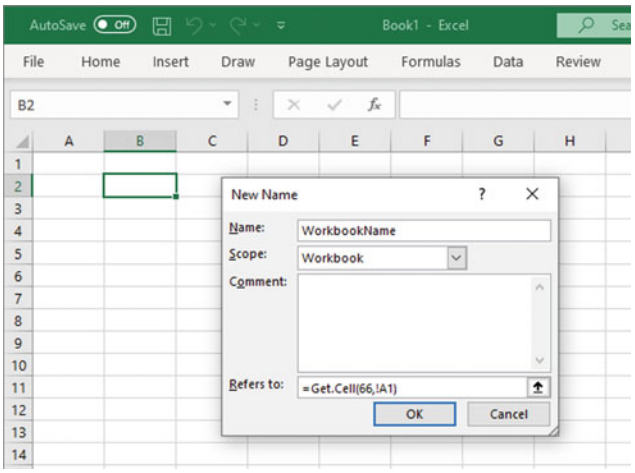
(5) In cell b3, enter " = TRANSPOSE(FilesAll)" This will transpose the horizontal list to a vertical list.



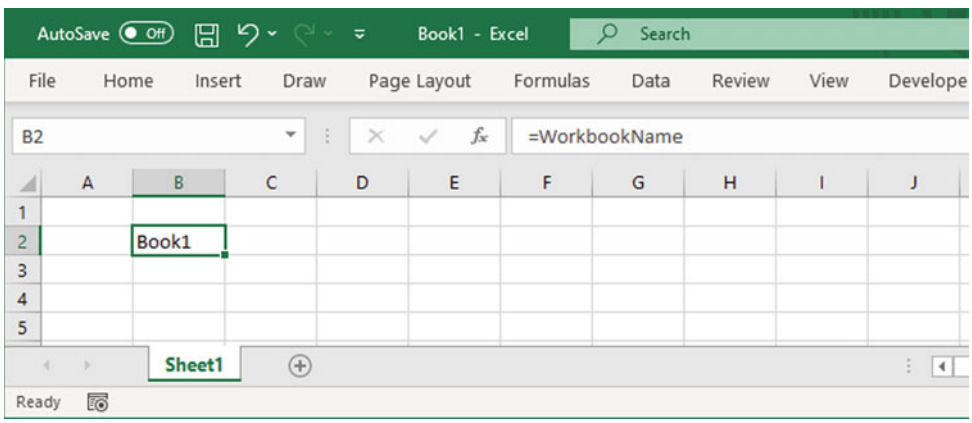
6.6.1.1 Get.CELLS Function

In this section, we will demonstrate the *Get.CELL* function to retrieve the name of the workbook:

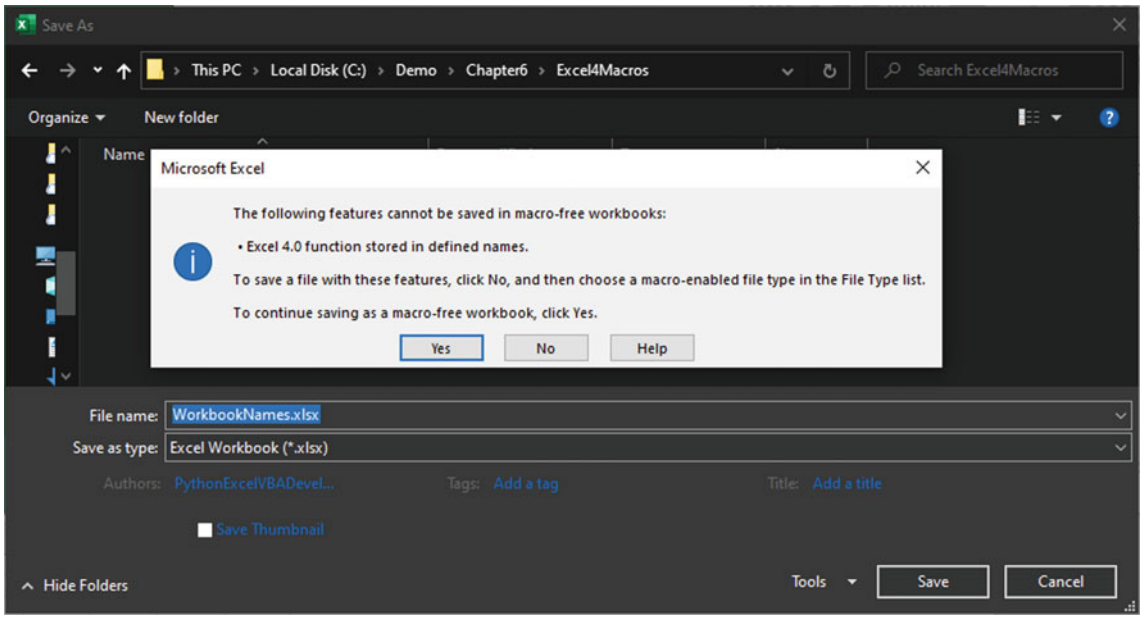
(1) Click on *Name Manger* in the *Formulas* tab.
(2) Click on the *New* button.
(3) Create a name called *WorkbookName*. This name should refer to = *Get.Cell(66,!A1).*



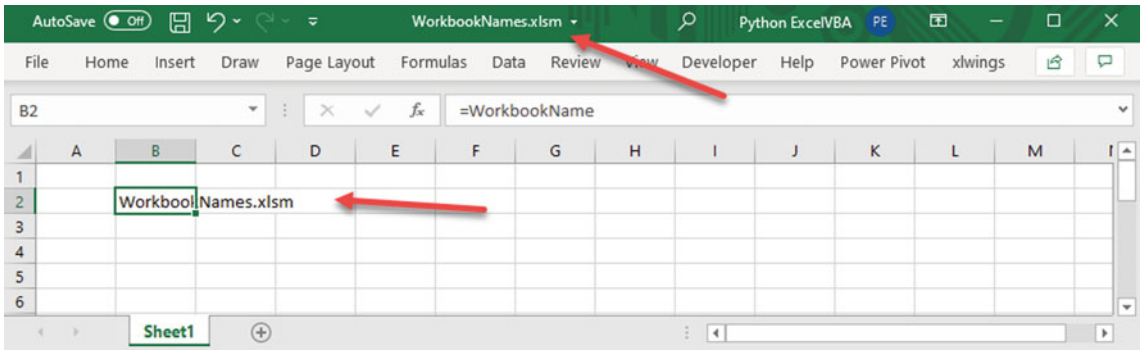
In cell b2, enter " = WorkbookName."



A workbook that uses Excel 4.0 macros in Excel names needs to save the workbook with an “xlsm” file extension, this extension that the workbook contains macros. If you save the workbook as an”xlsx” extension, Excel will give you the following warning:



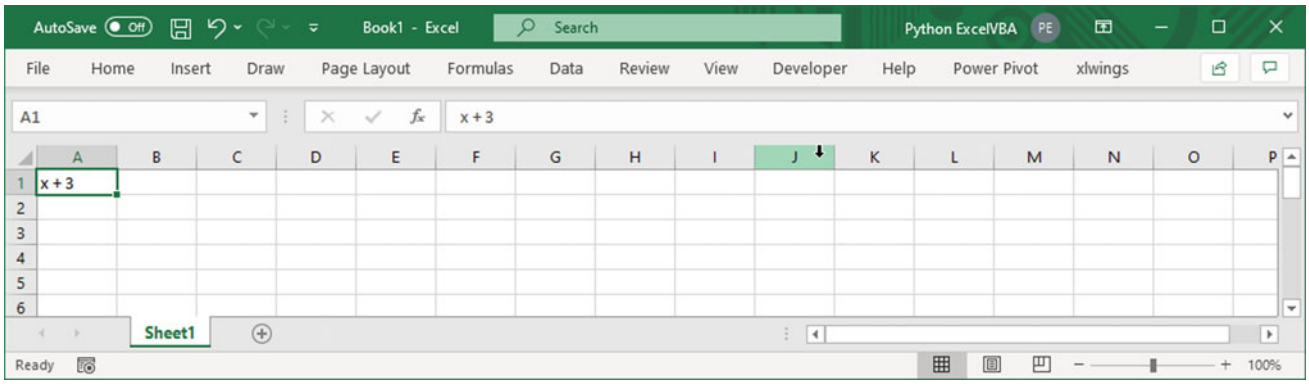
After the workbook was saved as an “xlsm” workbook, the Excel 4 macro function displayed the new workbook name.



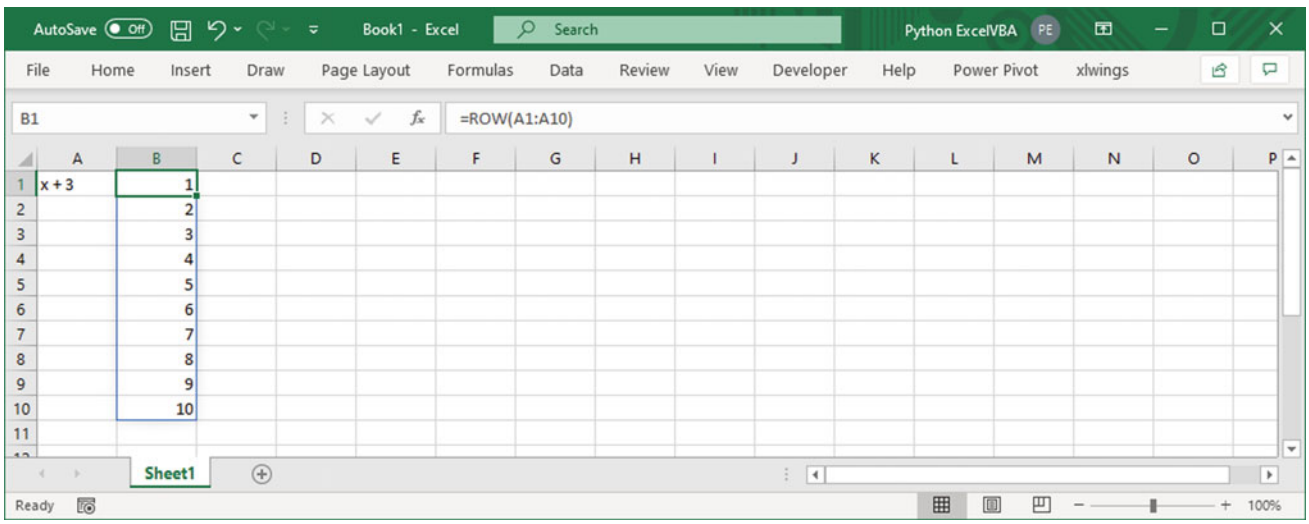
6.6.1.2 EVALUATE Function

The *EVALUATE* Excel 4.0 macro function evaluates a text formula. The example below will make it more clear what is meant by “evaluates a text formula.”

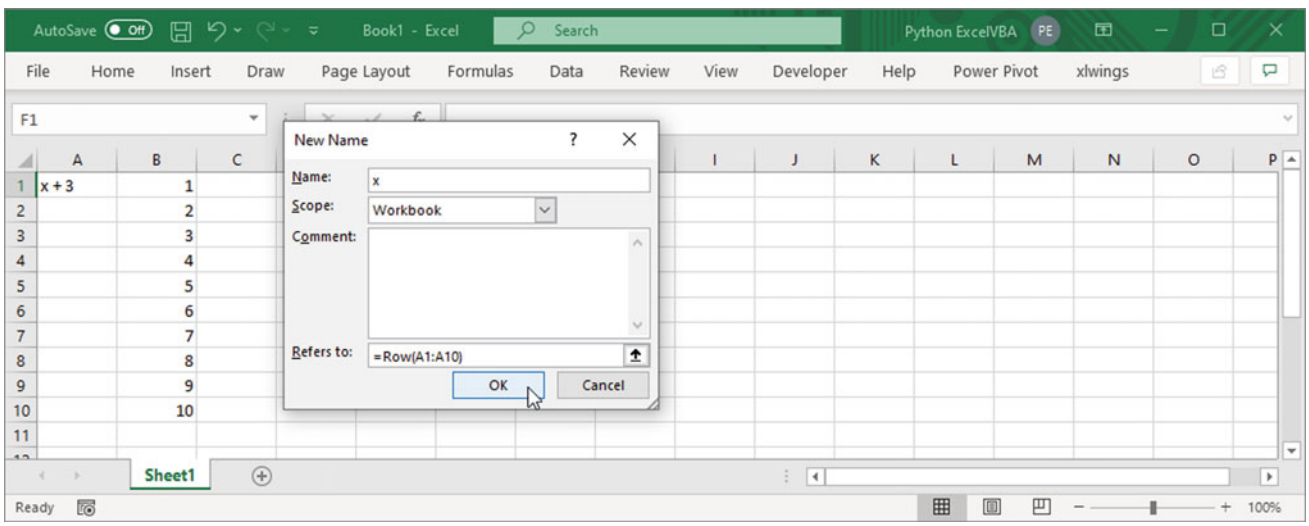
Suppose we would like to evaluate the formula "x + 3" from 1 to 10.



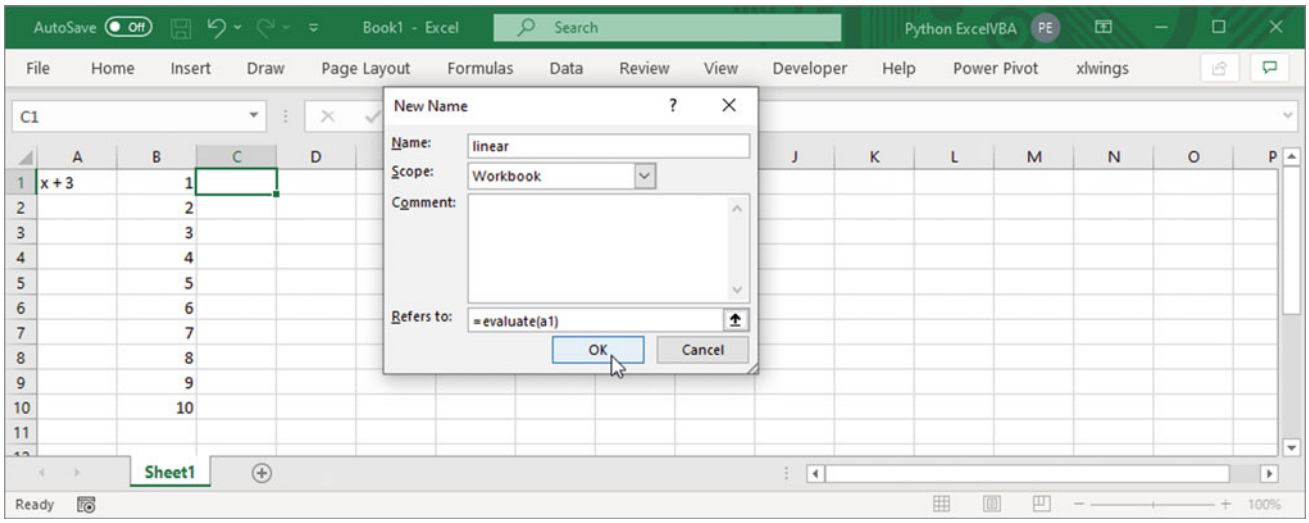
The *Row* Excel function returns all the rows in a range reference. Below shows the *Row* function return the list of rows associated with the range a1–a10.



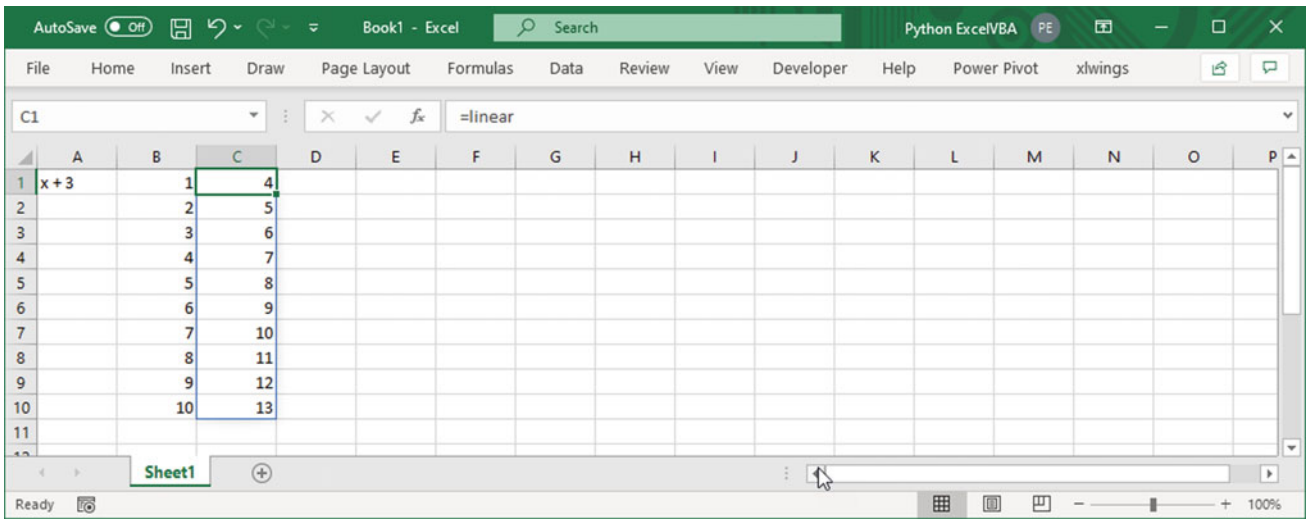
Let's now define an Excel Name as *x* shown below:



Let's now define *linear* as shown below:

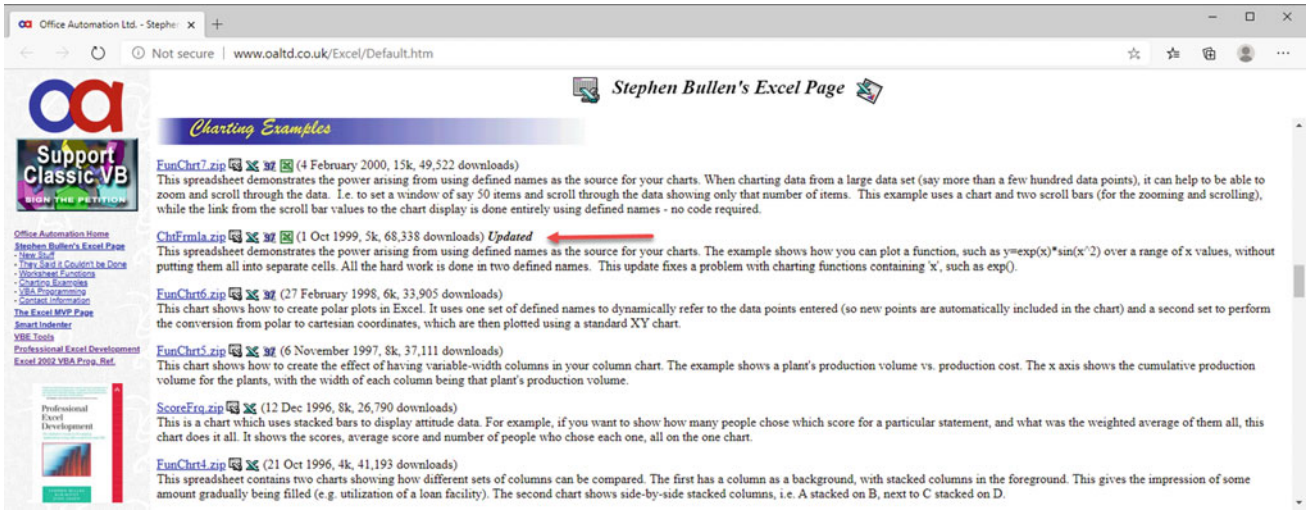


Type in cell c3 " = linear". The result is the Excel Name *linear* returns results of the formula in cell a1.

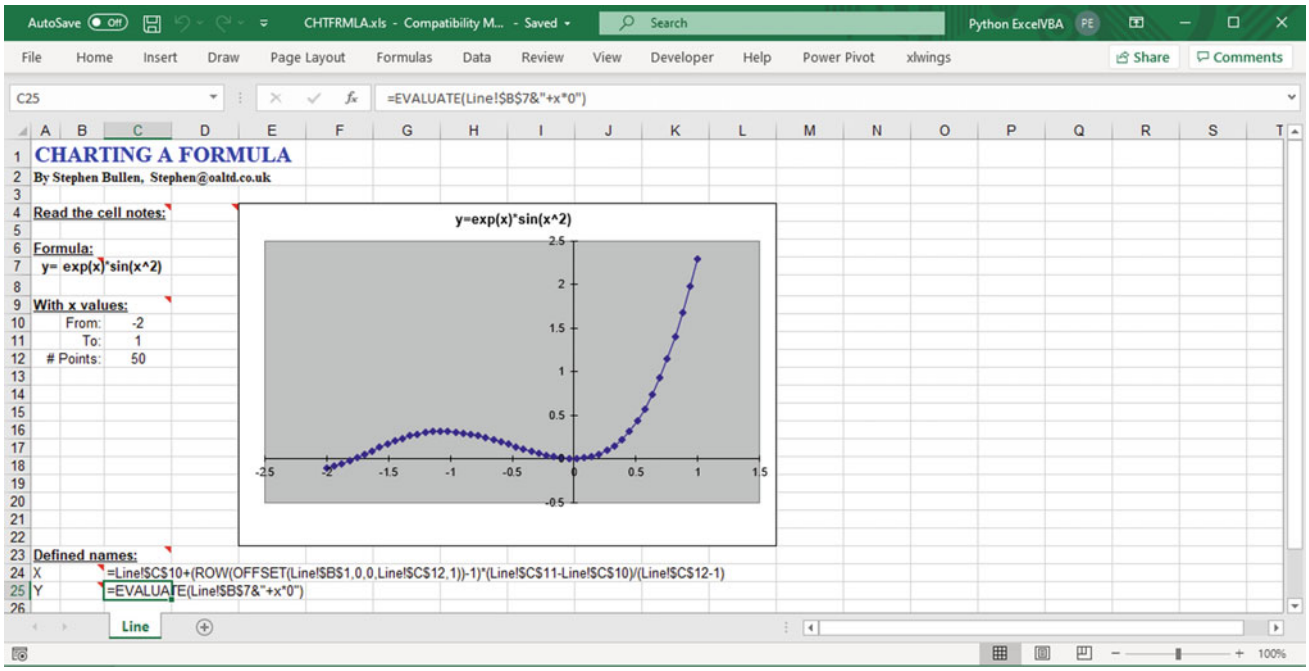


6.7 Stephen Bullen's Charting Method

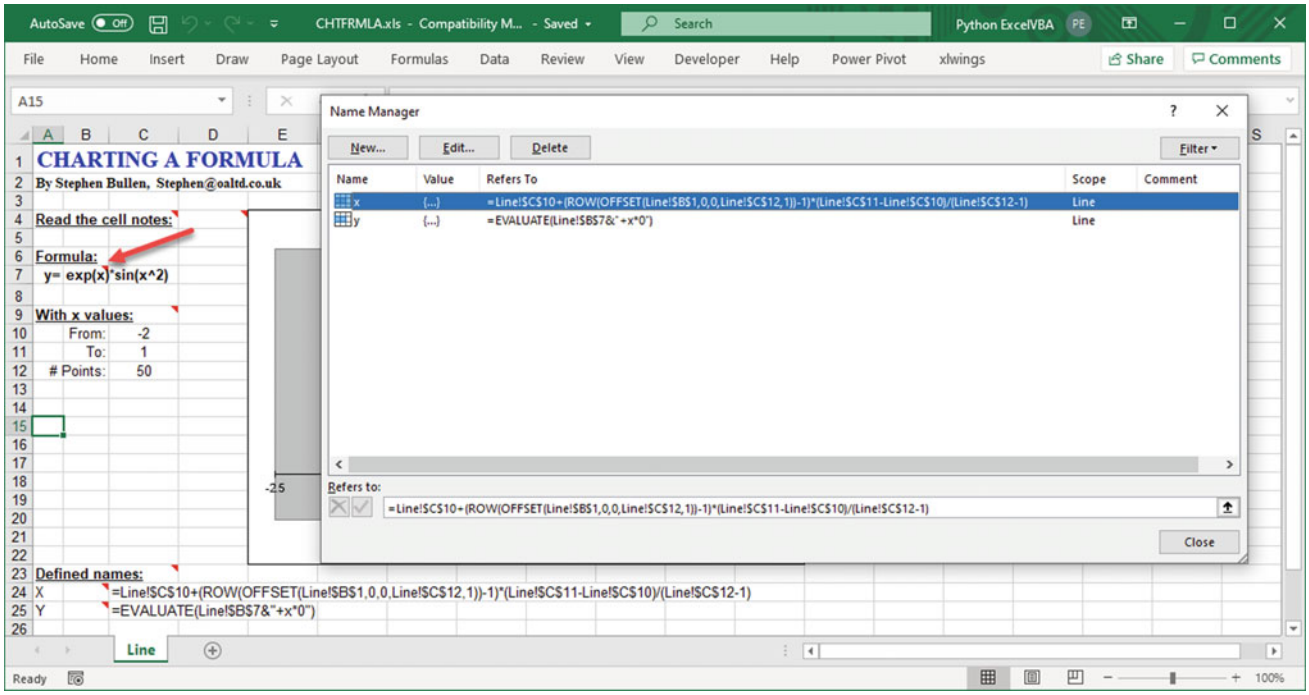
We used Excel's VBA programming language to chart the binomial and Poisson distributions. In this section, we will show Stephen Bullen's charting method. His charting method requires no VBA programming. Stephen Bullen accomplished this task by using Excel 4.0 macro functions in Excel Names. Stephen Bullen is one of the world's leading Microsoft Excel experts. Stephen Bullen's website is http://www.oaltd.co.uk/Excel/Default.htm.



Use the link http://www.oaltd.co.uk/DLCount/DLCount.asp?file=ChtFrmla.zip to get the file *ChtFrmla.*zip. Inside the zip file is the file *chtfrma.xls* that demonstrates Stephen Bullen's charting method. We will use this file to chart the Binomial and Poisson distribution. The *chtfrmla.xls* file is shown below:

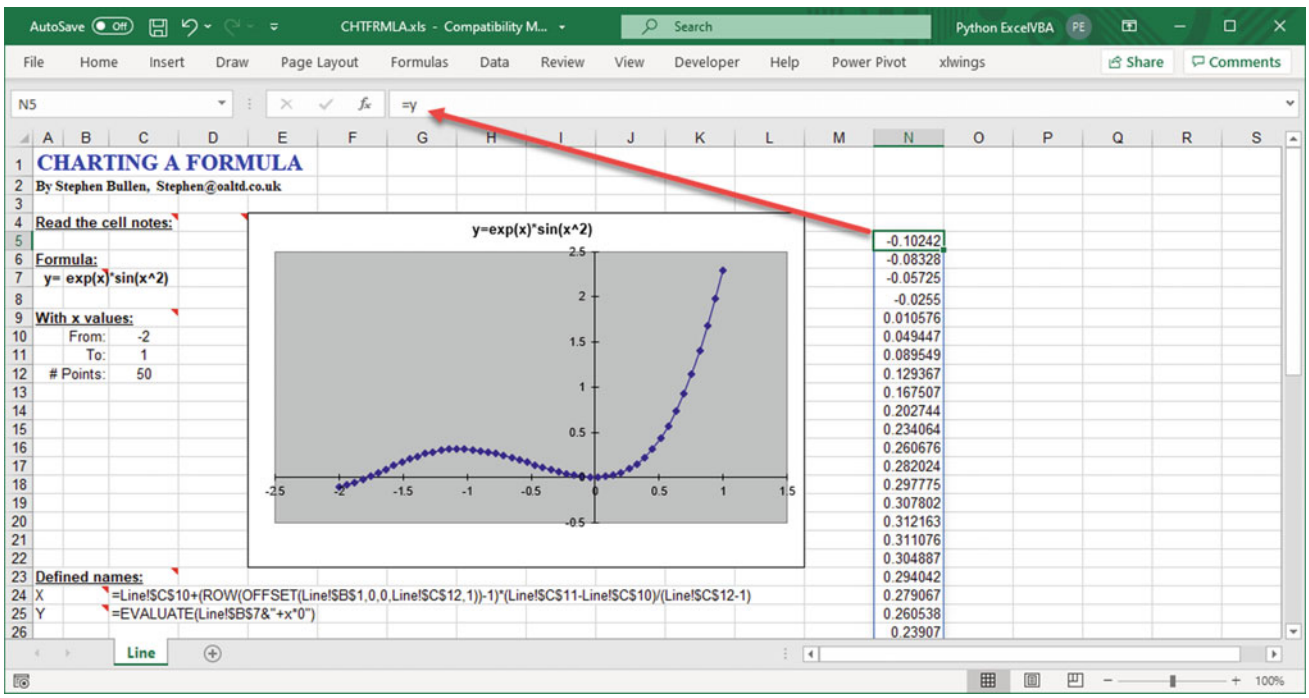


Entering a formula in cell b7 will update the chart. The key feature in making the charting work is the *x* and *y* Excel Names.

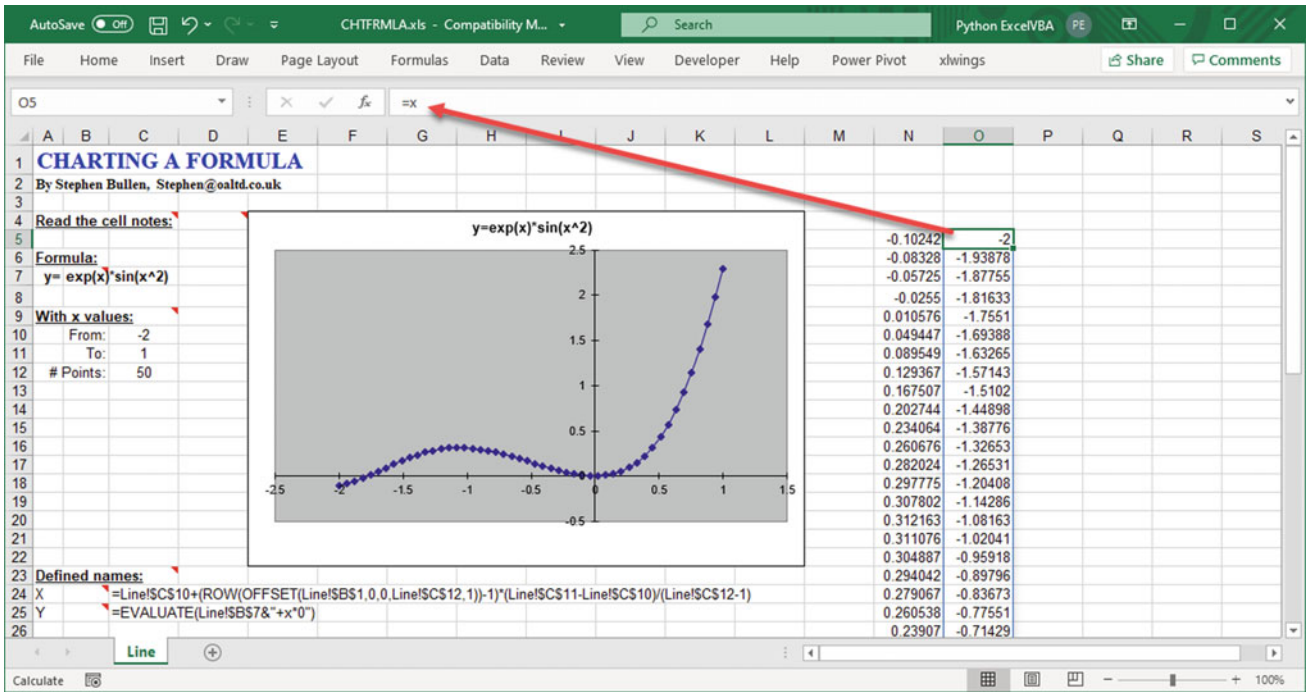


The Excel Name *y* uses the Excel 4.0 macro function *Evaluate,* which evaluates the text formula in cell b7. The Excel Name *x* generates the sequential data series for the formula in cell b7.

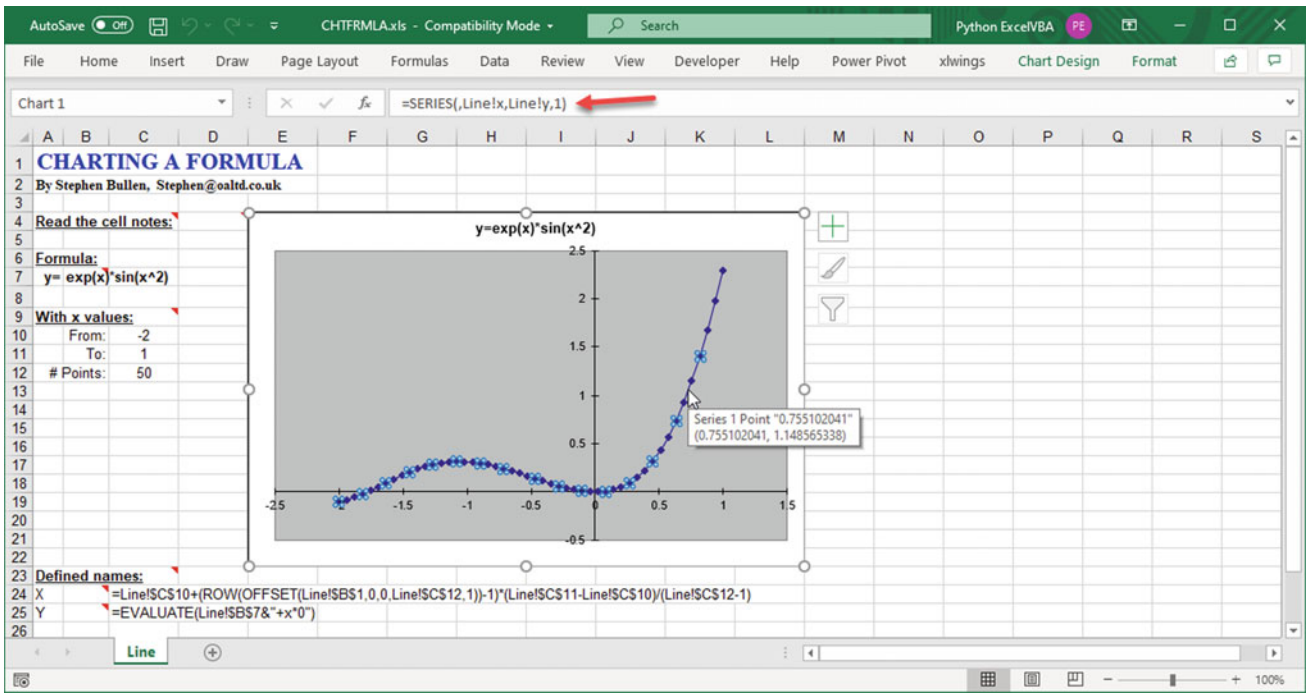
The Excel Name *y* evaluates to an array of numbers. This evaluation is demonstrated below.



The Excel Name *x* evaluates to an array of numbers; This is shown below.

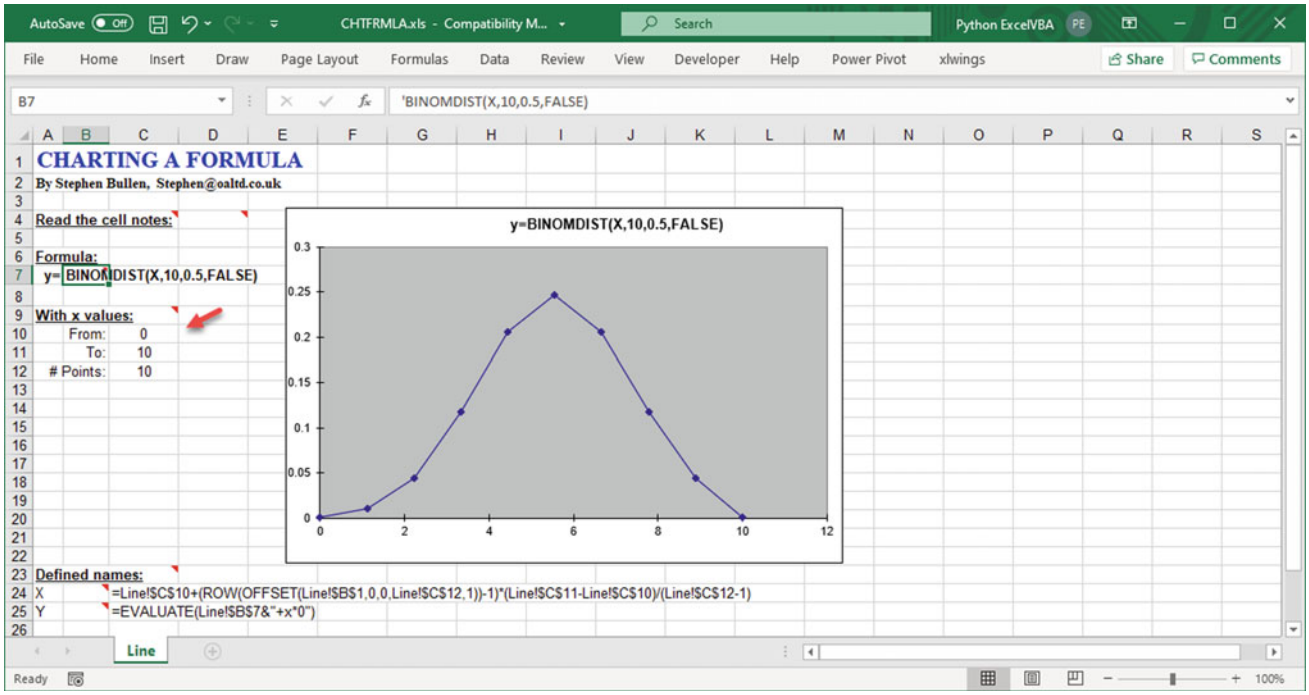


Clicking on the line in the graph demonstrates that the chart's data source is the Excel Name *x* and Excel Name *y*.

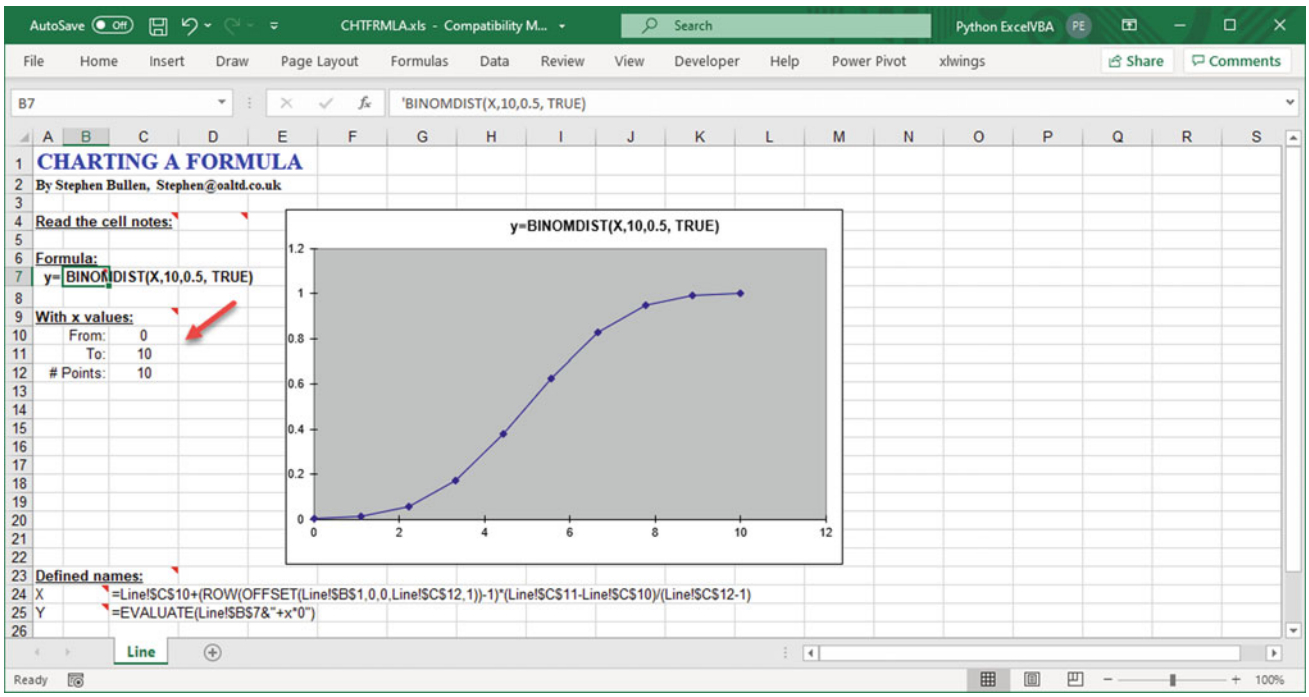


6.7.1 Binomial Distribution

Entering the binomial function 'BINOMDIST(X,10,0.5, FALSE) in cell b7 and updating the "from," "to," and "points" for the x values results in the following binomial PDF graph.

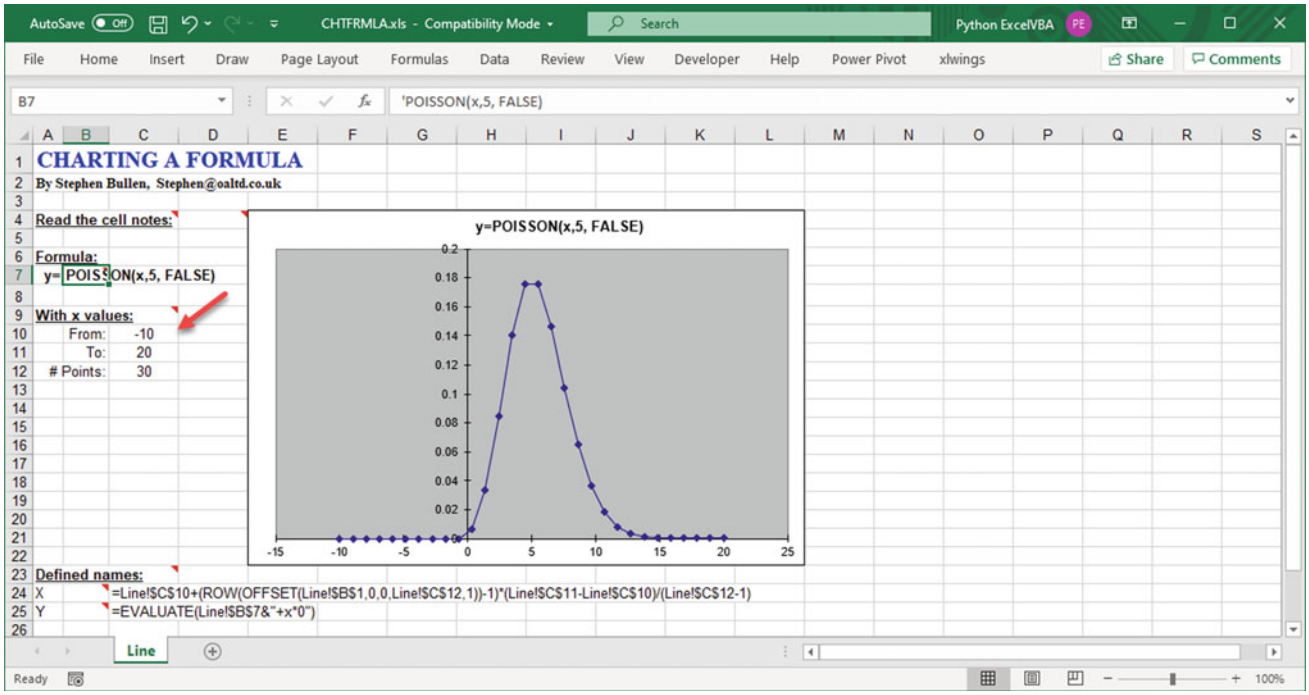


Entering the binomial function 'BINOMDIST(X,10,0.5, TRUE) in cell b7 and updating the "from," "to," and "points" for the x values results in the following binomial CDF graph.

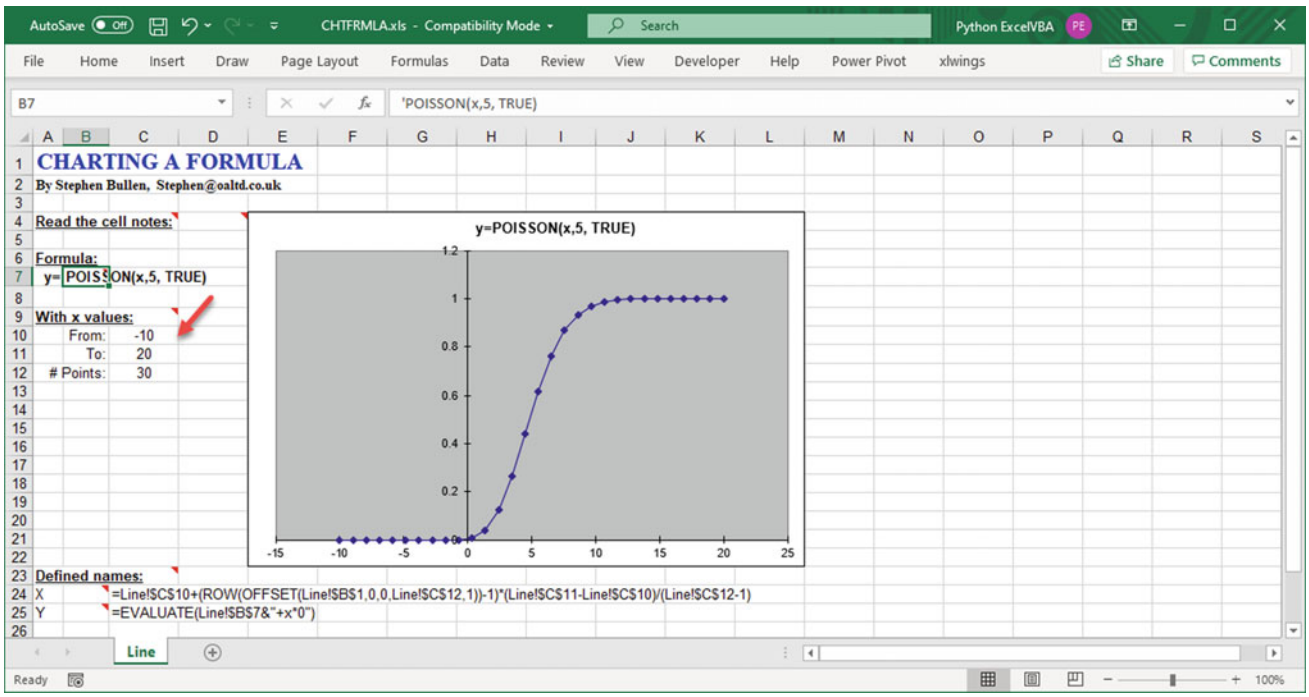


6.7.2 Poisson Distribution

Entering the Poisson function 'POISSON(x,5, FALSE) in cell b7 and updating the "from," "to," and "points" for the x values results in the following Poisson PDF graph:



Entering the Poisson function 'POISSON(x,5, TRUE) in cell b7 and updating the "from," "to," and "points" for the x values results in the following Poisson PDF graph:



6.8 Summary

In this chapter, we saw that a *discrete random variable* is a variable that has a countable number of outcomes. We also saw that a *probability distribution* is a list of all the probabilities for the outcomes of a variable. We study the binomial, Poisson distributions. We saw that as the probability of success increases for a binomial distribution, the more the binomial distribution will shift to the right. For the Poisson distribution, we saw that as the number of occurrences increases, the more the Poisson shift to the right. The Poisson also resembles more and more the normal distribution as the number of occurrences increases. The normal distribution is a very important distribution, and we will talk about it a lot more in the next couple of chapters.

This chapter used Excel, Python, and R to create the binomial and Poisson distribution.

This chapter also demonstrated Stephen Bullen's very sophisticated Excel function method for charting these two distributions.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Check for updates

Normal and Lognormal Distributions

7

7.1 Introduction

In the last chapter, we examined the probability distribution of discrete random variables. In this chapter, we will look at the probability distribution of *continuous random variables*.

Unlike discrete random variables, which are limited to a finite or countable number of distinct (integer) values, values of continuous variables are not limited to being integers; theoretically, they are infinitely divisible. A continuous random variable may take on any value within an interval.

The following chapter is broken down as follows. Section 7.2 discusses uniform distribution, while Sect. 7.3 talks about normal distribution. Standard normal distribution is examined in Sect. 7.4, and lognormal distribution is discussed in Sect. 7.5. The normal quantile–quantile (QQ) plot in Excel is discussed in Sect. 7.6, and in Sect. 7.7 we discuss the normal quantile–quantile (QQ) plot in Python. Finally, in Sect. 7.8, we summarize the chapter.

7.2 Uniform Distribution

One interesting characteristic of continuous random variables is that the entire range of probabilities under a continuous random variable has a total area equal to one. This is most clearly shown in the uniform continuous distribution. The formula for the uniform distribution is

$$f(x) = \frac{1}{d-c}, c < x < d$$

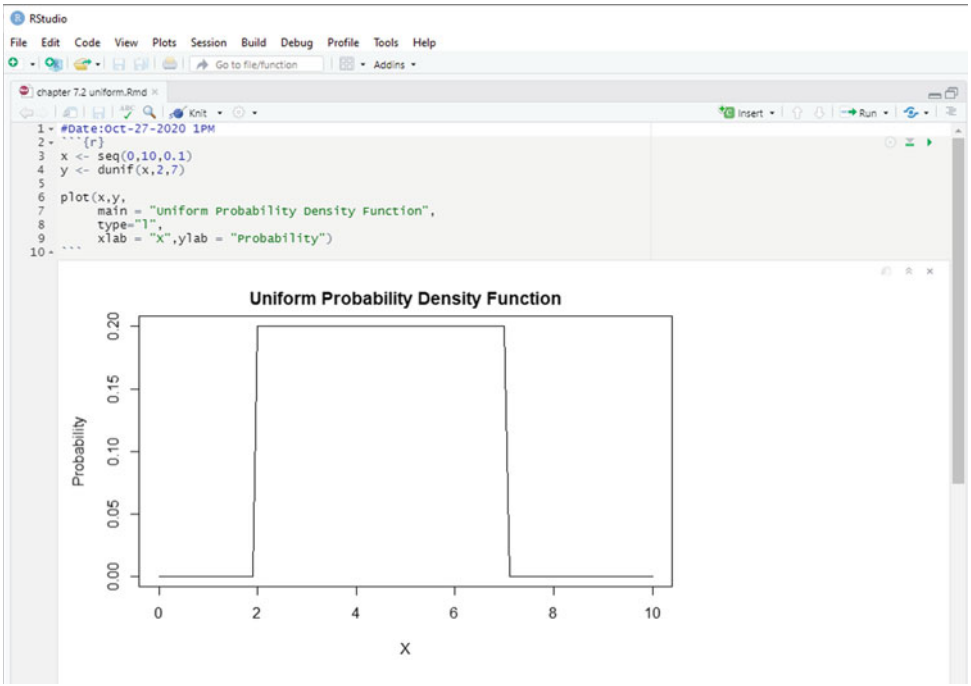
The c is the beginning point and the d is the ending point of the uniform distribution. This is a uniform distribution because any two points between points c and d will have the same probability.

7.2.1 Uniform Distribution in *R*

The *R* code that creates the uniform distribution graph is given below:

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_7

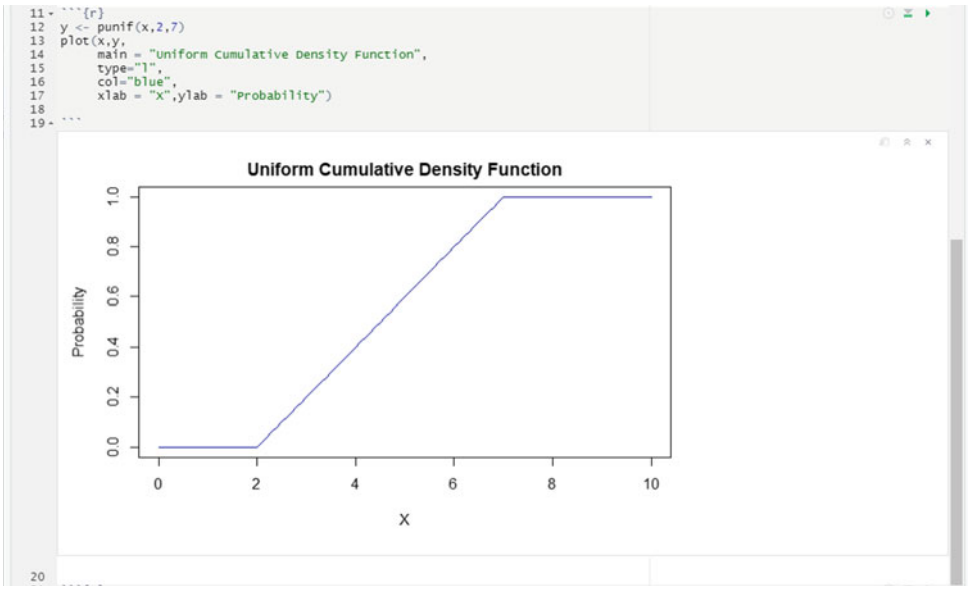


We can confirm that this plot has an area that is equal to 1 by seeing that the height is 0.2 and the width is 5. If we multiply these two numbers together, we get 1.

From the above graph, we can see the probability of any point of the uniform distribution. For example, the probability is 0.2 when the uniform distribution is 4. The probability is also 0.2 when the uniform distribution is 6. This graph and the function that defines this graph are commonly known as the probability density function (PDF) of the uniform distribution or the uniform probability density function.

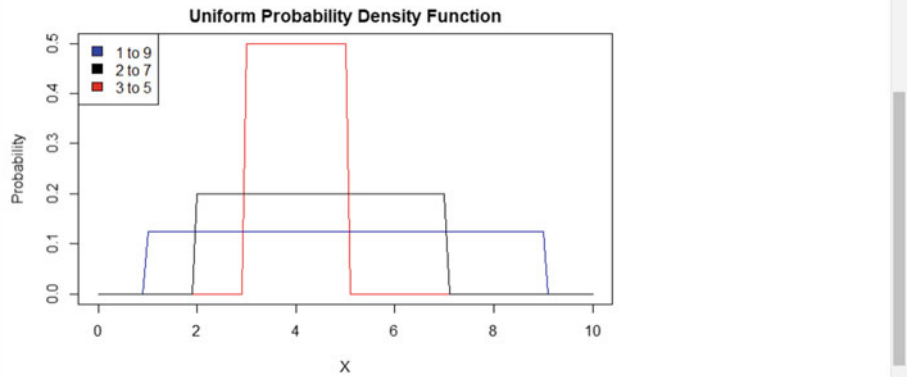
Another important concept is to know the total probability of a distribution from left to right to a certain point in the distribution. The graph or function that answers this question is called the cumulative density function (CDF). For example, the CDF when the uniform distribution is equal to 4 is $(4 - 2) * 0.2 = 0.4$.

The cumulative uniform distribution graph using *R* is created below:



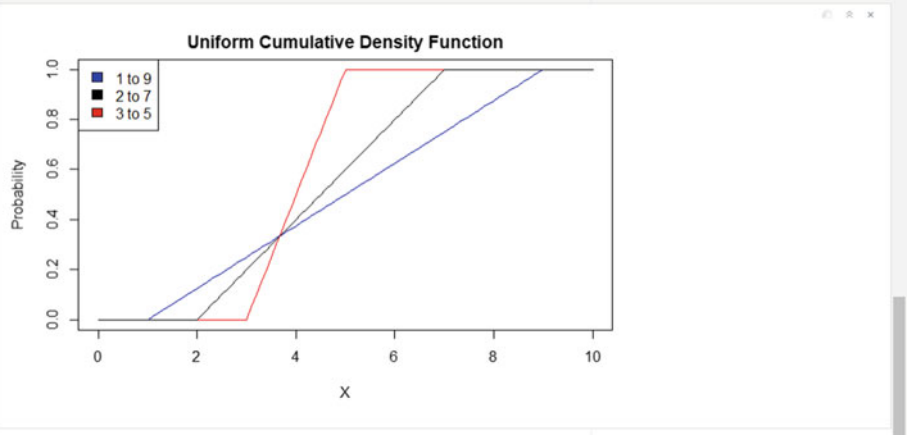
The *c* and *d* points that influence the uniform distribution are given below. We will do this by graphing three distributions together. One distribution will be longer than the above distribution, and one distribution will be shorter than the above distribution.

```r
x <- seq(0,10,0.1)
y <- dunif(x,3,5)

plot(x,y,
     main = "Uniform Probability Density Function",
     type="l",
      col="red",
     xlab = "X",ylab = "Probability")

y <- dunif(x,1,9)
lines(x,y,col="blue")

y <- dunif(x,2,7)
lines(x,y,col="black")

legend("topleft",c("1 to 9","2 to 7","3 to 5"),fill = c("blue","black","red"))
```



If we do the geometry calculation on all the graphs, we will find that all the graphs have an area equal to 1. The cumulative uniform density functions for the above probability density functions are shown below:

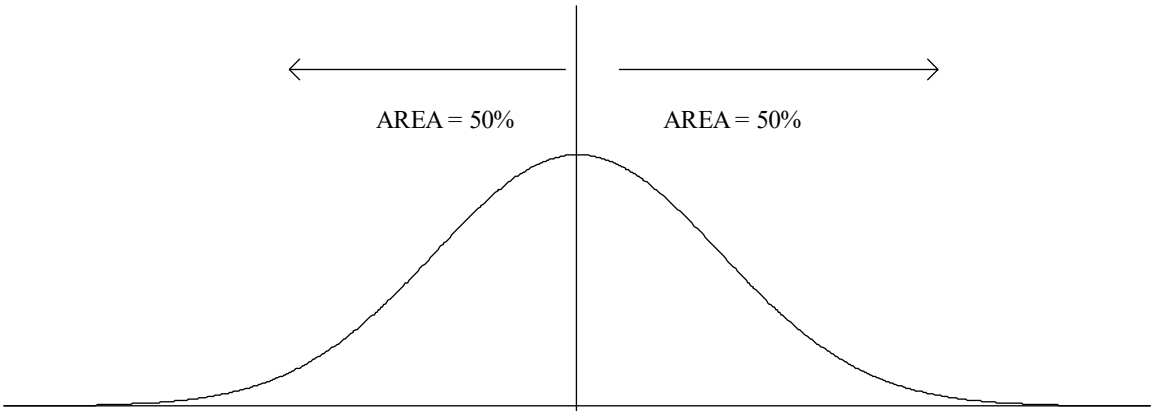
```r
y <- punif(x,3,5)
plot(x,y,
     main = "Uniform Cumulative Density Function",
     type="l",
     col="red",
     xlab = "X",ylab = "Probability")
y <- punif(x,1,9)
lines(x,y,col="blue")
y <- punif(x,2,7)
lines(x,y,col="black")
legend("topleft",c("1 to 9","2 to 7","3 to 5"),fill = c("blue","black","red"))
```



7.3 Normal Distribution

Continuous variables have many shapes, but one of the most important is the normal random variable.

The probability density function of the normal distribution looks like the following:



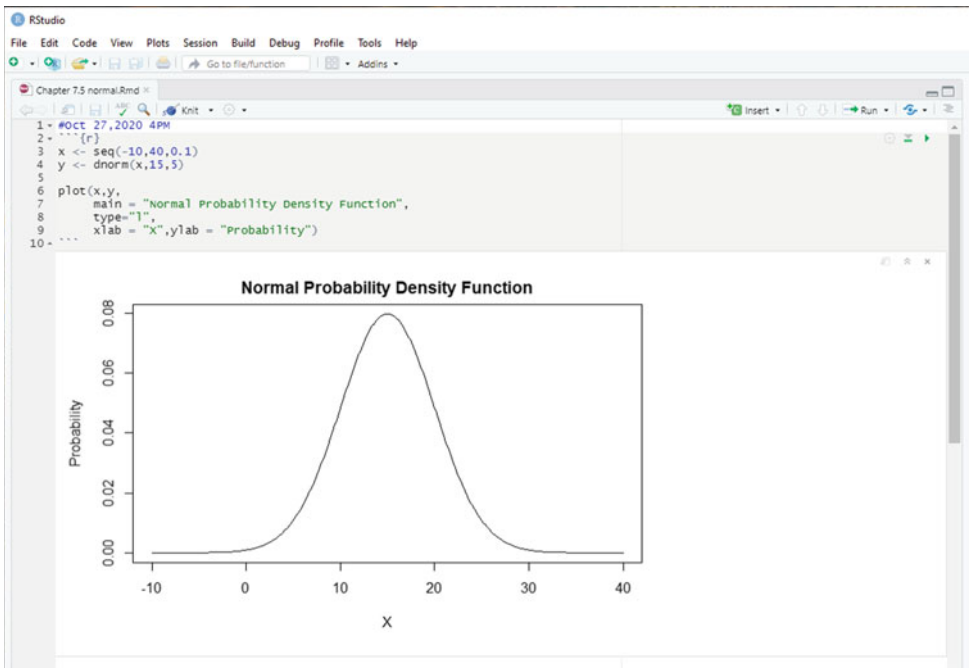
The formula for this function is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, -\infty < x < \infty$$

We should keep in mind some of the characteristics of the normal distribution. First, the distribution goes indefinitely in both directions. The area under the normal distribution is equal to 1. The normal distribution has a bell shape where the greatest probability density is at the midpoint. The normal distribution is symmetrical at its midpoint. That is, the probability to the left of the normal distribution is equal to 0.5, and the probability to the right is also equal to 0.5. The shape of the normal distribution is influenced by the standard deviation, σ, and the mean, μ.

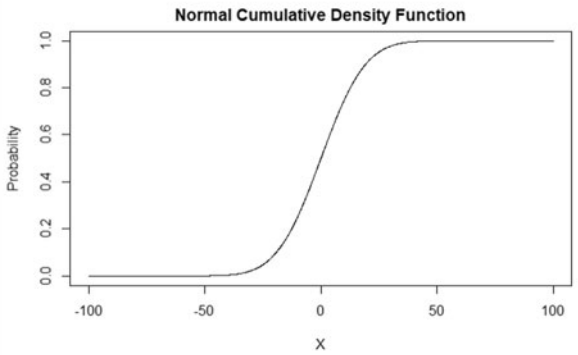
7.3.1 Normal Distribution in *R*

The *R* commands for creating a normal distribution with a mean of 15 and a standard deviation of 5 are as follows:



We will now calculate and plot the cumulative density function of the normal distribution to demonstrate that the area or total probability of a normal distribution is equal to 1.

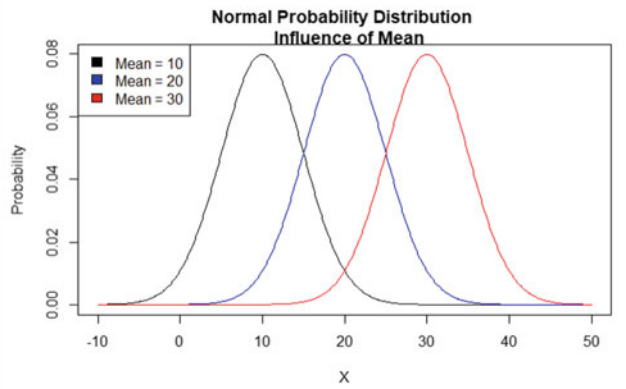
```r
x <- seq(-100,100,0.1)
y <- pnorm(x,0,15)
plot(x,y,
     main = "Normal Cumulative Density Function",
     type="l",
     col="black",
     xlab = "X",ylab = "Probability")
```



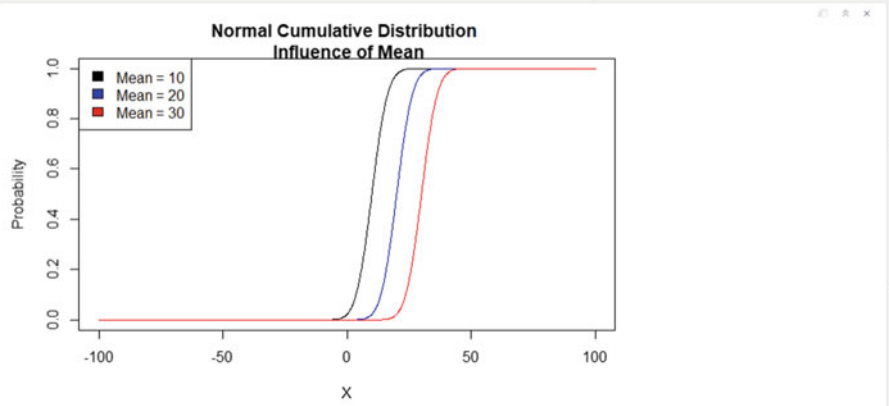
The normal distribution is influenced by its two parameters. The first parameter, the mean, anchors the location of the normal distribution. The second parameter, the standard deviation, determines the peakness of the normal distribution.

The influence of the mean is shown below. Notice that the normal distribution shifts to the right as the mean increases.

```r
#demonstrates how the mean anchors the normal distribution
x <- seq(-10,50,0.1)
y <- dnorm(x,10,5)

plot(x,y,
     main = "Normal Probability Distribution \n  Influence of Mean",
     type="l",
     xlab = "X",ylab = "Probability")

y <- dnorm(x,20,5)
lines(x,y,col="blue")

y <- dnorm(x,30,5)
lines(x,y,col="red")

legend("topleft",c("Mean = 10","Mean = 20","Mean = 30"),fill = c("black","blue","red"))
```

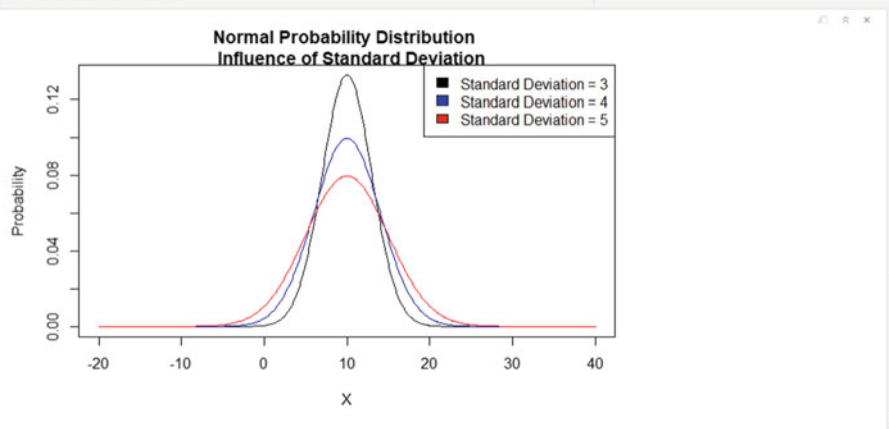


```r
#demonstrates the how the mean influence the normal cumulative distribution
x <- seq(-100,100,0.1)
y <- pnorm(x,10,5)
plot(x,y,
     main = "Normal Cumulative Distribution \n Influence of Mean",
     type="l",
     col="black",
     xlab = "X",ylab = "Probability")

y <- pnorm(x,20,5)
lines(x,y,col="blue")

y <- pnorm(x,30,5)
lines(x,y,col="red")

legend("topleft",c("Mean = 10","Mean = 20","Mean = 30"),fill = c("black","blue","red"))
```

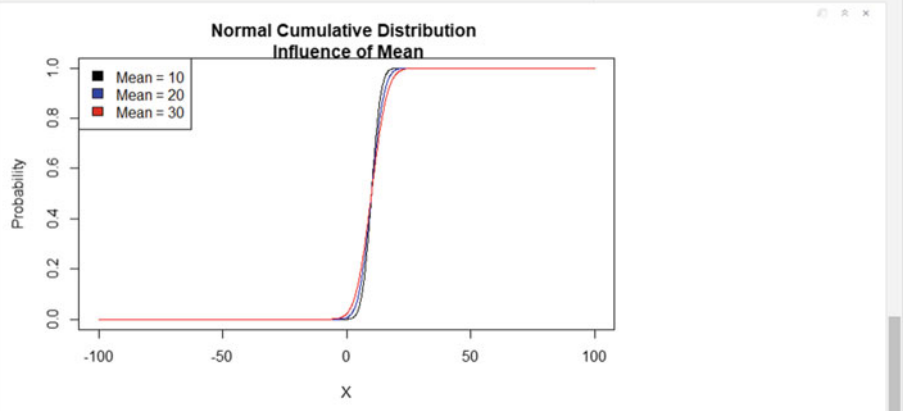


The influence of the standard deviation is shown below. Notice that the normal distribution becomes more peak as the standard deviation decreases.

```r
#demonstrates how the mean anchors the normal distribution
x <- seq(-20,40,0.1)
y <- dnorm(x,10,3)

plot(x,y,
     main = "Normal Probability Distribution \n  Influence of Standard Deviation",
     type="l",
     xlab = "X",ylab = "Probability")

y <- dnorm(x,10,4)
lines(x,y,col="blue")

y <- dnorm(x,10,5)
lines(x,y,col="red")

legend("topright",c("Standard Deviation = 3","Standard Deviation = 4","Standard Deviation = 5"),fill =
c("black","blue","red"))
```

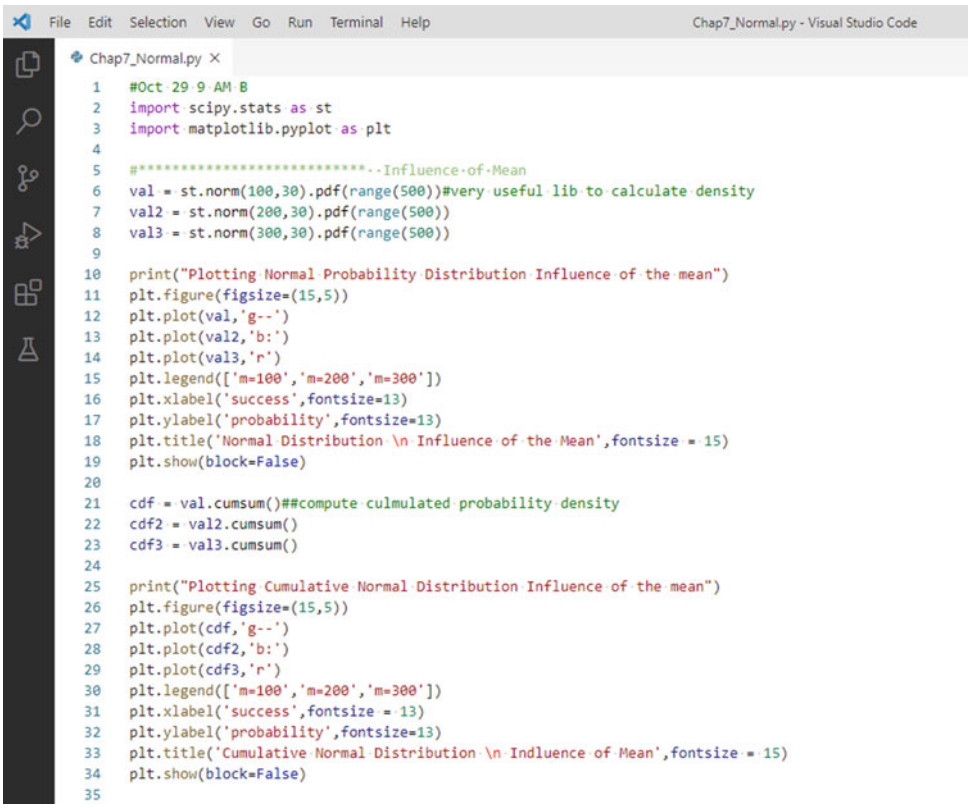


```r
#demonstrates the how the mean influence the normal cumulative distribution
x <- seq(-100,100,0.1)
y <- pnorm(x,10,3)
plot(x,y,
     main = "Normal Cumulative Distribution \n Influence of Mean",
     type="l",
     col="black",
     xlab = "X",ylab = "Probability")

y <- pnorm(x,10,4)
lines(x,y,col="blue")

y <- pnorm(x,10,5)
lines(x,y,col="red")

legend("topleft",c("Mean = 10","Mean = 20","Mean = 30"),fill = c("black","blue","red"))
```



7.3.2 Normal Distribution in *Python*

The *Python* commands that demonstrate how the mean influences the normal distribution are given below:




```python
#Oct 29 9 AM B
import scipy.stats as st
import matplotlib.pyplot as plt

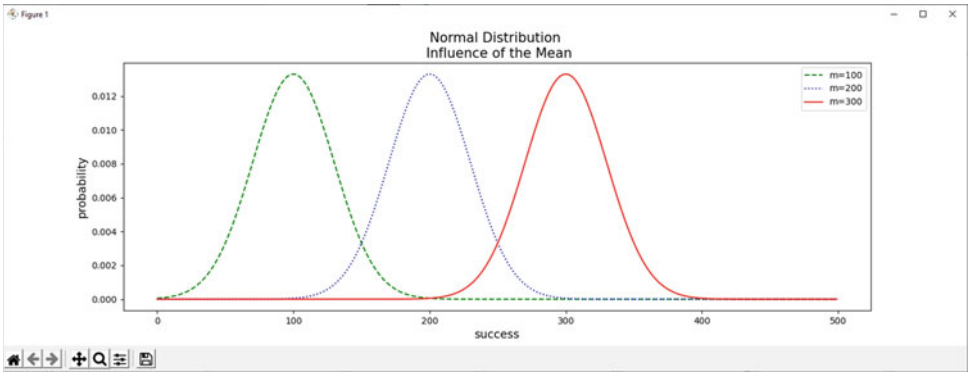
#**************************Influence of Mean
val = st.norm(100,30).pdf(range(500))#very useful lib to calculate density
val2 = st.norm(200,30).pdf(range(500))
val3 = st.norm(300,30).pdf(range(500))

print("Plotting Normal Probability Distribution Influence of the mean")
plt.figure(figsize=(15,5))
plt.plot(val,'g--')
plt.plot(val2,'b:')
plt.plot(val3,'r')
plt.legend(['m=100','m=200','m=300'])
plt.xlabel('success',fontsize=13)
plt.ylabel('probability',fontsize=13)
plt.title('Normal Distribution \n Influence of the Mean',fontsize = 15)
plt.show(block=False)

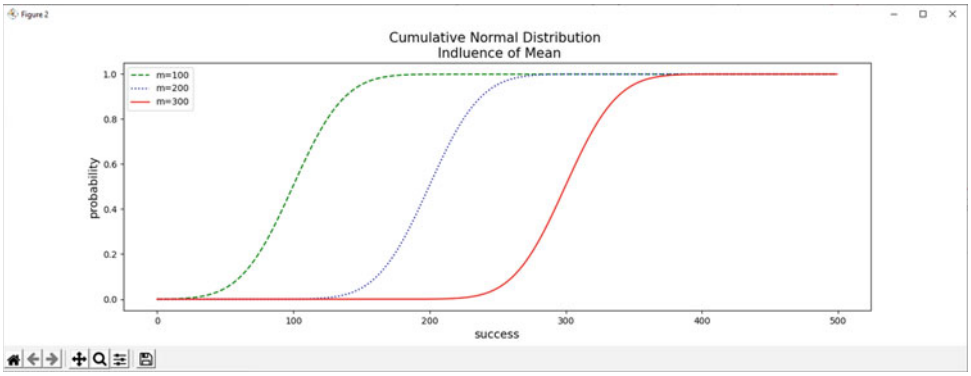
cdf = val.cumsum()##compute culmulated probability density
cdf2 = val2.cumsum()
cdf3 = val3.cumsum()

print("Plotting Cumulative Normal Distribution Influence of the mean")
plt.figure(figsize=(15,5))
plt.plot(cdf,'g--')
plt.plot(cdf2,'b:')
plt.plot(cdf3,'r')
plt.legend(['m=100','m=200','m=300'])
plt.xlabel('success',fontsize = 13)
plt.ylabel('probability',fontsize=13)
plt.title('Cumulative Normal Distribution \n Indluence of Mean',fontsize = 15)
plt.show(block=False)
```

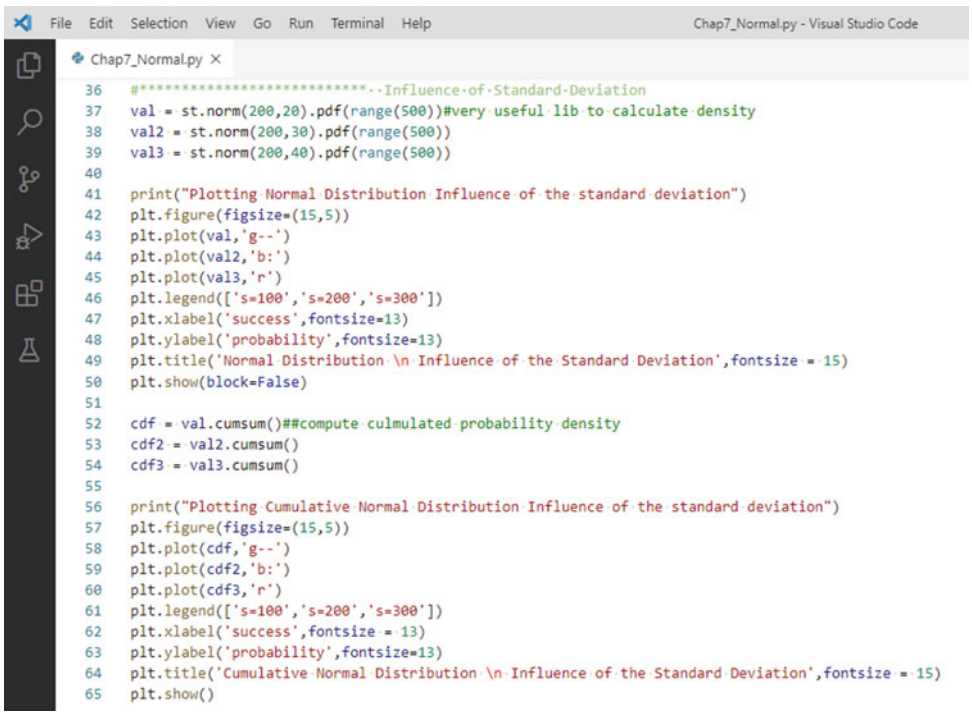
The graph produced by the *Python* command is given below. Note that the normal distribution shifts to the right as the mean increases.



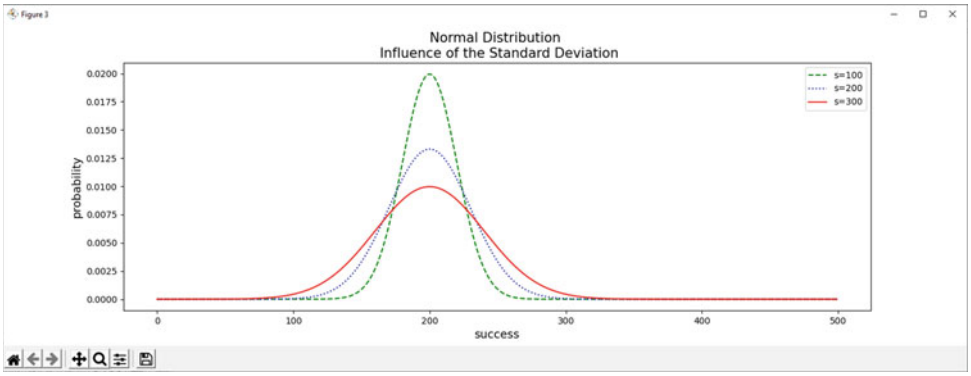
The corresponding cumulative normal distribution graph is given below:



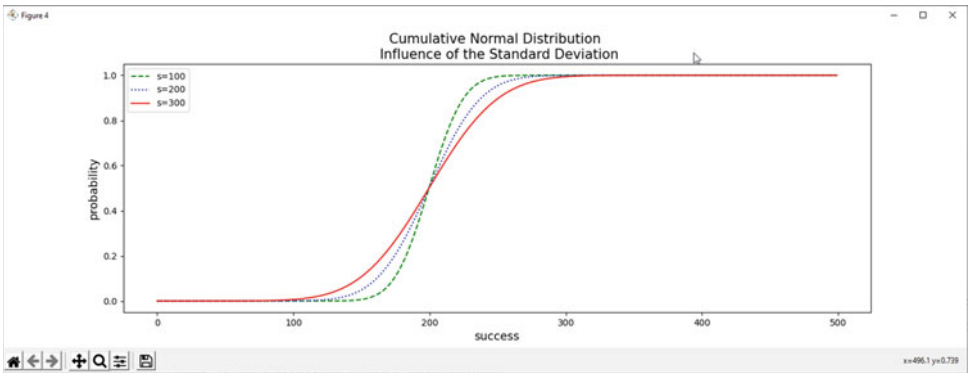
The *Python* commands that demonstrate how the standard deviation influences the normal distribution are given below:



The graph produced by the *Python* command is given below. Note that the normal distribution becomes more peak as the standard deviation decreases.



The corresponding cumulative normal distribution graph is given below:

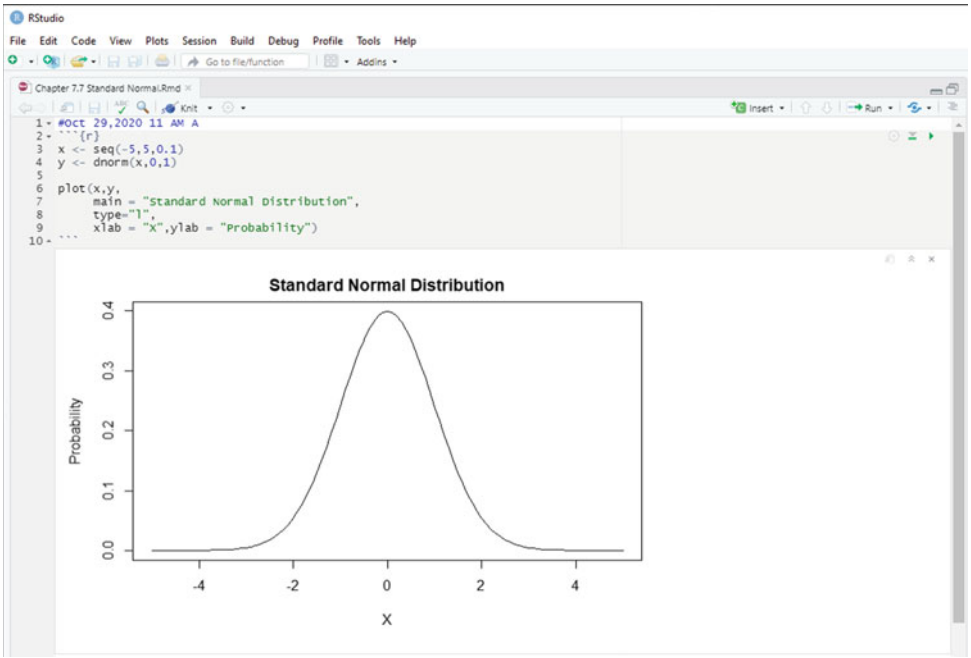


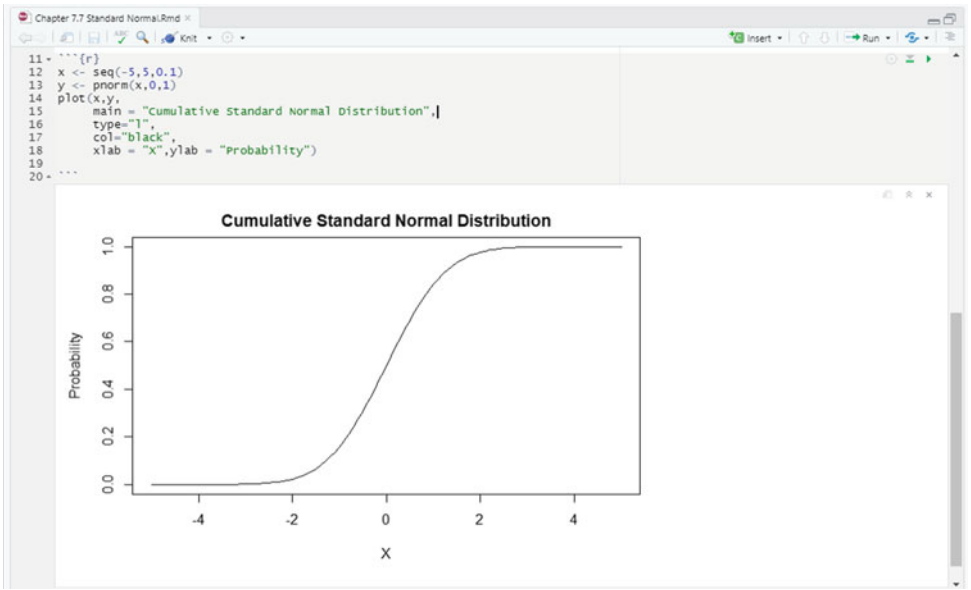
7.4 Standard Normal Distribution

We will find that the normal distribution is one of the most widely used continuous distributions in statistics. In fact, there is an infinite number of normal curves in statistics. Statistical analysis transforms normal curves to a *standard normal distribution* to analyze normal curves more easily. The Z-score is used to do the transformation. The standard normal distribution is a normal curve with a mean of 0 and a standard deviation of 1.

7.4.1 Standard Normal Distribution in *R*

The *R* commands to create the standard normal distribution, which has a mean of 0 and a standard deviation of 1 are given below:

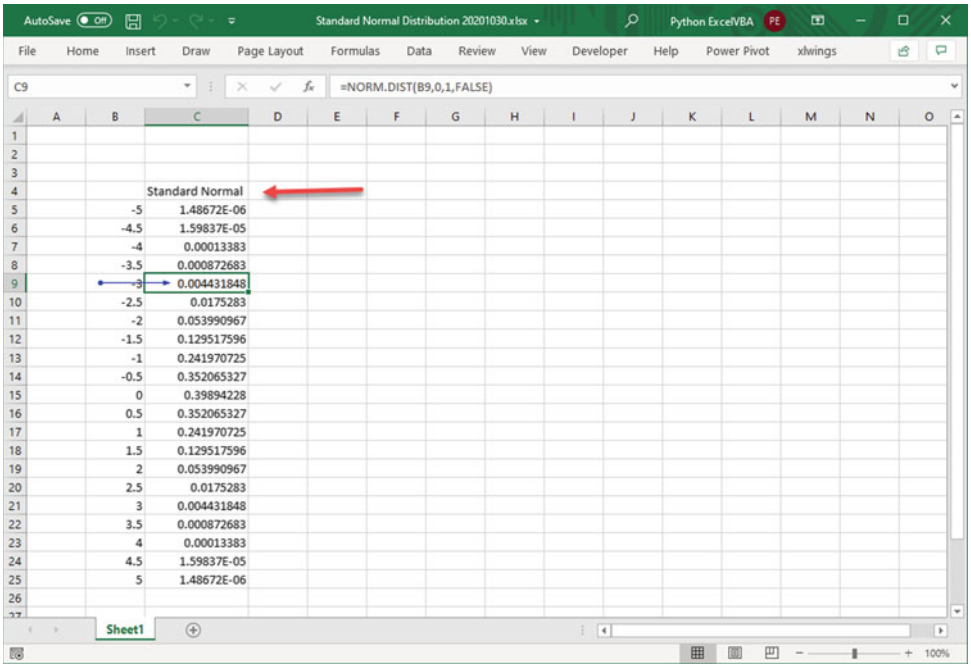




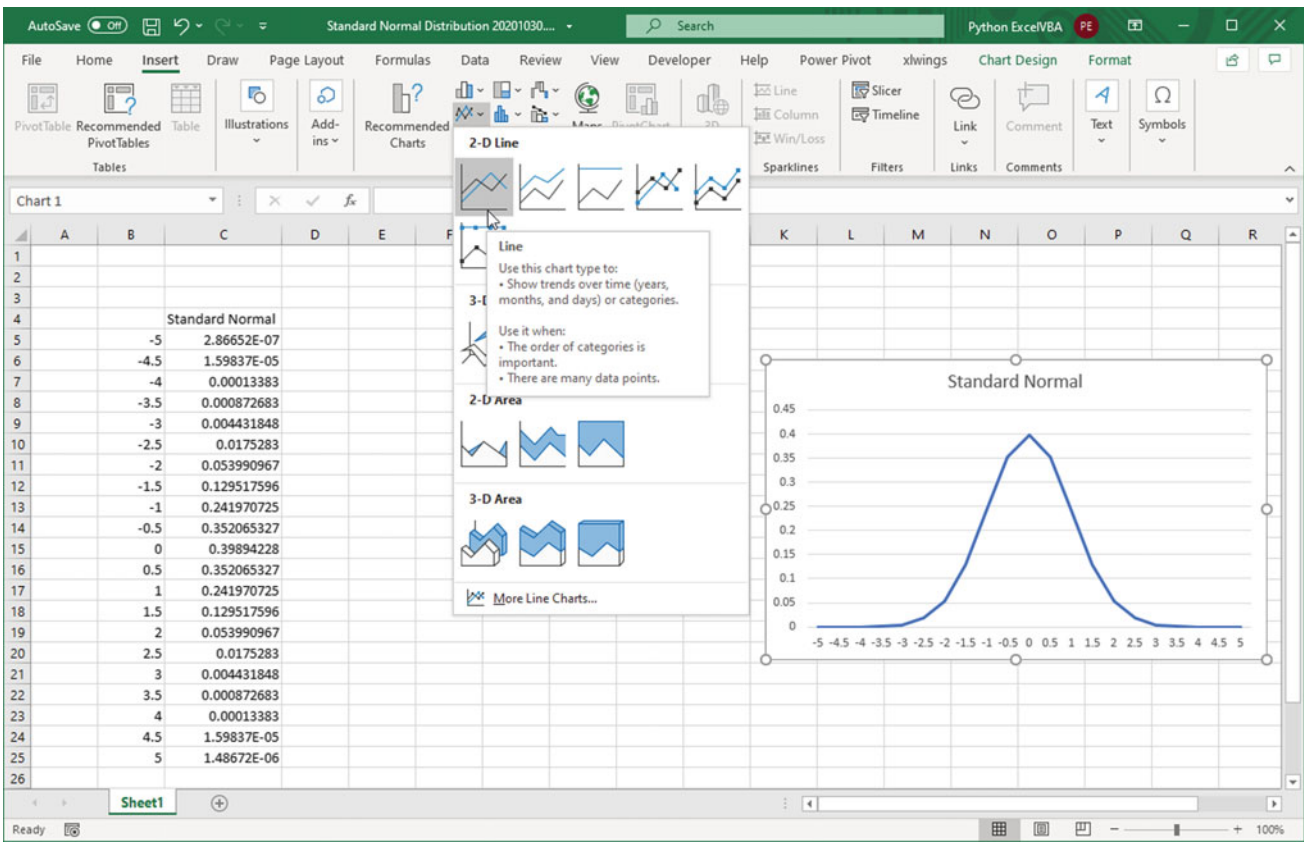
7.4.2 Standard Normal Distribution in Excel

The steps to create the standard normal distribution in Excel are given below:

(1) Enter the standard normal table, as shown below.
(2) The table below uses the Excel function *NORM.DIST*. Note the last parameter is FALSE.
(3) In cell c4, enter "Standard Normal."

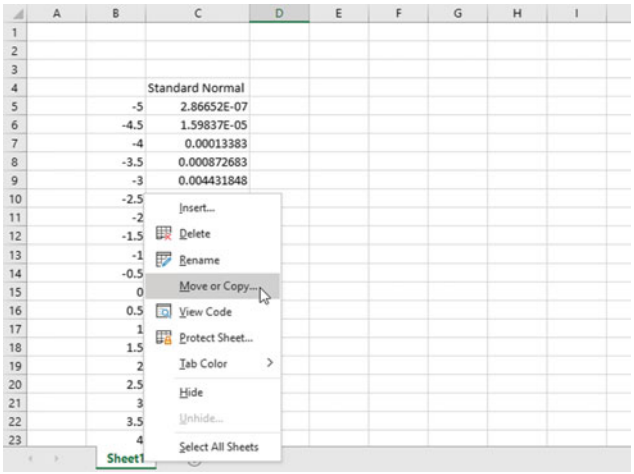


(4) With cell c4 selected, choose the 2-D line in the *Insert* page, this is shown below:

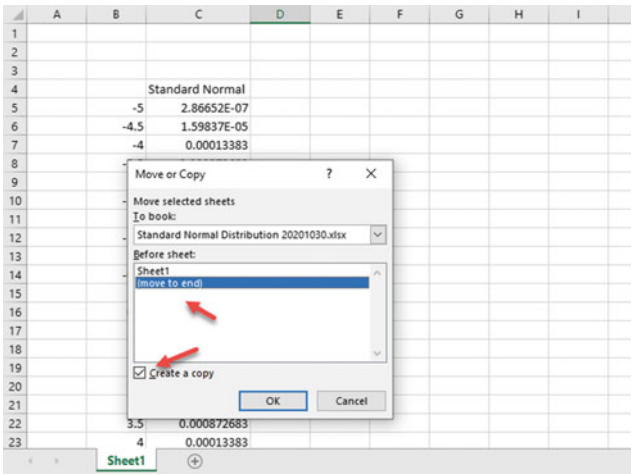


The steps to create the cumulative standard normal distribution in Excel are given below:

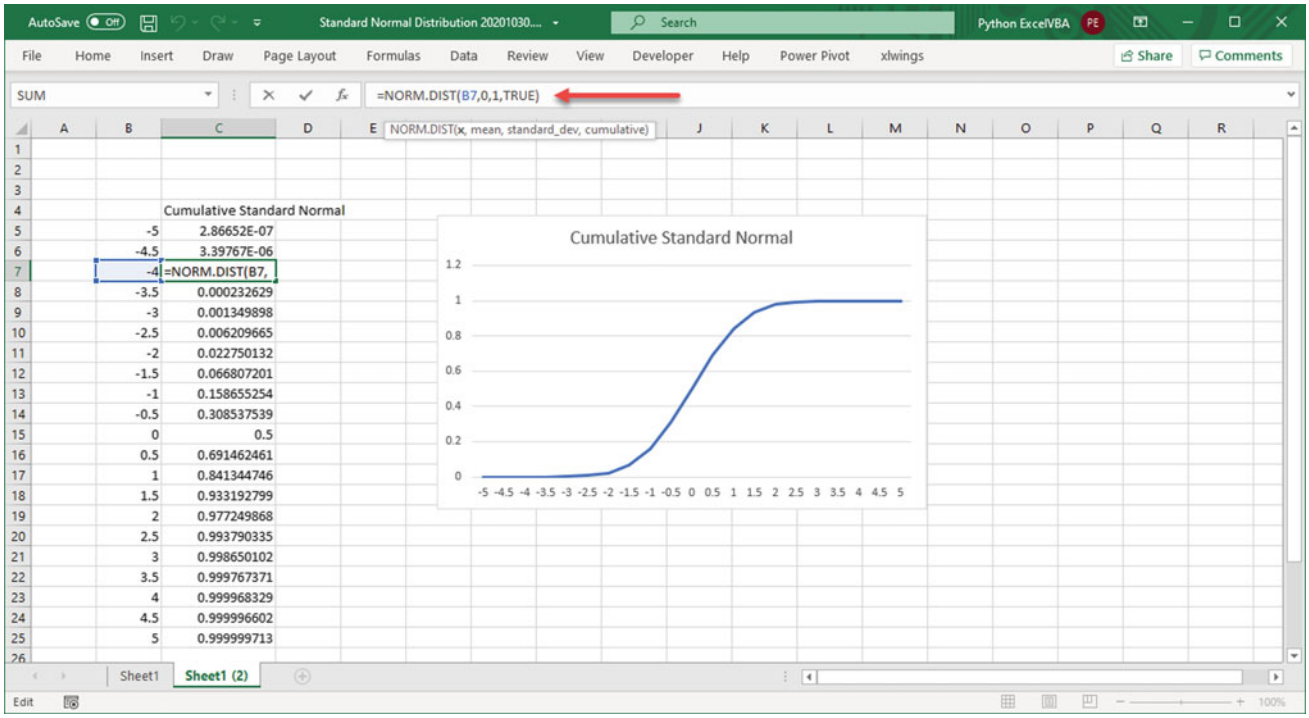
(1) Make a copy of the standard normal sheet by right-mouse clicking on the tab and choosing *Move or Copy*.



(2) Click on *Create a copy.*
(3) Select *(move to end).*



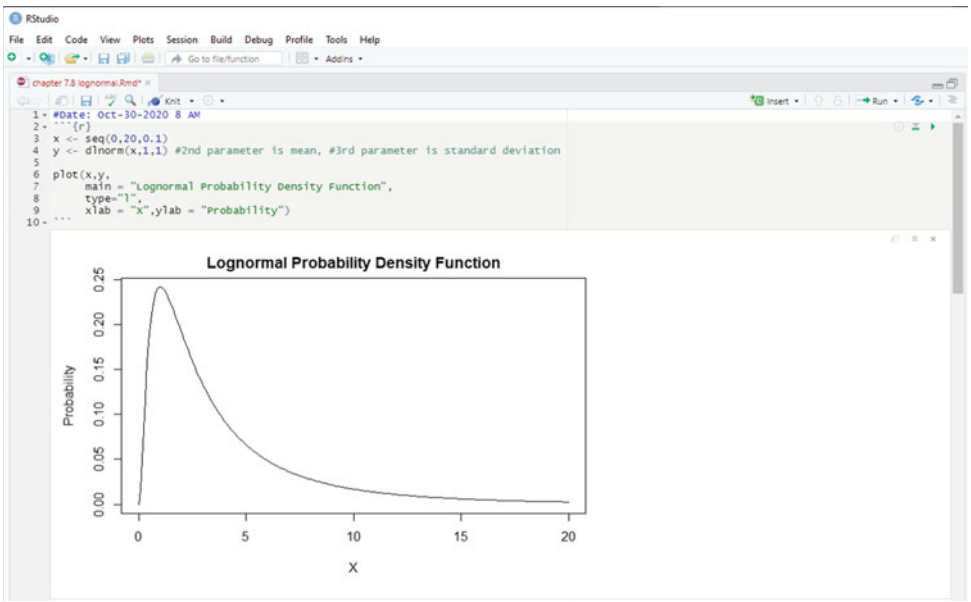
(4) Enter "Cumulative Standard Normal" in cell c4.
(5) In the table, change the last parameter of the *NORM.DIST* to TRUE.



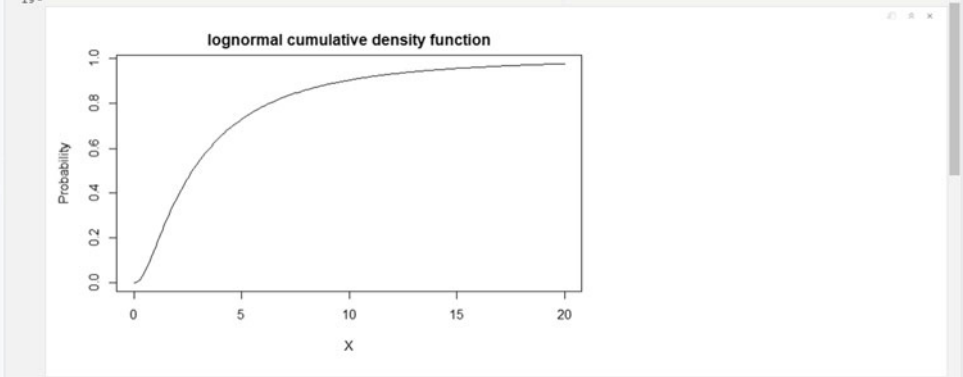
7.5 Lognormal Distribution

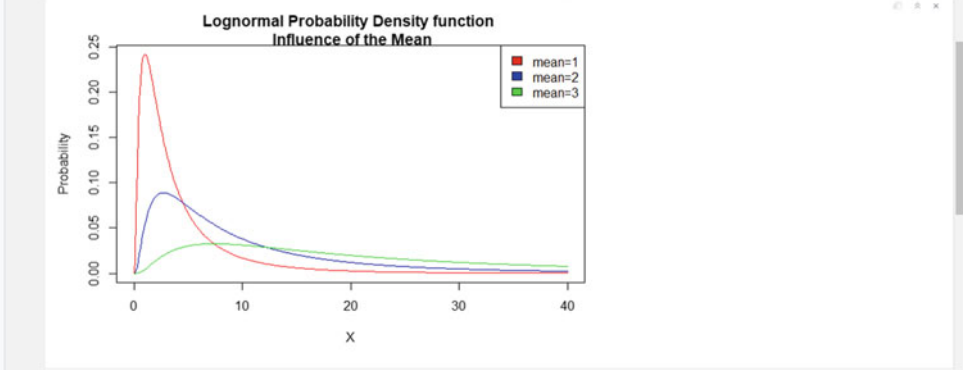
7.5.1 Lognormal Distribution in R

The *R* commands to produce a lognormal distribution graph are given below:



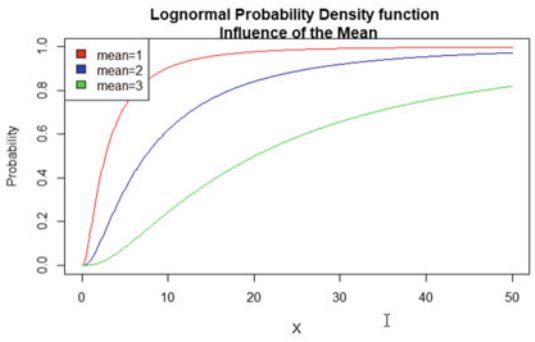
The *R* commands to produce a cumulative lognormal distribution graph are given below:

```
```{r}
x <- seq(0,20,0.1)
y <- plnorm(x,1,1) #2nd parameter is mean, #3rd parameter is standard deviation
plot(x,y,
     main = "lognormal cumulative density function",
     type="l",
     xlab = "X",ylab = "Probability")
```
```



We will now demonstrate how the mean and the standard deviation influence the lognormal distribution. The first thing that we will show is how the mean influences the lognormal distribution. We will use three lognormal distributions. One of them will have a mean of 1. One of them will have a mean of 2. One of them will have a mean of 3.

```
```{r}
x <- seq(0,40,0.1)
y <- dlnorm(x,1,1) #2nd parameter is mean, #3rd parameter is standard deviation
plot(x,y,
     main = "Lognormal Probability Density function \n Influence of the Mean",
     type="l",
     col="red",
     xlab = "X",ylab = "Probability")
y <- dlnorm(x,2,1)
lines(x,y,col="blue")
y <- dlnorm(x,3,1)
lines(x,y,col="green")
legend("topright",c("mean=1","mean=2","mean=3"),fill = c("red","blue","green"))
```
```



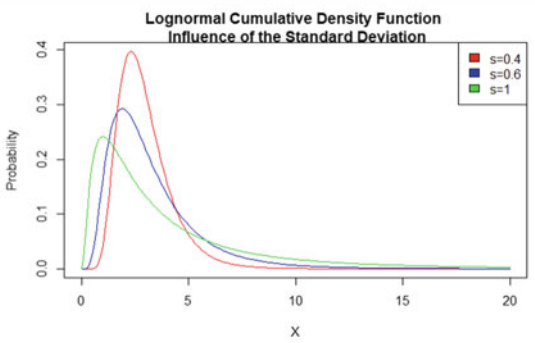
From the above exercise, we can see that increasing the mean flattens the lognormal distribution.
The corresponding cumulative lognormal density function for the above lognormal density function is given below:

```r
x <- seq(0,50,0.1)
y <- plnorm(x,1,1) #2nd parameter is mean, #3rd parameter is standard deviation
plot(x,y,
     main = "Lognormal Probability Density function \n Influence of the Mean",
     type="l",
     col="red",
     xlab = "X",ylab = "Probability")
y <- plnorm(x,2,1)
lines(x,y,col="blue")
y <- plnorm(x,3,1)
lines(x,y,col="green")
legend("topleft",c("mean=1","mean=2","mean=3"),fill = c("red","blue","green"))
```



We will now show how the standard deviation influences the lognormal distribution.

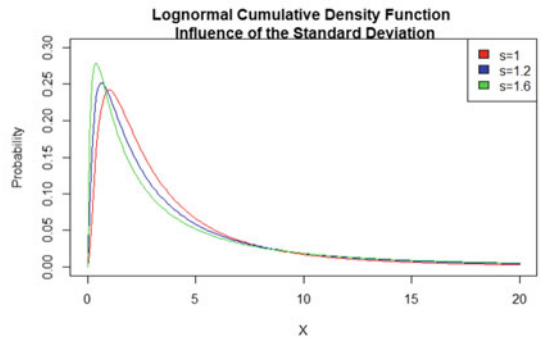
```r
x <- seq(0,20,0.1)
y <- dlnorm(x,1,0.4) #2nd parameter is mean, #3rd parameter is standard deviation
plot(x,y,
     main = "Lognormal Cumulative Density Function \n Influence of the Standard Deviation",
     type="l",
     col="red",
     xlab = "X",ylab = "Probability")
y <- dlnorm(x,1,0.6)
lines(x,y,col="blue")
y <- dlnorm(x,1,1)
lines(x,y,col="green")
legend("topright",c("s=0.4","s=0.6","s=1"),fill = c("red","blue","green"))
```



We can see as the standard deviation decreases while keeping the mean constant, it causes the lognormal distribution to peak.

Now we will see how the lognormal distribution behaves as the standard deviation increases.

```r
x <- seq(0,20,0.1)
y <- dlnorm(x,1,1) #2nd parameter is mean, #3rd parameter is standard deviation
plot(x,y,
     main = "Lognormal Cumulative Density Function \n Influence of the Standard Deviation",
     type="l",
     col="red",
     ylim = c(0,0.3),
     xlab = "X",ylab = "Probability")
y <- dlnorm(x,1,1.2)
lines(x,y,col="blue")
y <- dlnorm(x,1,1.4)
lines(x,y,col="green")
legend("topright",c("s=1","s=1.2","s=1.6"),fill = c("red","blue","green"))
```



From the above graph, we can see that as the standard deviation increases, the lognormal peaks and shifts to the left.

7.5.2 Lognormal Distribution in Python

The *Python* commands to produce lognormal distribution graphs are given below:

```
[1] #Jun-19-2020
    import numpy as np
    import scipy.stats as st
    import matplotlib.pyplot as plt
    import math as m
    from scipy.special import erf
```

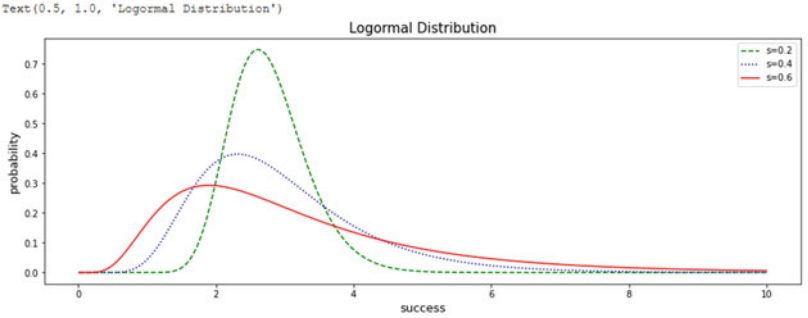
```
[2] def LognormPlot(mu, sd):
      x = np.linspace(0.01,10,num = 200, endpoint=True, dtype = None, axis = 0)
      length = len(x)
      y = []
      for i in range(length):
        y.append(1/(sd*x[i]*m.sqrt(2* m.pi)) * m.exp(-1/(sd*sd*2)*(m.log(x[i])-mu)**2))
      return x,y
```

```
[3] def cdfPlot(mu, sd):
      x = np.linspace(0.01,10,num = 200, endpoint=True, dtype = None, axis = 0)
      length = len(x)
      y = []
      for i in range(length):
        y.append(1/2*(1+erf((m.log(x[i])-mu)/(sd*m.sqrt(2)))))
      return x,y
```

```
[4] # 2nd argument is mean
    # 1st argument is
    a,val = LognormPlot(1,0.2)#very useful lib to calculate density
    a,val2 = LognormPlot(1,0.4)
    a,val3 = LognormPlot(1,0.6)

    plt.figure(figsize=(15,5))
    plt.plot(a,val,'g--')
    plt.plot(a,val2,'b:')
    plt.plot(a,val3,'r')
    plt.legend(['s=0.2','s=0.4','s=0.6'])
    plt.xlabel('success',fontsize=13)
    plt.ylabel('probability',fontsize=13)
    plt.title('Logormal Distribution',fontsize = 15)
```

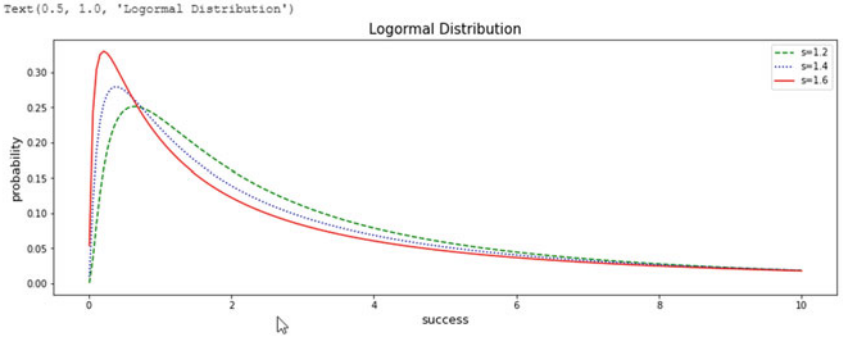
```
Text(0.5, 1.0, 'Logormal Distribution')
```




```
a,val = LognormPlot(1,1.2)#very useful lib to calculate density
a,val2 = LognormPlot(1,1.4)
a,val3 = LognormPlot(1,1.6)

plt.figure(figsize=(15,5))
plt.plot(a,val,'g--')
plt.plot(a,val2,'b:')
plt.plot(a,val3,'r')
plt.legend(['s=1.2','s=1.4','s=1.6'])
plt.xlabel('success',fontsize=13)
plt.ylabel('probability',fontsize=13)
plt.title('Logormal Distribution',fontsize = 15)
```

```
Text(0.5, 1.0, 'Logormal Distribution')
```



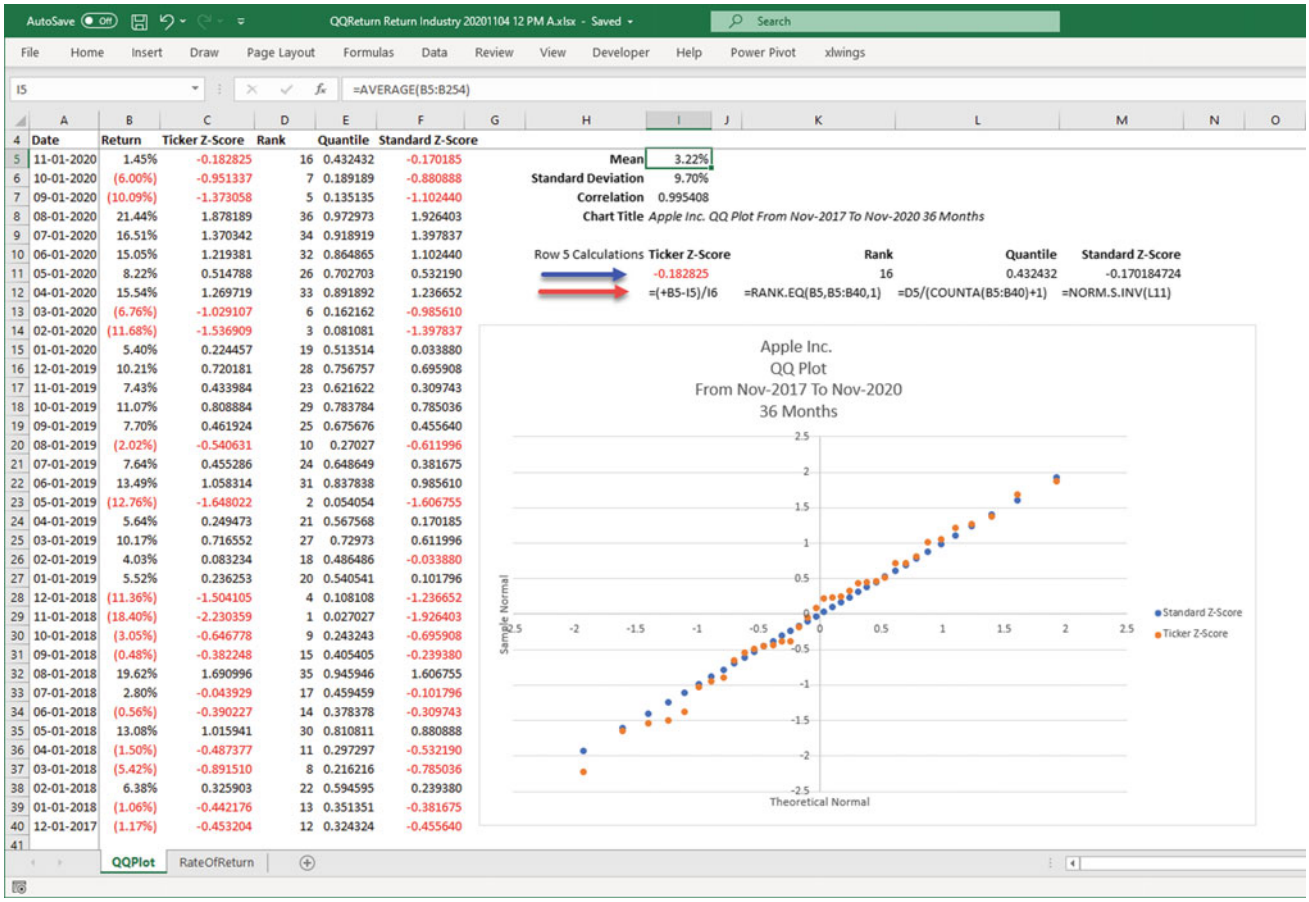
7.6 Normal Quantile–Quantile (QQ) Plot in Excel

In this section, we will test to see if a set of returns from a ticker for a specific period is approximately normal. We will use a Quantile–Quantile (QQ) plot to do this test. The QQ plot will help us visually determine if a data set is approximately normally distributed. If the data set is approximately normally distributed, the QQ plot will be approximately a straight line.

It is instructive to remember that a quantile splits an area or set of numbers into equal parts. A quartile is a quantile that is split into four equal parts.

To test the normality of returns, we will compare the rank returns in ascending order to a standard normal distribution. We do this by calculating the *Z-scores* of the returns and comparing it against the *Z-score* of a standard normal distribution.

The *Normal Q–Q Plot* of Apple Inc's returns from January 2000 to November 2020 is shown below. We used workbook of Chap. 4 that uses power query to retrieve and calculate a ticker's returns.



From the QQ plot, we can see that Apple Inc's returns from November 2017 to November 2020 are approximately normal because the orange data series is approximately normal.

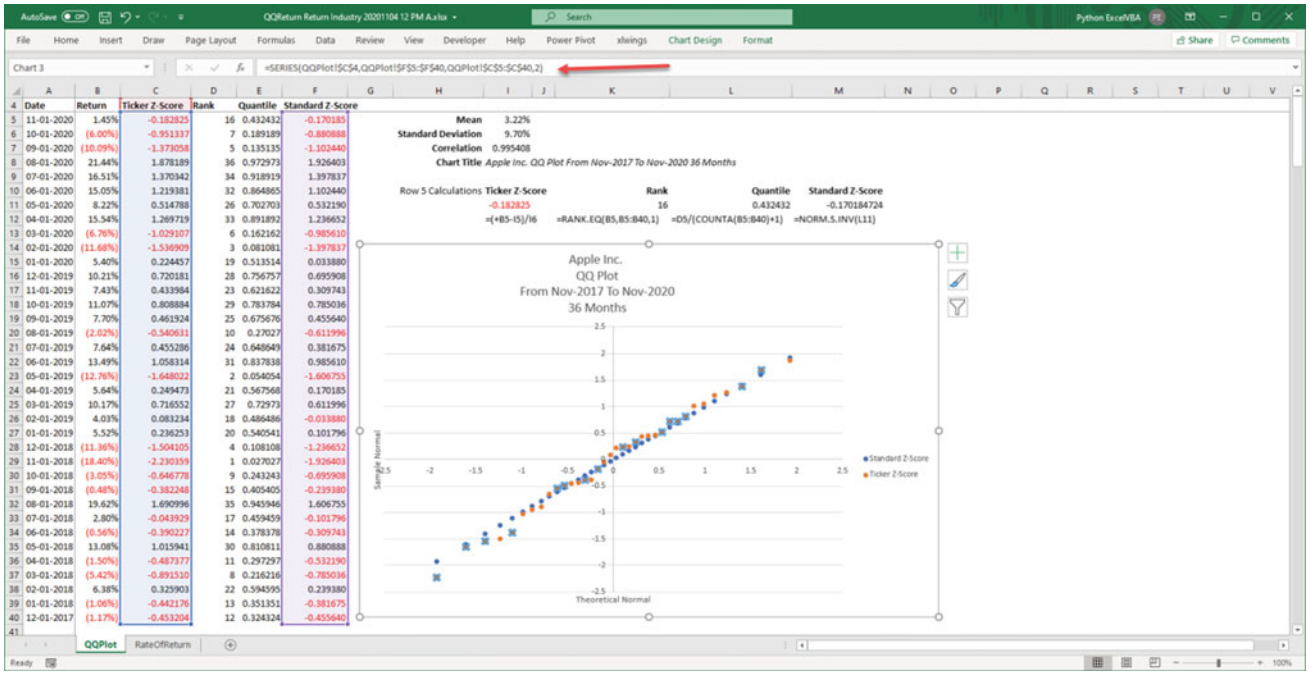
The first thing we will do is calculate the *Z-score* of every return of a ticker. In calculating the *Z-score,* we need to know the mean of the returns and the standard deviation of the returns. Cell J5 shows the mean of the returns. Cell J6 shows the standard deviation of the returns. The formula for the *Z-score* of the November 2020 return for Apple is shown in cell J12.

The next thing we need to do is to rank the returns in ascending order. We use the Excel formula *RANK.EQ* to do this. The formula to rank the returns is shown in cell L12.

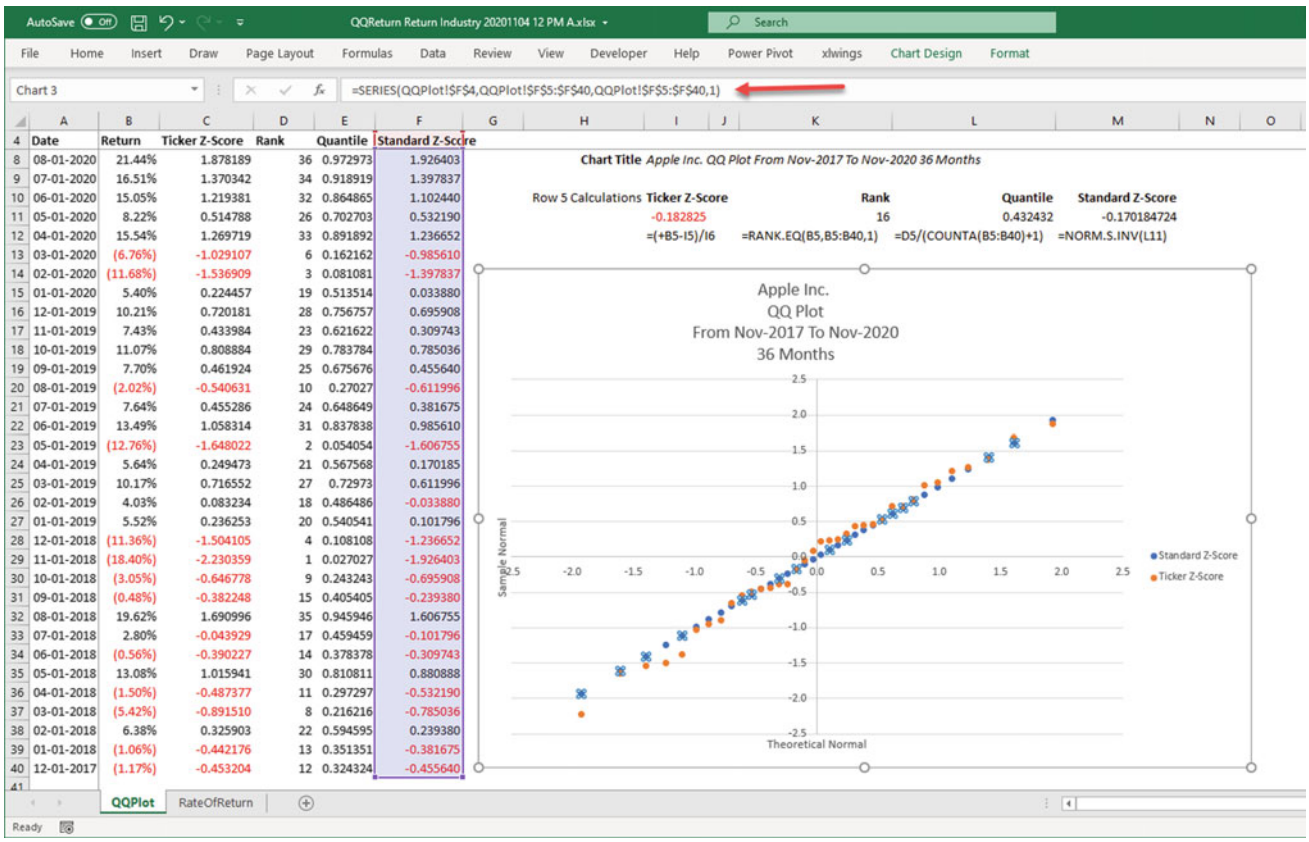
The next thing we need to do is to calculate the quantile for each return. We will use the calculated quantile to find the corresponding *Z-score* of a standard normal distribution. The formula to calculate the quantile for each return is shown in cell M12. There are 36 Apple Inc. returns. These returns will result in 37 equal areas under the normal distribution.

The next thing we need to do is find the corresponding *Z-score* of the standard normal distribution for each return. We will use the quantile for each return to do this. We will use the Excel formula *NORMAL.S.INV* to do this. The formula to calculate the corresponding standard normal distribution *Z-score* for each return is shown in cell N12.

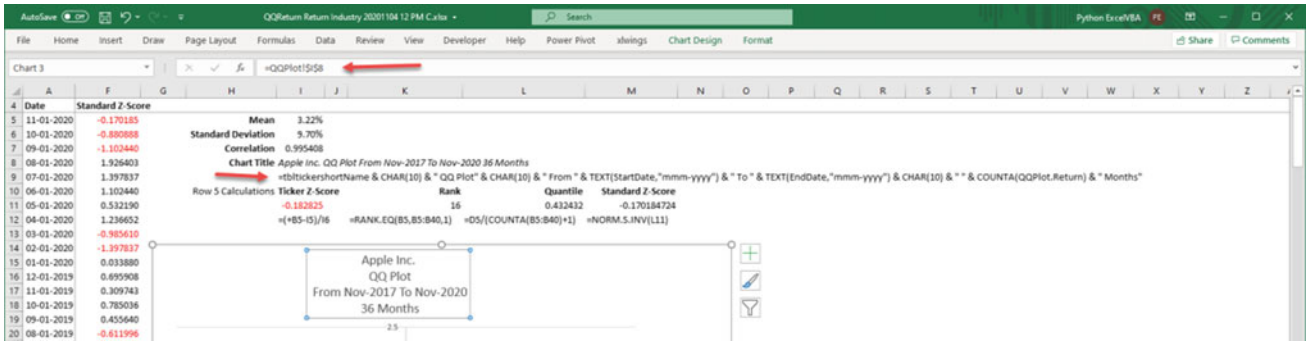
The graph has two data series. One of the data series plots the *Ticker Z-score* against the *Standard Z-Score*. If the data set of Apple Inc's return is approximately normal, then the plot of this data series will be approximately normal.



The other data series plots the *Standard Z-Score* against itself.



The chart title references cell I8. Cell I9 shows the formula in cell I8. Notice that the formula contains the function *CHAR (10)*. This function is the carriage function.



7.7 Normal Quantile–Quantile (QQ) Plot in Python

The Python code to create a QQ plot for a ticker is given below:

```
[27] #https://pypi.org/project/yfinance/
     pip install yfinance

[28] pip install requests

[29] pip install bs4

[39] #Date: Apr-03-2021
     import yfinance as yf
     import matplotlib.pyplot as plt
     import numpy as np
     import datetime as dt#get current date
     import pandas as pd
     from scipy import stats
     import requests
     from bs4 import BeautifulSoup
     from google.colab import files
```

```
[40] def html_beautifulsoup(url):
         response = requests.get(url)
         html = response.content

         #beautifulsoup html will not look like soure
         #review beautifulsoup html to use find and findall
         soup = BeautifulSoup(html,"lxml")
         return soup

     #soup_html: soup html is not same as source html
     #get ticker description from ticker Yahoo!finance page
     #finance.yahoo.com/quote/<ticker>
     def yahoo_description(soup_html):
         element = soup_html.find('h1',class_='D(ib) Fz(18px)')
         return element.text
```

```
[41] def stock_return(Stock_Name, Start_T, End_T, T_Interval):
       Stock_price = yf.download(Stock_Name,start=Start_T,End=End_T,interval = T_Interval)['Adj Close'].dropna()
       Ret = Stock_price.pct_change()[1:].dropna()
       info = Stock_Name
       Ret = Ret.sort_values()
       stdRet = (Ret-np.mean(Ret))/np.std(Ret)
       ##get z-score
       zscore = stats.zscore(list(range(1,len(Ret)+1)))
       cdf = stats.norm.cdf(zscore)
       a=stdRet.to_list()
       stderr = sum(pow(a-zscore,2))
       return stdRet,info, zscore,Ret,cdf,stderr
```

The following Python code prompts the user for ticker information:

```
[44]
    #initialize dates
    previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
    Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
    years_ago = dt.date.today()-dt.timedelta(days=5 * 365)
    Default_start = years_ago.strftime("%Y-%m-%d")

    # request user input
    Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
    Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
    End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
    T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'

    soup_html = html_beautifulsoup("https://finance.yahoo.com/quote/"  + Ticker)
    tickerDescription = yahoo_description(soup_html)
    #ticker=Ticker

    #get stock info
    stdRet,info,zscore,Ret,cdf,stderr = stock_return(Ticker,Start,End,T_interval)
```

```
Please enter ticker(Default = JNJ)-->AAPL
Enter start date(Default = 5 Years ago, 2016-04-05)-->2017-11-01
Enter end date(Default = Yesterday, 2021-04-03)-->2020-11-01
Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo
[**********100%**********]  1 of 1 completed
```

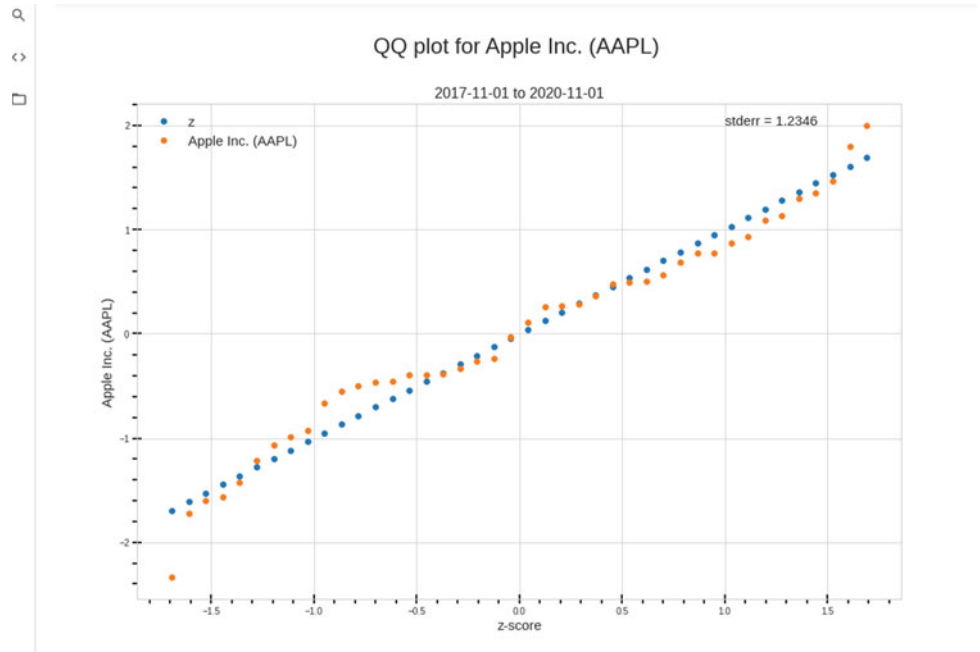
The following Python code plots the QQ plot for the specified ticker:

```
#plot chart
plt.figure(figsize=(15,10))
plt.style.use('seaborn-whitegrid')#set the background to be grid
plt.minorticks_on()
plt.tick_params(which='minor', length=5, width=2, direction='out')
plt.tick_params(which='major', length=10, width=2, direction='inout')
plt.plot(zscore,zscore,'o')
plt.plot(zscore,stdRet,'o')
plt.text(1,2,'stderr = {:.4f}'.format(stderr),fontsize = 15)
plt.legend(['z',tickerDescription],prop={'size': 15})
plt.suptitle('   QQ plot for '+tickerDescription,fontsize = 24)
plt.xlabel('z-score',fontsize = 15)
plt.ylabel(tickerDescription,fontsize = 15)
plt.title(Start+' to '+End,fontsize = 16)
stdRet = pd.DataFrame(stdRet.T)

print("")
```

The following Python code prints out the rate of return for the particular ticker:

```
#print chart dataset
pd.set_option('display.max_rows', None)
pd.options.display.float_format = '{:.4f}'.format
zscore = stats.zscore(list(range(1,len(Ret)+1)))
zscore = pd.DataFrame(zscore)
zscore = zscore.set_index(Ret.index)
cdf = pd.DataFrame(cdf)
cdf = cdf.set_index(Ret.index)
chart = pd.concat([Ret,stdRet,cdf,zscore],axis=1)
chart.columns = ['return','Return Z-Score','cdf','Standard Z-Score']
print(chart)
```



The following prints out the rate of return for a particular ticker:

```
#print chart dataset
pd.set_option('display.max_rows', None)
pd.options.display.float_format = '{:.4f}'.format
zscore = stats.zscore(list(range(1,len(Ret)+1)))
zscore = pd.DataFrame(zscore)
zscore = zscore.set_index(Ret.index)
cdf = pd.DataFrame(cdf)
cdf = cdf.set_index(Ret.index)
chart = pd.concat([Ret,stdRet,cdf,zscore],axis=1)
chart.columns = ['return','Return Z-Score','cdf','Standard Z-Score']
print(chart)
```

Date	return	Return Z-Score	cdf	Standard Z-Score
2018-11-01	-0.1840	-2.3313	0.0454	-1.6913
2019-05-01	-0.1276	-1.7179	0.0538	-1.6088
2020-02-01	-0.1168	-1.6009	0.0635	-1.5263
2018-12-01	-0.1136	-1.5664	0.0744	-1.4438
2020-09-01	-0.1009	-1.4284	0.0867	-1.3613
2021-02-01	-0.0811	-1.2131	0.1005	-1.2788
2020-03-01	-0.0676	-1.0661	0.1158	-1.1963
2020-10-01	-0.0600	-0.9842	0.1327	-1.1138
2018-03-01	-0.0542	-0.9212	0.1512	-1.0313
2018-10-01	-0.0305	-0.6634	0.1714	-0.9488
2019-08-01	-0.0202	-0.5516	0.1932	-0.8663
2018-04-01	-0.0150	-0.4956	0.2166	-0.7838
2017-12-01	-0.0117	-0.4596	0.2416	-0.7013
2018-01-01	-0.0106	-0.4479	0.2680	-0.6188
2018-06-01	-0.0056	-0.3932	0.2959	-0.5363
2021-01-01	-0.0055	-0.3922	0.3250	-0.4538
2018-09-01	-0.0048	-0.3848	0.3552	-0.3713
2021-04-01	0.0000	-0.3324	0.3864	-0.2888
2021-04-01	0.0070	-0.2569	0.4183	-0.2063
2021-03-01	0.0088	-0.2364	0.4508	-0.1238
2018-07-01	0.0280	-0.0285	0.4835	-0.0413
2019-02-01	0.0403	0.1054	0.5165	0.0413
2020-01-01	0.0540	0.2541	0.5492	0.1238
2019-01-01	0.0552	0.2666	0.5817	0.2063
2019-04-01	0.0564	0.2805	0.6136	0.2888
2018-02-01	0.0638	0.3610	0.6448	0.3713
2019-11-01	0.0743	0.4748	0.6750	0.4538
2019-07-01	0.0764	0.4973	0.7041	0.5363
2019-09-01	0.0770	0.5043	0.7320	0.6188
2020-05-01	0.0822	0.5599	0.7584	0.7013
2020-11-01	0.0936	0.6842	0.7834	0.7838
2019-03-01	0.1017	0.7724	0.8068	0.8663
2019-12-01	0.1021	0.7763	0.8286	0.9488
2019-10-01	0.1107	0.8697	0.8488	1.0313
2020-12-01	0.1165	0.9328	0.8673	1.1138
2018-05-01	0.1308	1.0877	0.8842	1.1963
2019-06-01	0.1349	1.1324	0.8995	1.2788
2020-06-01	0.1505	1.3020	0.9133	1.3613
2020-04-01	0.1554	1.3550	0.9256	1.4438
2020-07-01	0.1651	1.4610	0.9365	1.5263
2018-08-01	0.1962	1.7987	0.9462	1.6088
2020-08-01	0.2144	1.9959	0.9546	1.6913

7.8 Summary

In this chapter, we looked at the uniform, normal, standard normal, and lognormal continuous distributions. We saw that the normal distribution is influenced by the mean and standard deviation. For the normal distribution, we saw that changing the mean of the normal distribution causes the normal distribution to shift. We saw that changing the standard deviation of the normal distribution causes the normal distribution to become fatter or skinnier.

We saw that the lognormal distribution is influenced by the mean and the standard deviation. We saw that increasing the mean causes the lognormal distribution to become flatter. We saw that increasing the standard deviation causes the lognormal to peak and approach 0. We saw that decreasing the standard deviation causes the lognormal distribution to peak.

We used Excel, Python, and R to create and demonstrate these distributions.

In this chapter, we used Excel and Python to create the QQ plot for Apple Inc's rate of return from November 2017 to November 2011. From the QQ plot, we can conclude that Apple Inc's rate of return is approximately normal.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th Edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba." Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language." BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data." O'Reilly Media, 2017.

Check for updates

8 Sampling Distributions and Central Limit Theorem

8.1 Introduction

Many times, it is impossible or too costly to analyze the population data. Because of this, we are only able to analyze a sample from the population. After analyzing the sample data, can we have a good general understanding of the population data from the sample data? The answer is yes. In this chapter, we will study why the answer is yes.

This chapter is divided into the following sections. Section 8.2 discusses sample distribution in Excel, Sect. 8.3 talks about how the mean of sample distribution equals mean of population, and Sect. 8.4 discusses the sample distribution in Python. Section 8.5 goes over the central limit theorem, and finally Sect. 8.6 summarizes the chapter.

8.2 Sample Distribution in Excel

To study this issue, we will sample from a very small population. The concepts that we learn here will apply to large populations. To make things simple, let's suppose that the population that we are interested in analyzing has the following six elements:

$$1, 2, 3, 4, 5, 6$$

The mean of the above population would be 3.5.

Suppose we are interested in learning about the mean of the population. Suppose we are restricted to only randomly sampling two items from the population at a time. One possible sample that we might get is a sample that consists of elements 1 and 2. The mean of this sample would be 1.5. What can we say about the population mean from this sample? Not much. Let's say we randomly get another sample from the population. What is the chance of us getting another sample that has elements 1 and 2 again? The chance would be small but greater than zero. Our second sample might contain elements 1 and 3; this leads us to the question, "What are all the possible combinations for a sample size of 2 from the population of interests?" All the possible combinations and the mean of each possible combination are shown below:

Possible Sample Combinations		
	Mean	
1	2	1.5
1	3	2
1	4	2.5
1	5	3
1	6	3.5
2	3	2.5
2	4	3
2	5	3.5
2	6	4

(continued)

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_8

Possible Sample Combinations		
	Mean	
3	4	3.5
3	5	4
3	6	4.5
4	5	4.5
4	6	5
5	6	5.5

Looking at the above table, we will notice more occurrences of samples with a mean of 3.5 than 2; this means that there would be a greater chance for us to get a sample with a mean of 3.5. In the above table, there are 15 elements. Let's suppose that there is an equal chance to select any of the samples randomly. If this is the case, we can then create the table below that shows the probability of getting each specific mean.

Sampling Distribution	
1.5	6.67%
2	6.67%
2.5	13.33%
3	13.33%
3.5	20.00%
4	13.33%
4.5	13.33%
5	6.67%
5.5	6.67%

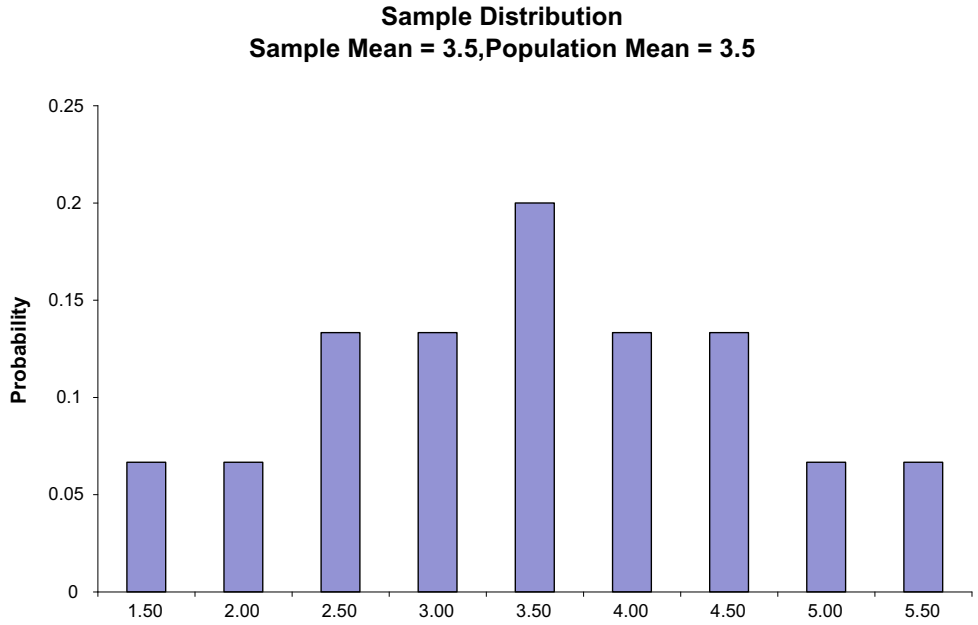
The table above is often called a sampling distribution because it's the possible distribution of all the means from the samples.

When we get a sample with elements 1 and 2, we can say that there was a 6.67% chance of getting a sample with a mean of 1.5. And if we get a sample that contains elements 1 and 3, we can say that there was a 20% chance of getting a sample with a mean of 3.5.

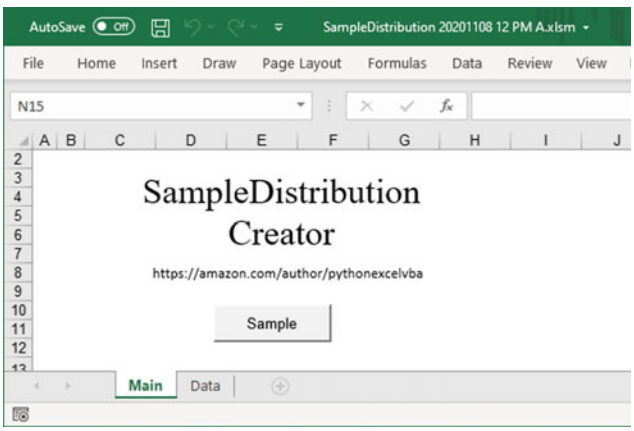
It is important to note that the mean of this sample distribution is

$$\begin{aligned}&(1.5 * 0.067) + (2 * 0.067) + (2.5 * 0.1333)\\&\quad + (3 * 0.1333) + (3.5 * 0.2) + (4 * 0.1333)\\&\quad + (4.5 * 0.1333) + (5 * 0.067) + (5.5 * 0.067) = 3.5\end{aligned}$$

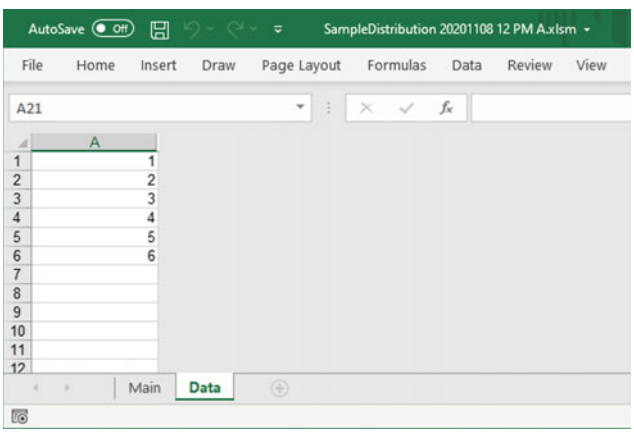
Which is the same as the population mean. A histogram of the sample distribution is given below:



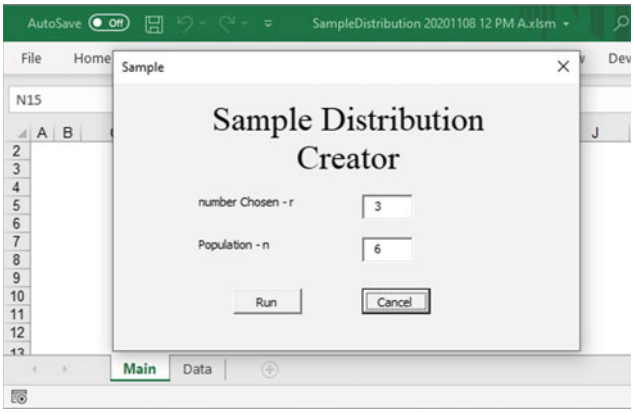
Let's now create a sample distribution with a sample size of 3 and 4.



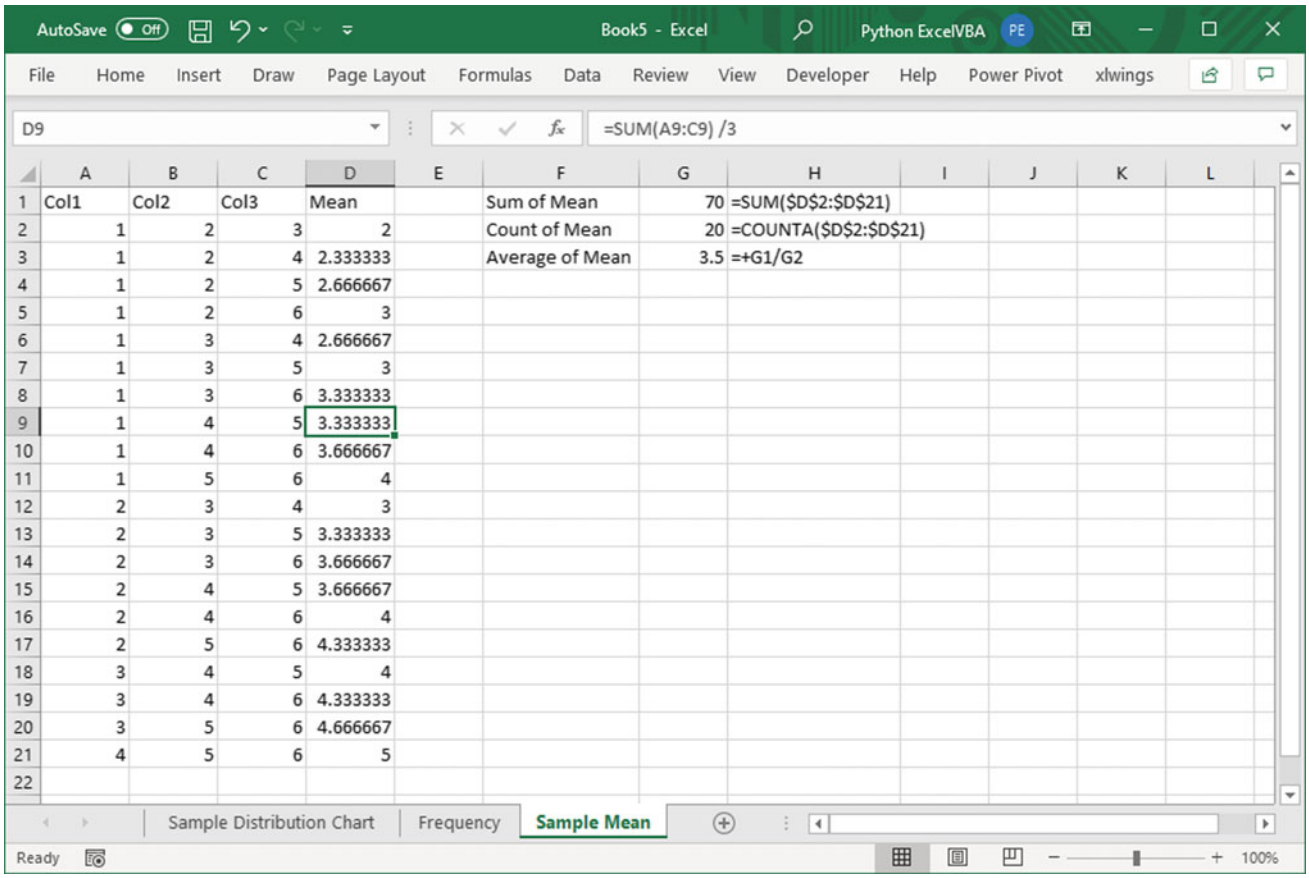
The workbook *sampledistribution.xlsm* consists of two worksheets. The *Main* worksheet is where we push the button to create a sample distribution from the population distribution. The *Data* worksheet is where we indicate the *population* data set. Let's put the population data set in the worksheet *Data* as shown below:



Now we will create the sample distribution when the same size is 3 by pressing the button on the Main worksheet. Pressing the button will result in the following dialog box:

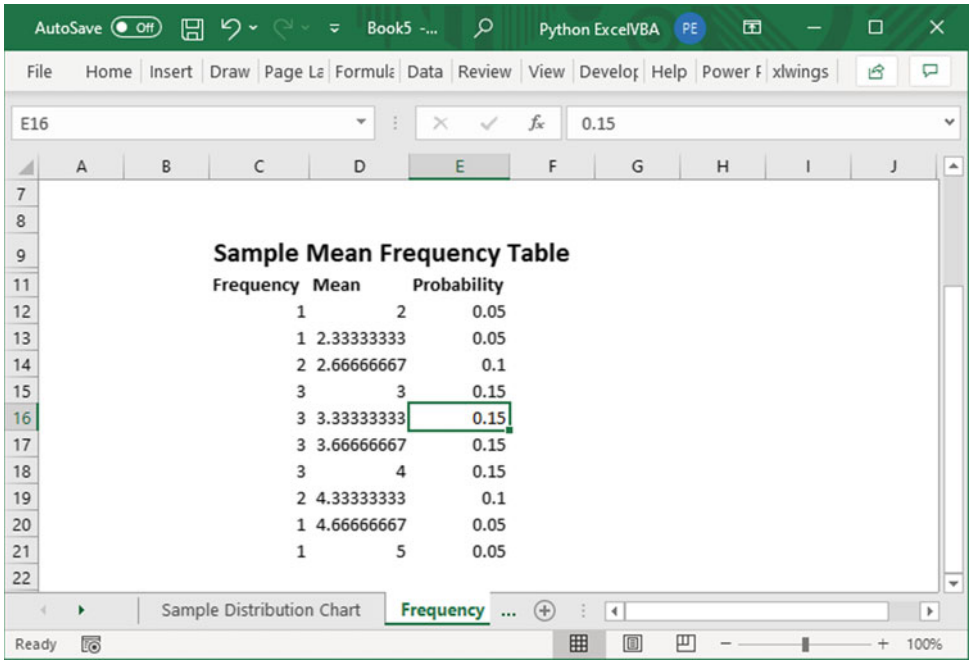


If we fill out the above dialog box as shown above and then press the *Run* button, a workbook with the following sheets will be created. On the left side of the first sheet, *Sample Mean*, list all the combinations of a sample size of 3 from a population of 6. Column D shows the mean of every sample. In cell, G3 shows the average of the mean.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Col1	Col2	Col3	Mean		Sum of Mean	70	=SUM($D$2:$D$21)				
2	1	2	3	2		Count of Mean	20	=COUNTA($D$2:$D$21)				
3	1	2	4	2.333333		Average of Mean	3.5	=+G1/G2				
4	1	2	5	2.666667								
5	1	2	6	3								
6	1	3	4	2.666667								
7	1	3	5	3								
8	1	3	6	3.333333								
9	1	4	5	3.333333								
10	1	4	6	3.666667								
11	1	5	6	4								
12	2	3	4	3								
13	2	3	5	3.333333								
14	2	3	6	3.666667								
15	2	4	5	3.666667								
16	2	4	6	4								
17	2	5	6	4.333333								
18	3	4	5	4								
19	3	4	6	4.333333								
20	3	5	6	4.666667								
21	4	5	6	5								
22												

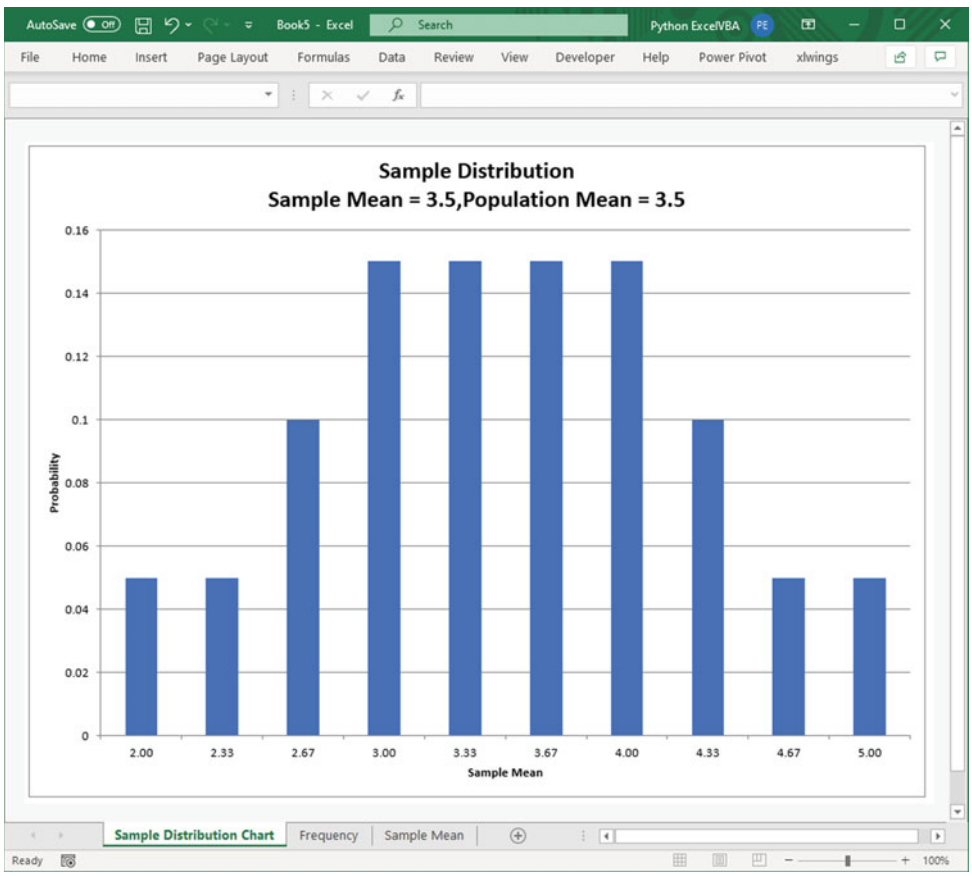
The second sheet, *Frequency*, contains a table that shows the frequency of the means on the *Sample Mean* sheet.



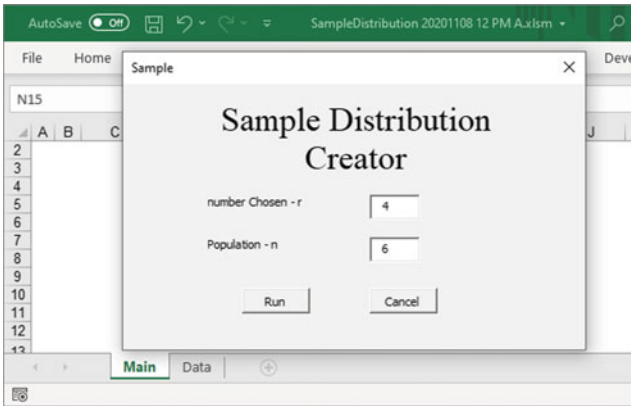
Sample Mean Frequency Table

Frequency	Mean	Probability
1	2	0.05
1	2.33333333	0.05
2	2.66666667	0.1
3	3	0.15
3	3.33333333	0.15
3	3.66666667	0.15
3	4	0.15
2	4.33333333	0.1
1	4.66666667	0.05
1	5	0.05

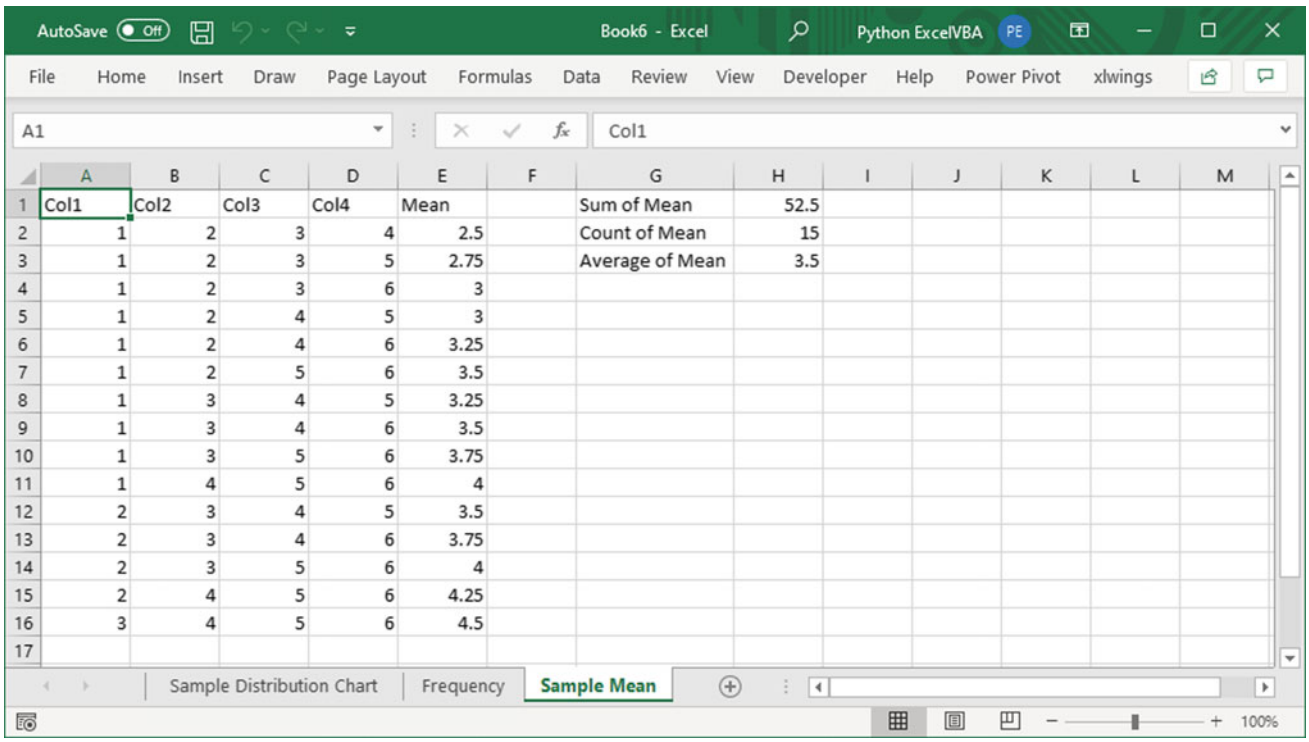
The third sheet, *Sample Distribution Chart*, shows the chart of the table in the *Frequency* sheet.



Now we will create the sample distribution when the same size is 4 by pressing the button on the *Main* worksheet. Pressing the button will result in the following dialog box:

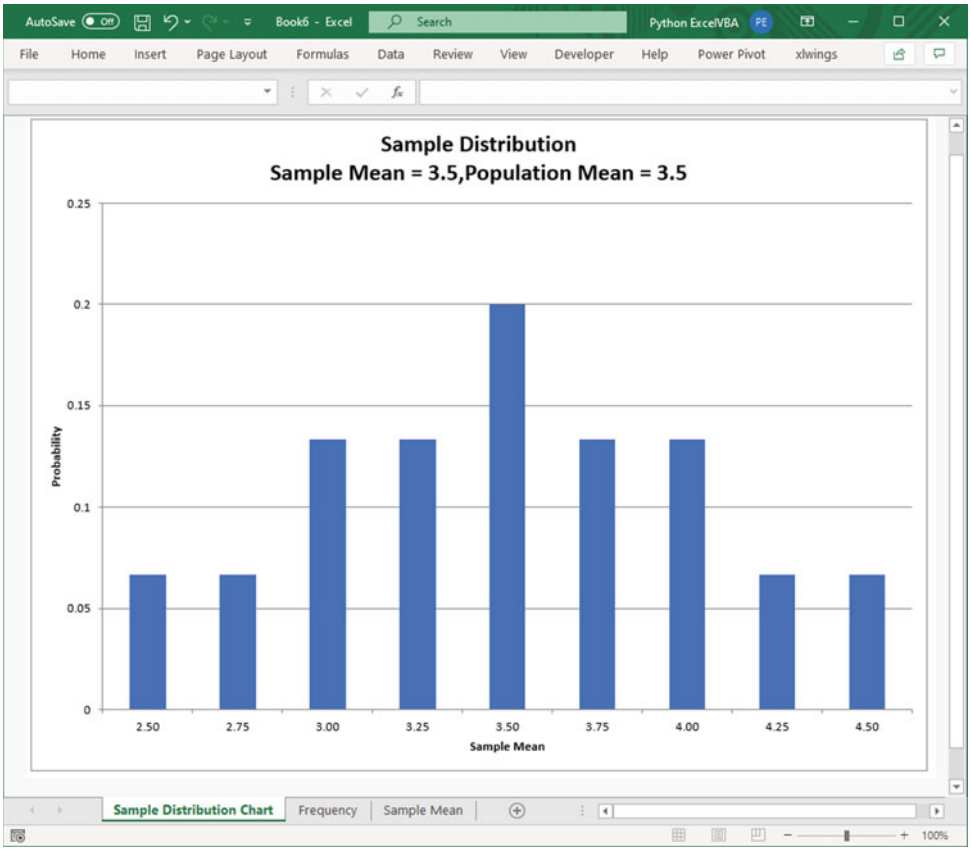


The corresponding *Sample Mean* sheet is given below. Notice that the number of combinations decreased, and the average mean is still 3.5.



	A	B	C	D	E	F	G	H
1	Col1	Col2	Col3	Col4	Mean		Sum of Mean	52.5
2	1	2	3	4	2.5		Count of Mean	15
3	1	2	3	5	2.75		Average of Mean	3.5
4	1	2	3	6	3			
5	1	2	4	5	3			
6	1	2	4	6	3.25			
7	1	2	5	6	3.5			
8	1	3	4	5	3.25			
9	1	3	4	6	3.5			
10	1	3	5	6	3.75			
11	1	4	5	6	4			
12	2	3	4	5	3.5			
13	2	3	4	6	3.75			
14	2	3	5	6	4			
15	2	4	5	6	4.25			
16	3	4	5	6	4.5			
17								

The corresponding *Sample Distribution Chart is given below*:

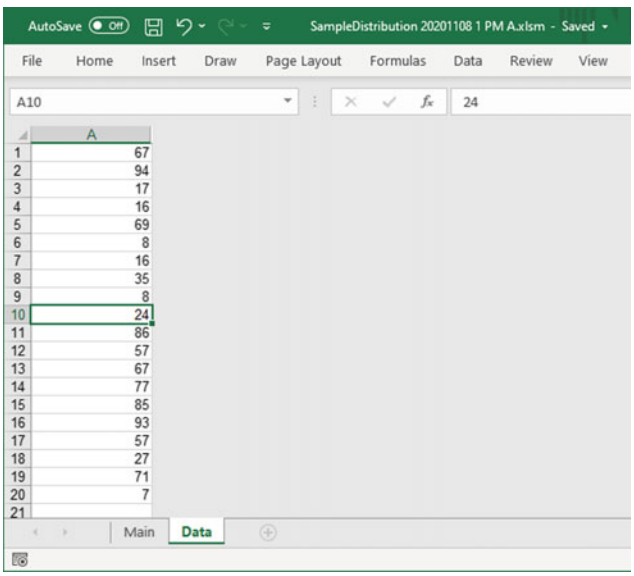


Notice that as the sample size increases, the resulting sample distribution approaches a normal distribution.

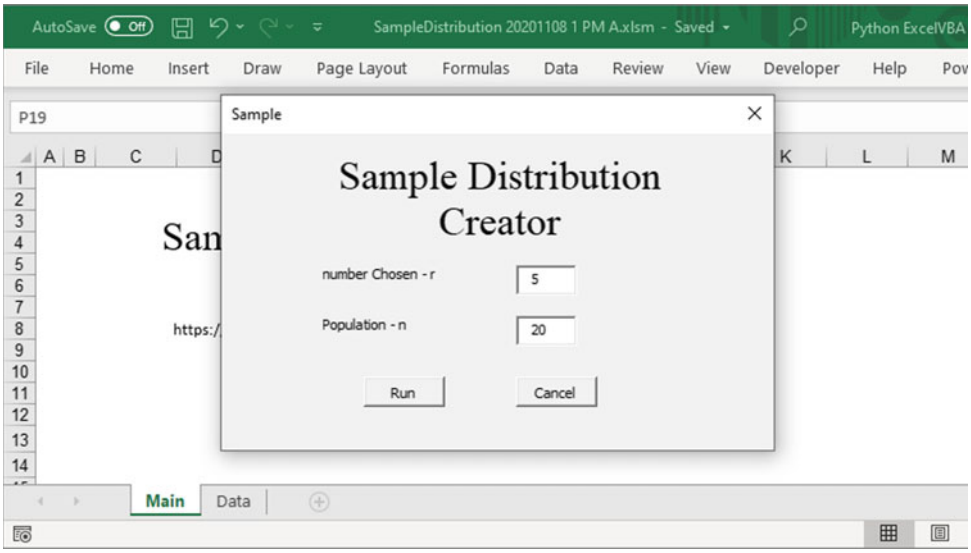
8.3 Mean of Sample Distribution Equals Mean of Population

It is important to note that the mean of a sample distribution is equal to the mean of the population.

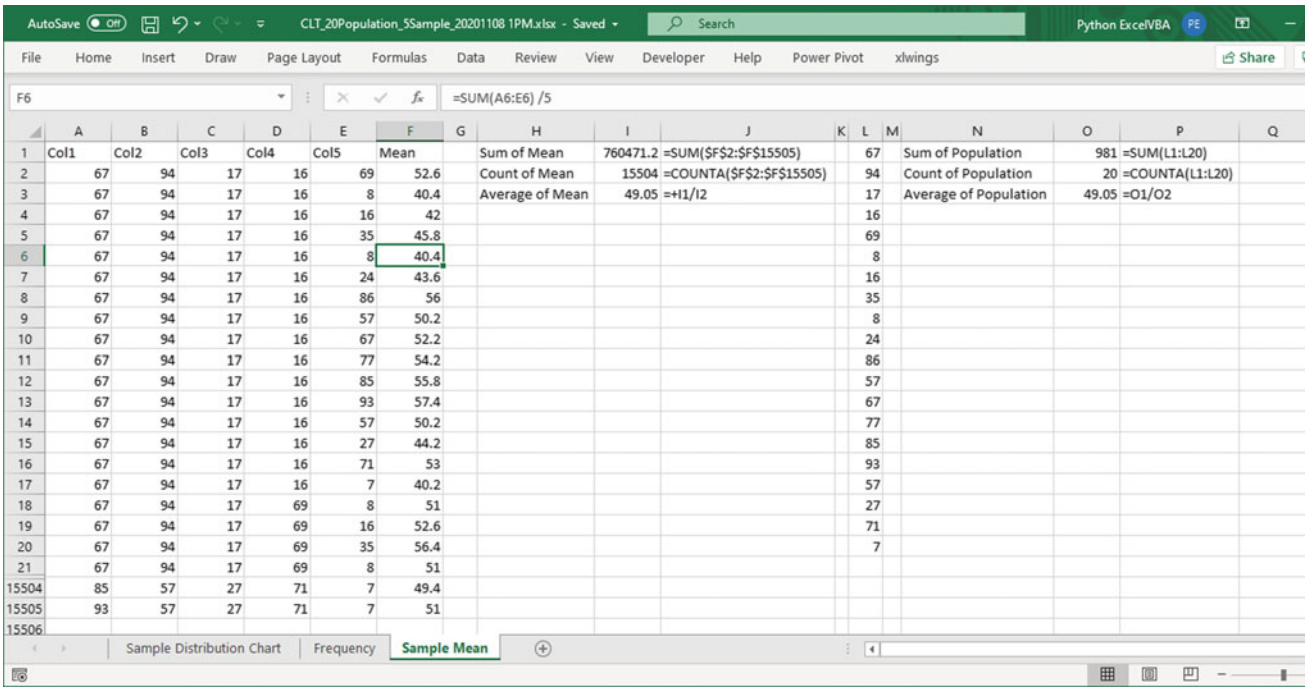
We will demonstrate this with the following population. The population below was randomly created.



With this population, we will create the sample distributions with the sizes 5 and 15. The following is the sample distribution when the sample size is 5.



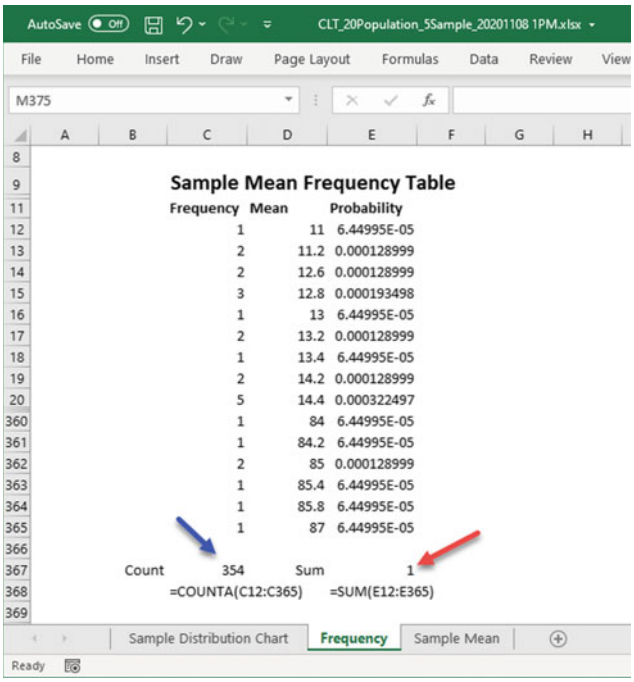
The corresponding *Sample Mean* sheet is given below:



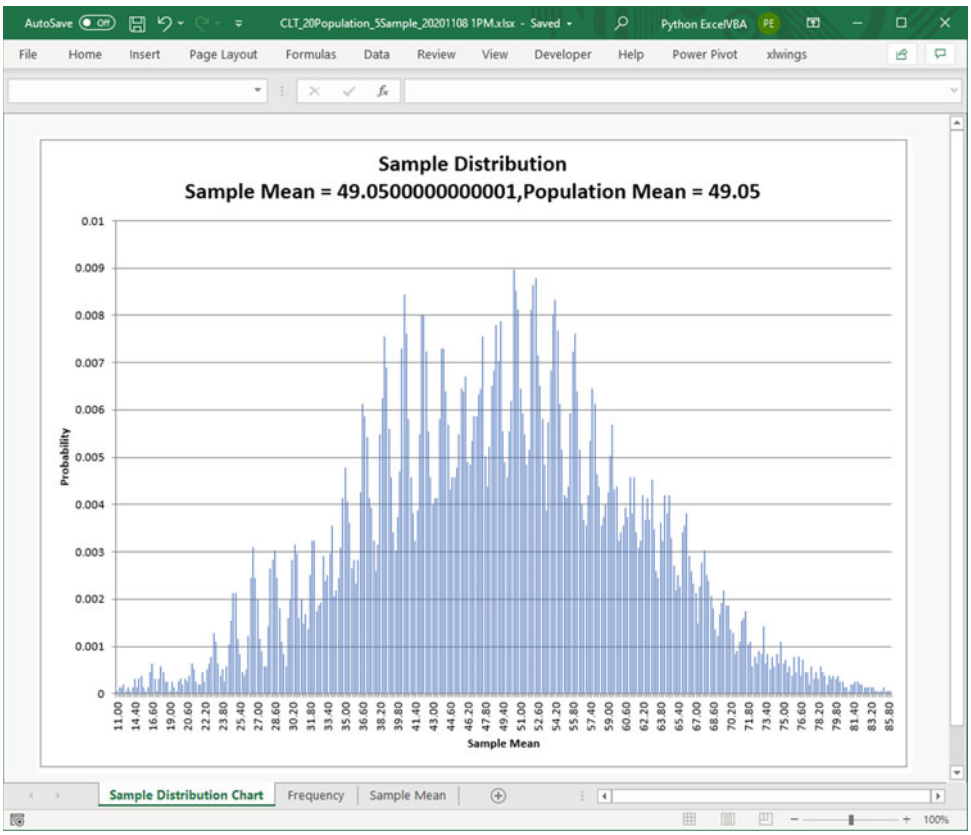
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Col1	Col2	Col3	Col4	Col5	Mean		Sum of Mean	760471.2	=SUM($F$2:$F$15505)		67		Sum of Population	981	=SUM(L1:L20)	
2	67	94	17	16	69	52.6		Count of Mean	15504	=COUNTA($F$2:$F$15505)		94		Count of Population	20	=COUNTA(L1:L20)	
3	67	94	17	16	8	40.4		Average of Mean	49.05	=+I1/I2		17		Average of Population	49.05	=O1/O2	
4	67	94	17	16	16	42						16					
5	67	94	17	16	35	45.8						69					
6	67	94	17	16	8	40.4						8					
7	67	94	17	16	24	43.6						16					
8	67	94	17	16	86	56						35					
9	67	94	17	16	57	50.2						8					
10	67	94	17	16	67	52.2						24					
11	67	94	17	16	77	54.2						86					
12	67	94	17	16	85	55.8						57					
13	67	94	17	16	93	57.4						67					
14	67	94	17	16	57	50.2						77					
15	67	94	17	16	27	44.2						85					
16	67	94	17	16	71	53						93					
17	67	94	17	16	7	40.2						57					
18	67	94	17	69	8	51						27					
19	67	94	17	69	16	52.6						71					
20	67	94	17	69	35	56.4						7					
21	67	94	17	69	8	51											
15504	85	57	27	71	7	49.4											
15505	93	57	27	71	7	51											
15506																	

The *Sample Mean* sheet shows 15,504 combinations. The average of the sample means is calculated in cell I3. The population is shown in column L. The population average is calculated in cell F3. This sheet shows that the average of the sample mean is equal to the population average.

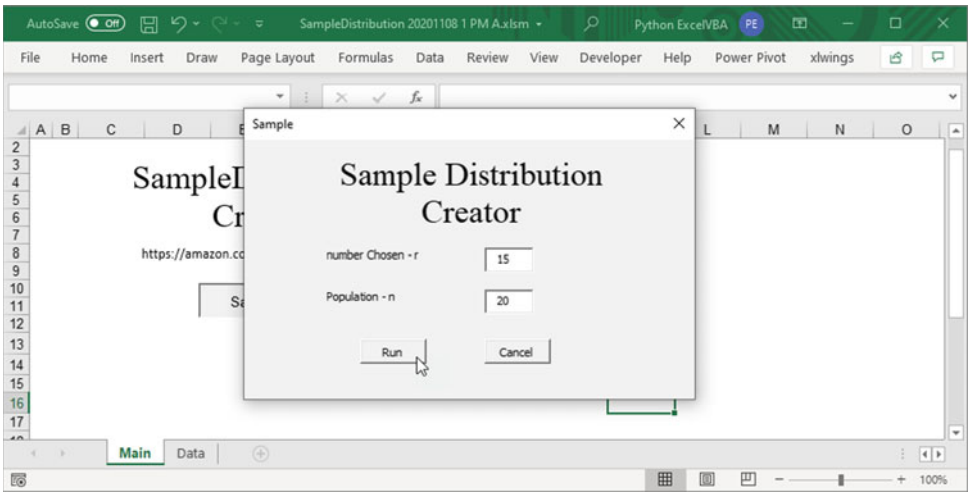
The corresponding *Frequency* sheet is given below:



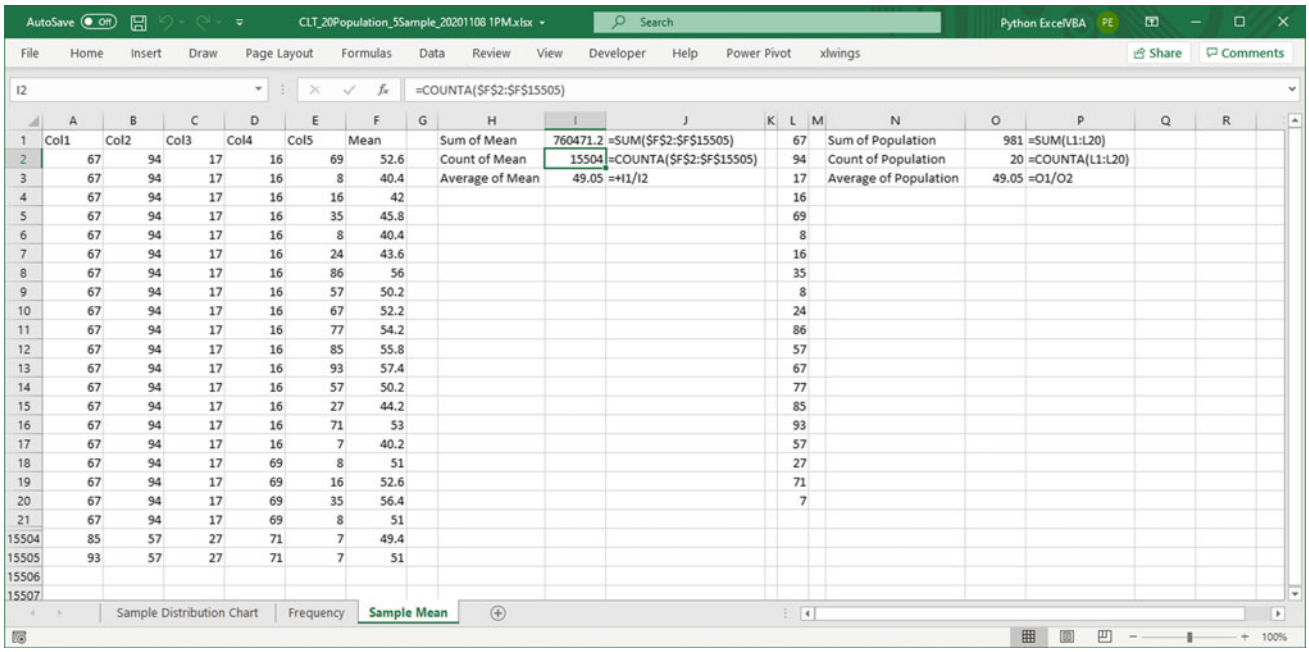
In the *Frequency* sheet, there are 354 unique means. Note that the sum of the probabilities is equal to 1. The corresponding *Sample Distribution Chart* sheet is given below:



The following is the sample distribution when the sample size is 15:

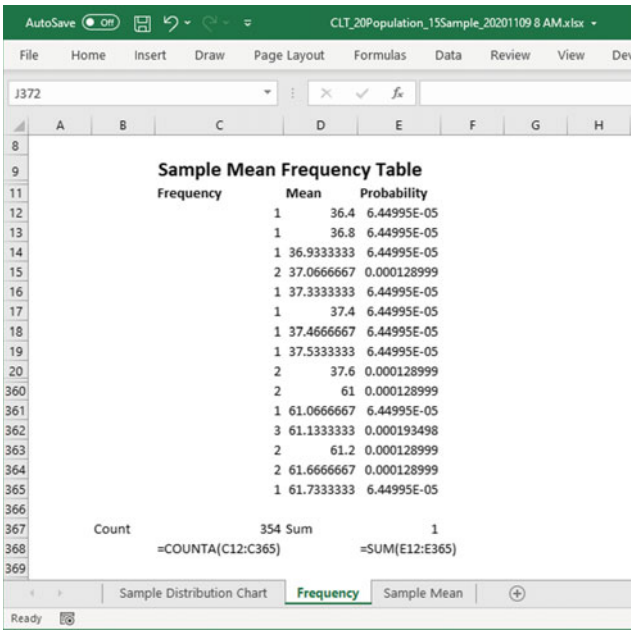


The corresponding *Sample Mean* sheet is given below:



The *Sample Mean* sheet shows 15,504 combinations. The average of the sample means is calculated in cell I3. The population is shown in column L. The population average is calculated in cell O3. This sheet shows that the average of the sample mean is equal to the population average. Note that even when the sample size increased from 5 to 15, the average of the sample mean equals the population average.

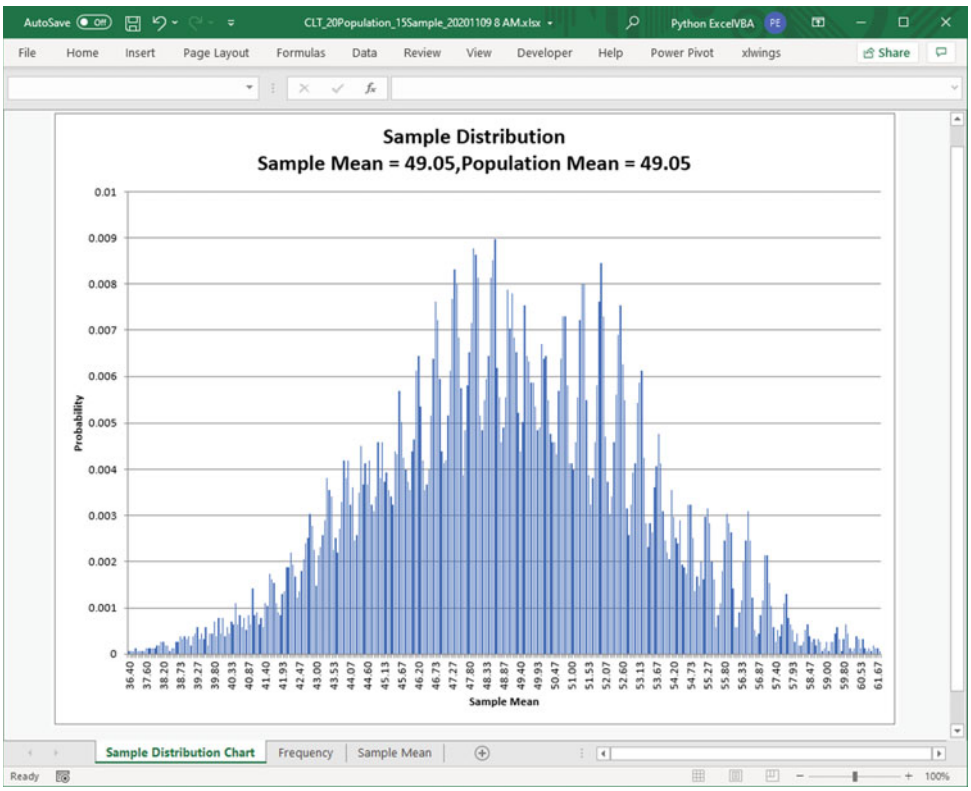
The corresponding *Frequency* sheet is given below:



Sample Mean Frequency Table

	Frequency	Mean	Probability
12	1	36.4	6.44995E-05
13	1	36.8	6.44995E-05
14	1	36.9333333	6.44995E-05
15	2	37.0666667	0.000128999
16	1	37.3333333	6.44995E-05
17	1	37.4	6.44995E-05
18	1	37.4666667	6.44995E-05
19	1	37.5333333	6.44995E-05
20	2	37.6	0.000128999
360	2	61	0.000128999
361	1	61.0666667	6.44995E-05
362	3	61.1333333	0.000193498
363	2	61.2	0.000128999
364	2	61.6666667	0.000128999
365	1	61.7333333	6.44995E-05
367	Count	354	Sum 1
368	=COUNTA(C12:C365)		=SUM(E12:E365)

In the *Frequency* sheet, there are 354 unique means. Note that the sum of the probabilities is also equal to 1. The corresponding *Sample Distribution Chart* sheet is given below:



8.4 Sample Distribution in Python

This section will use Python to demonstrate that the sample mean equals the population mean.

```
[17] #Apr-20-2020
     import numpy as np
     import matplotlib.pyplot as plt
     from itertools import combinations
     import pandas as pd
     from google.colab import files
```

```
class Combination:
  def __init__ (self, population):

    self.__population = population
    self.__combine = []
    self.__mean_result = []
    self.__counts=[]

  def comb_result(self,size):
    self.__combine = combinations(self.__population,int(size)) #Give the result of combination of population
    result = list(self.__combine)
    self.__mean_result = np.mean(result,axis=1)#calculate the mean of each combination
    dict_data = {'Combination':result,'Mean':self.__mean_result.T}
    frame_data = pd.DataFrame(dict_data)
    return frame_data

  def sample_dist_result(self):
    values, self.__counts = np.unique(self.__mean_result, return_counts=True) #count the unique mean
    self.__counts = np.divide(self.__counts,sum(self.__counts))
    formattedList = ['{:.2%}'.format(member) for member in self.__counts]
    dict_data = {'Sampling distribution':values,'Probability':formattedList}
    frame_data = pd.DataFrame(dict_data)
    return frame_data

  def Plot_hist(self):
    plt.figure(figsize=(10,5))
    plt.hist(self.__mean_result,bins=len(self.__counts),rwidth=0.5,density=True)# plot the density histogram
    plt.suptitle('Sample Distribution',fontsize=16)
    plt.title('Sample mean = {:.2f}, Population mean = {:.2f}'.format(np.mean(self.__population),np.mean(self.__population)))
    plt.xlabel('Mean')
    plt.ylabel('Probability')
```

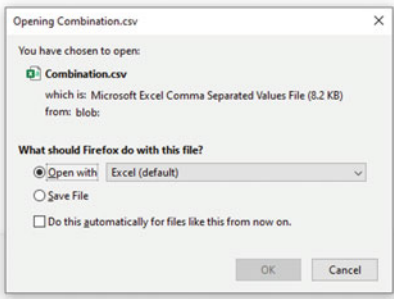
```
[30] #request user input
     population = input("Enter a list numbers separated by space. Default '67 94 17 16 69 8 16 35 8 24 86 57 67 77 85 93 57 27 71 7'-->") or \
         '67 94 17 16 69 8 16 35 8 24 86 57 67 77 85 93 57 27 71 7'
     population = population.split()
     population = [int(i) for i in population] #str to int
     size = input('Number of values in each sample? Default is 15 -->') or 15

     Enter a list numbers separated by space. Default '67 94 17 16 69 8 16 35 8 24 86 57 67 77 85 93 57 27 71 7'-->
     Number of values in each sample? Default is 15 -->
```

```
[31] #print list of combinations
     First = Combination(population)
     frame_data = First.comb_result(size)
     print(frame_data)

                                          Combination       Mean
     0      (67, 94, 17, 16, 69, 8, 16, 35, 8, 24, 86, 57,...  48.400000
     1      (67, 94, 17, 16, 69, 8, 16, 35, 8, 24, 86, 57,...  48.933333
     2      (67, 94, 17, 16, 69, 8, 16, 35, 8, 24, 86, 57,...  46.533333
     3      (67, 94, 17, 16, 69, 8, 16, 35, 8, 24, 86, 57,...  44.533333
     4      (67, 94, 17, 16, 69, 8, 16, 35, 8, 24, 86, 57,...  47.466667
     ...                                                  ...        ...
     15499  (69, 8, 16, 35, 24, 86, 57, 67, 77, 85, 93, 57...  51.933333
     15500  (69, 8, 16, 8, 24, 86, 57, 67, 77, 85, 93, 57,...  50.133333
     15501  (69, 8, 35, 8, 24, 86, 57, 67, 77, 85, 93, 57,...  51.400000
     15502  (69, 16, 35, 8, 24, 86, 57, 67, 77, 85, 93, 57...  51.933333
     15503  (8, 16, 35, 8, 24, 86, 57, 67, 77, 85, 93, 57,...  47.866667

     [15504 rows x 2 columns]
```



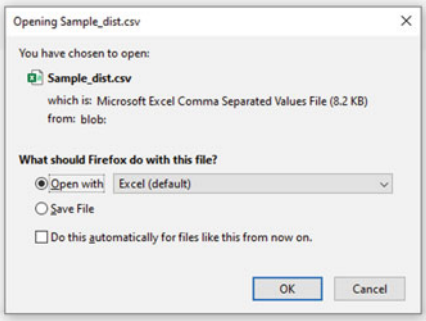

```
#Save combination list to your download folder
frame_data.to_csv('Combination.csv')
files.download('Combination.csv')
print("Finish saving combination list to your download folder")

Finish saving combination list to your download folder
```

```
[36] #print sample distribution
     frame_data = First.sample_dist_result()
     print(frame_data)

     Sampling distribution Probability
0                36.400000       0.01%
1                36.800000       0.01%
2                36.933333       0.01%
3                37.066667       0.01%
4                37.333333       0.01%
..                     ...         ...
349              61.066667       0.01%
350              61.133333       0.02%
351              61.200000       0.01%
352              61.666667       0.01%
353              61.733333       0.01%

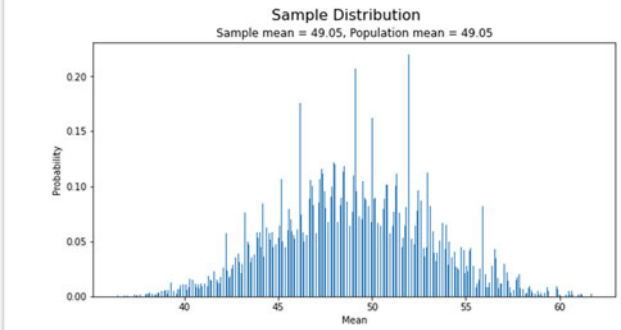
[354 rows x 2 columns]
```




```
#Save probablity distribution to computer download folder
frame_data.to_csv('Sample_dist.csv')
files.download('Sample_dist.csv')
print("Finish saving probablity list to computer 'Download' folder")

Finish saving probablity list to computer 'Download' folder
```

```
#graph sample distribution
First.Plot_hist()
```



8.5 Central Limit Theorem

In the above sections, we calculated the sample distribution. The sample distributions that we created in Sect. 8.2 were relatively simple to calculate. We saw that the sample distributions that we created in Sect. 8.3 were more complicated. In general, most sample distributions that we are interested in are very hard to calculate. Fortunately, because of the *central limit theorem*, we do not have to calculate sample distributions. The *central limit theorem* is stated below:

> As the sample size (n) from a given population gets 'large enough,' the sampling distribution of the mean, $\bar{x}$, can be approximated by a normal distribution with mean, μ, and standard deviation, $\sigma/\sqrt{n}$, regardless of the distribution of the individual values in the population.

Notice that the *central limit theorem* does not indicate what the sample size should be before it approximates a normal distribution. Some population distributions have sample distributions that approximate a normal distribution when the sample size is small, while other population distributions require a larger sample size before the sample distribution approximates a normal distribution. By convention, most people use a rule of thumb of having the sample size greater than 30 to guarantee that a sample distribution approximates a normal distribution.

To illustrate the *central limit theorem*, we will randomly select 300 samples for the distributions that we have studied. For each distribution, we will randomly select

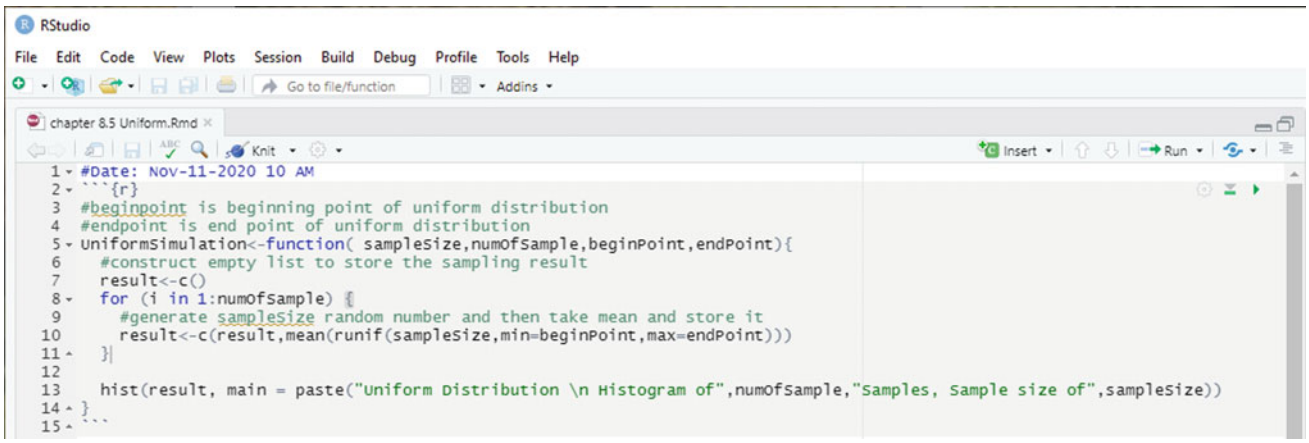
1. 300 samples of size 5.
2. 300 samples of size 10.
3. 300 samples of size 30.
4. 300 samples of size 50.

From this exercise, we will see that the sample mean distribution approximates a normal distribution for every distribution shown in this chapter. Some sample mean distribution will have an approximately normal distribution with a sample size as small as 5. In contrast, other sample mean distributions will approach a normal distribution as the sample size increases.

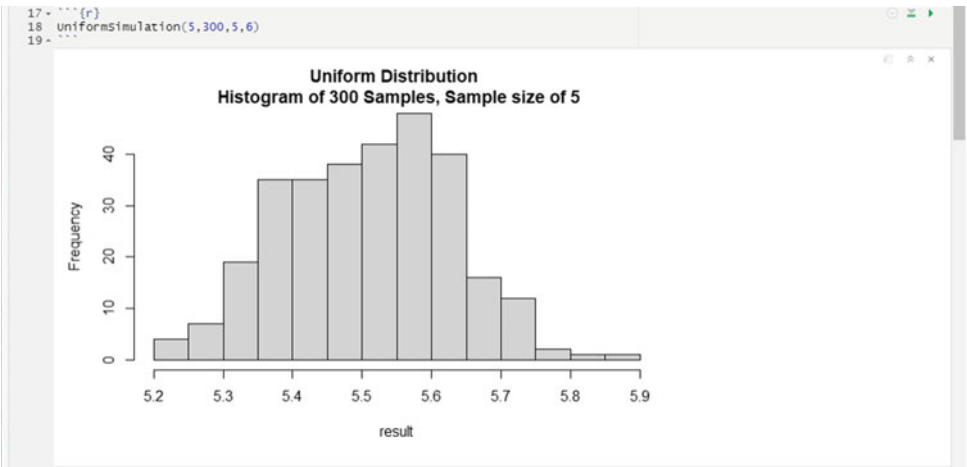
It should be noted that since these numbers are randomly generated, an exact reproduction of the histogram is not possible, but the general behavior of each histogram shown below should result.

8.5.1 Uniform Distribution in *R*

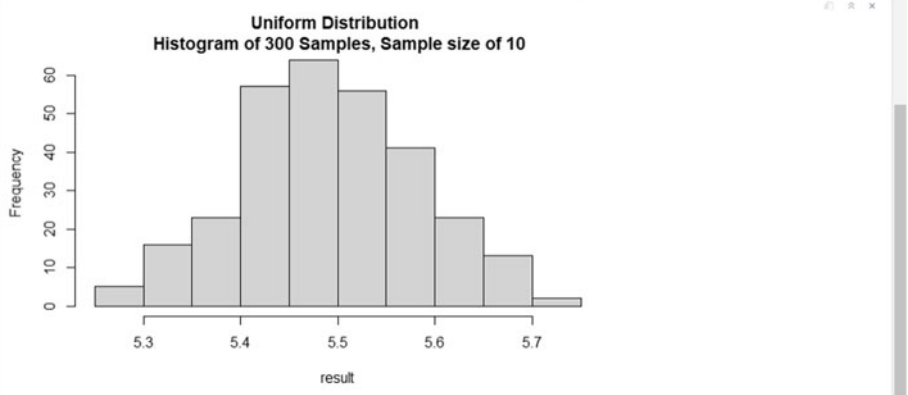
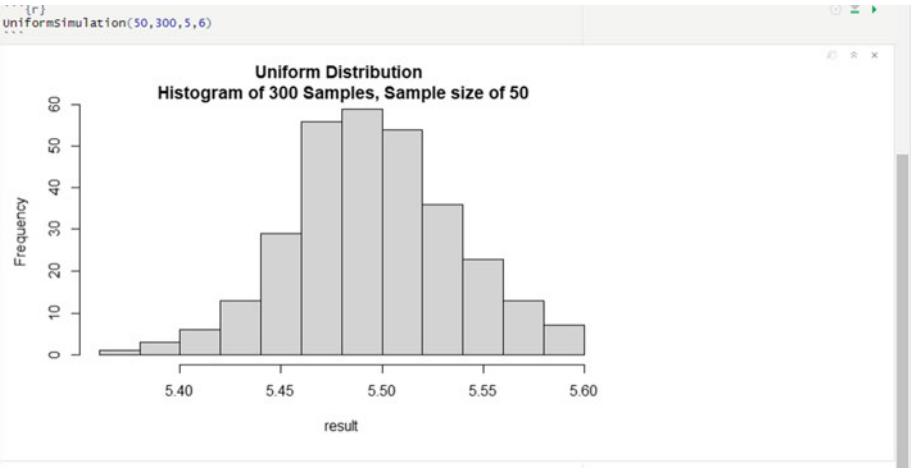
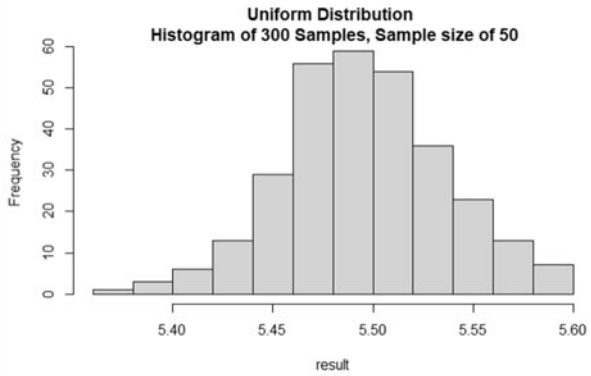
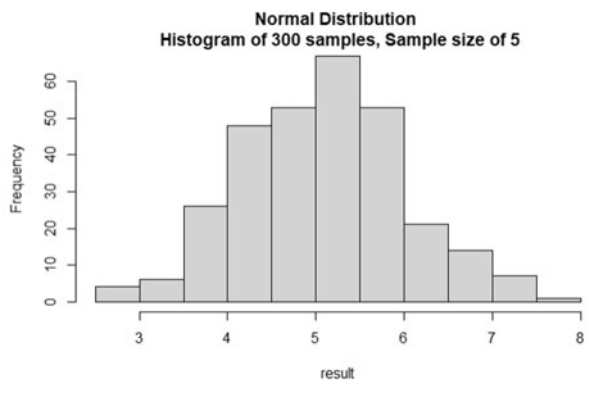
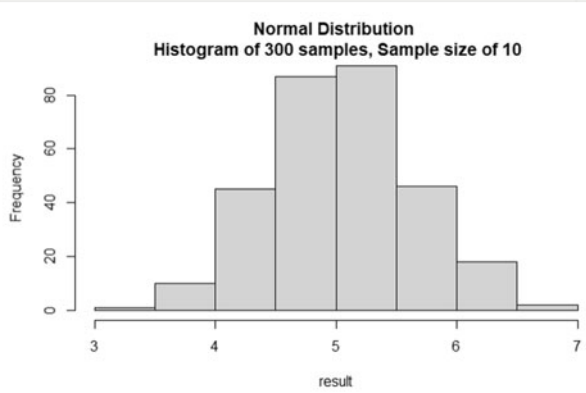
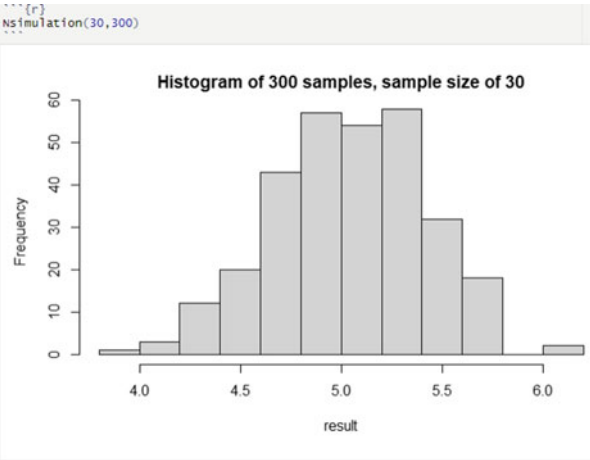
The *Uniformsimulation* function below generates the uniform distribution. The parameter *beginPoint* is the beginning point of the uniform distribution. The parameter *endpoint* is the end point of the Uniform distribution.

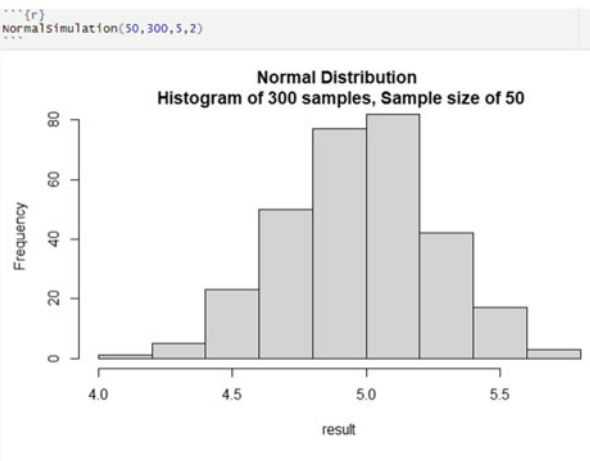


8.5.1.1 Sample Size = 5, 300 Samples



8.5.1.2 Sample Size = 10, 300 Samples

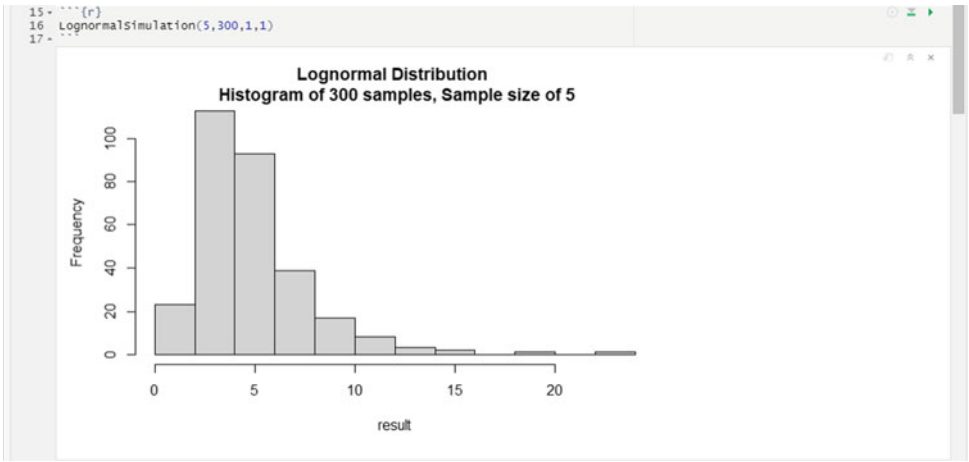
```
```{r}
UniformSimulation(10,300,5,6)
```
```



8.5.1.3 Sample Size = 30, 300 Samples

```
```{r}
UniformSimulation(50,300,5,6)
```
```



8.5.1.4 Sample Size = 50, 300 Samples

```
```{r}
uniformSimulation(50,300,5,6)
```
```



8.5.2 Normal Distribution in *R*

The above *NormalSimulation* function generates the normal distribution. The parameter *mean* is the mean of the normal distribution. The parameter *standardDeviation* is the standard deviation of the normal distribution.

8.5.2.1 Sample Size = 5, 300 Samples

```
```{r}
NormalSimulation(5,300,5,2)
```
```



8.5.2.2 Sample Size = 10, 300 Samples

```
```{r}
NormalSimulation(10,300,5,2)
```
```



8.5.2.3 Sample Size = 30, 300 Samples

```
```{r}
Nsimulation(30,300)
```
```



8.5.2.4 Sample Size = 50, 300 Samples

```
```{r}
NormalSimulation(50,300,5,2)
```
```



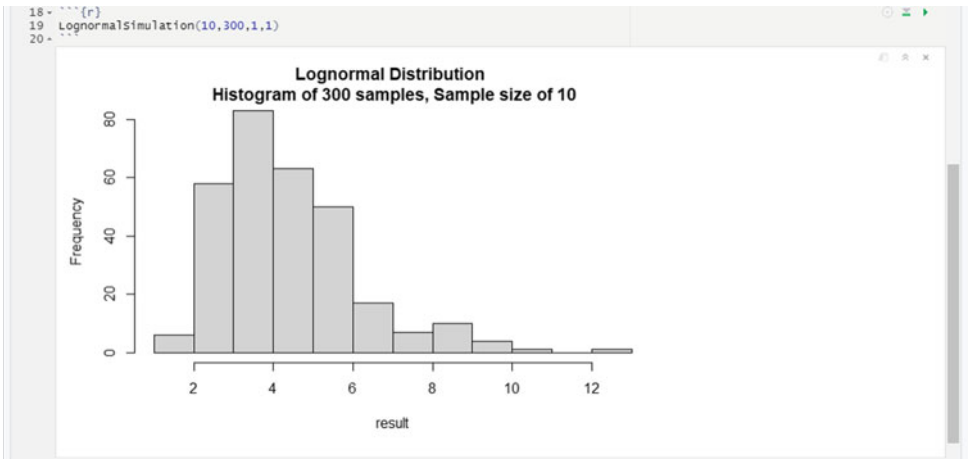
8.5.3 Lognormal Distribution in *R*

The above *LognormalSimulation* function generates the lognormal distribution. The parameter *mean* is the mean of the normal distribution. The parameter *standardDeviation* is the standard deviation of the *lognormal* distribution.

8.5.3.1 Sample Size = 5, 300 Samples

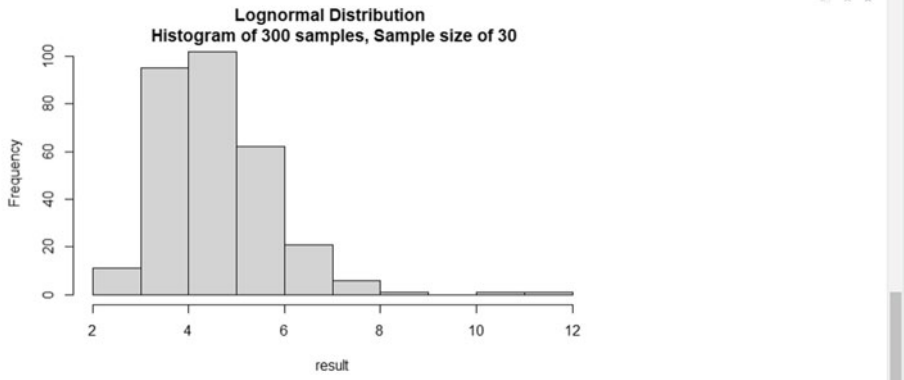


8.5.3.2 Sample Size = 10, 300 Samples

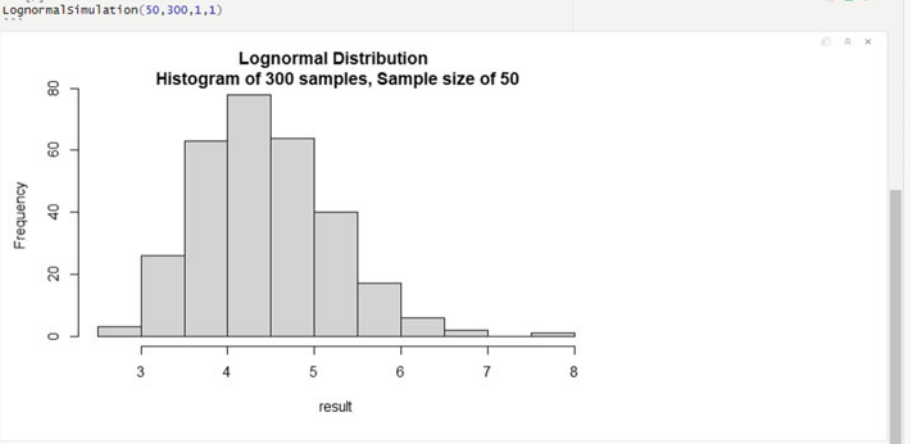
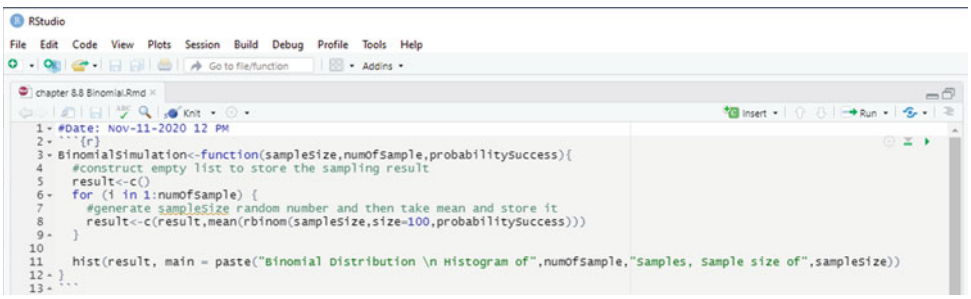
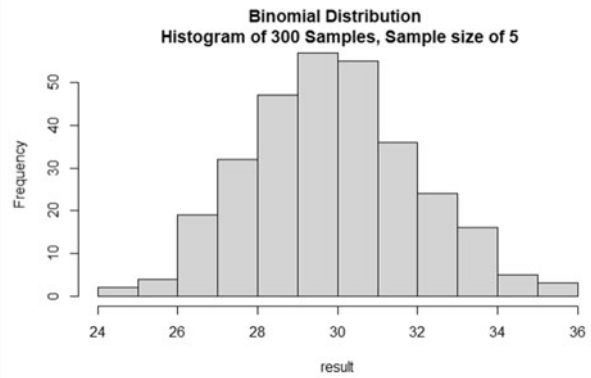
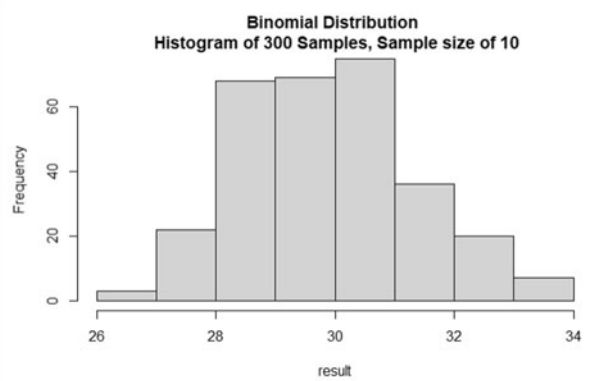
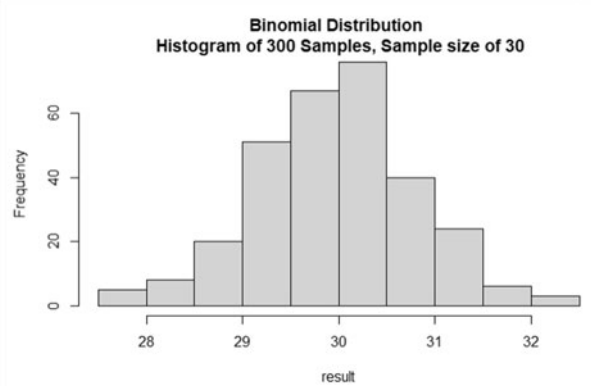
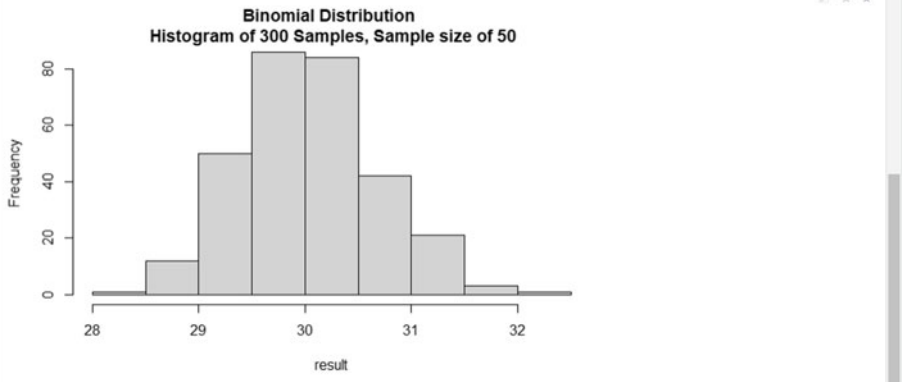
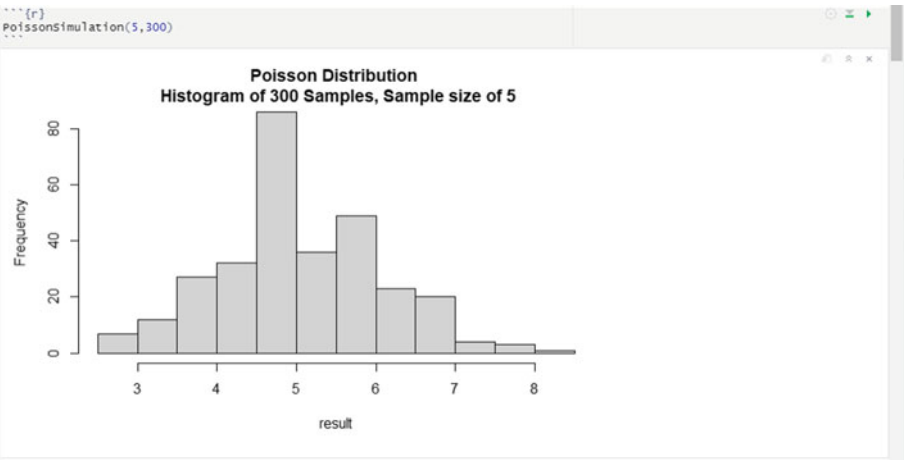
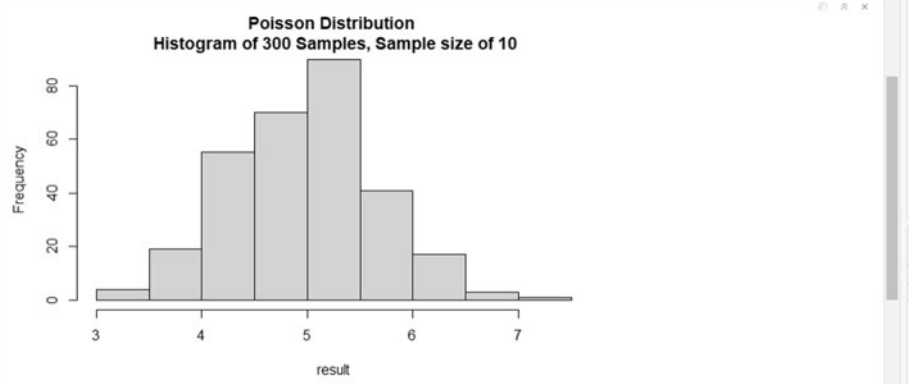


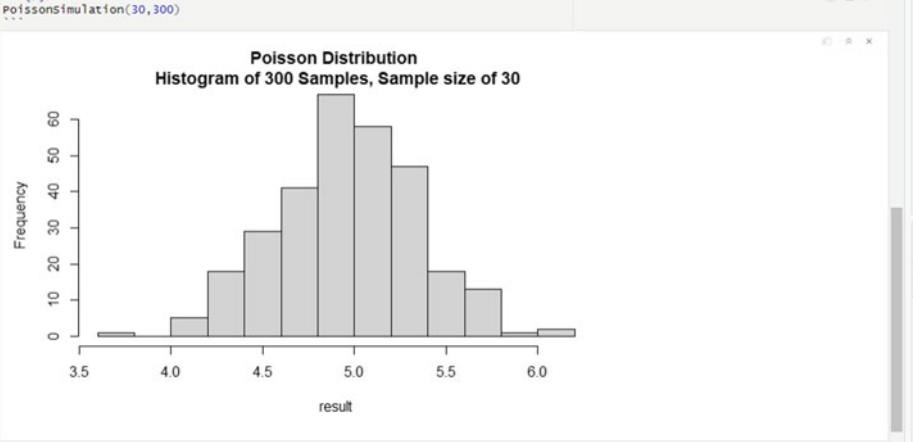
8.5.3.3 Sample Size = 30, 300 Samples

```r
```{r}
LognormalSimulation(30,300,1,1)
```
```

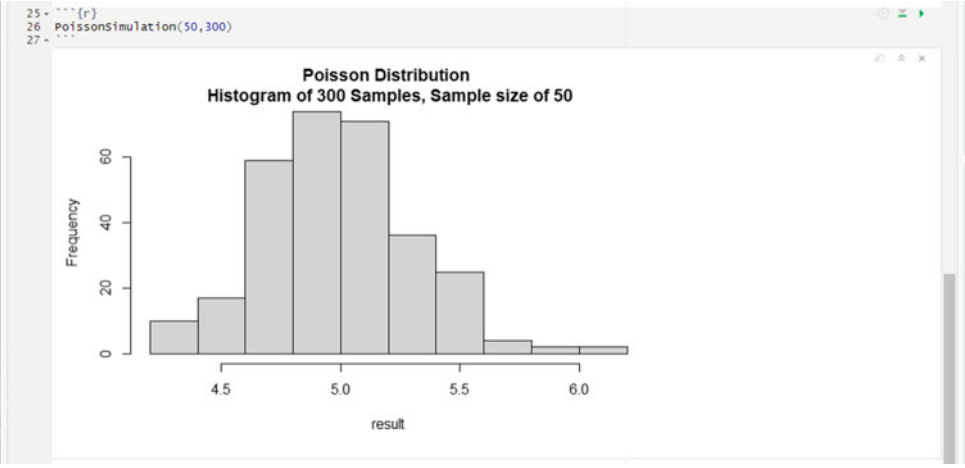


8.5.3.4 Sample Size = 50, 300 Samples

```r
```{r}
LognormalSimulation(50,300,1,1)
```
```



8.5.4 Binomial Distribution in *R*



The above *BinomialSimulation* function generates the binomial distribution. The parameter *probabilitySuccess* is the probability of success of the *binomial* distribution.

8.5.4.1 Sample Size = 5, 300 Samples

```
```{r}
BinomialSimulation(5,300,.3)
```
```



8.5.4.2 Sample Size = 10, 300 Samples

```
```{r}
BinomialSimulation(10,300,.3)
```
```



8.5.4.3 Sample Size = 30, 300 Samples

```
```{r}
BinomialSimulation(30,300,.3)
```
```



8.5.4.4 Sample Size = 50, 300 Samples

```
```{r}
BinomialSimulation(50,300,.3)
```
```



8.5.5 Poisson Distribution in *R*

The above *PoissonSimulation* function generates the Poisson distribution.

8.5.5.1 Sample Size = 5, 300 Samples

```
```{r}
PoissonSimulation(5,300)
```
```



8.5.5.2 Sample Size = 10, 300 Samples

```
```{r}
PoissonSimulation(10,300)
```
```



8.5.5.3 Sample Size = 30, 300 Samples

```
```{r}
PoissonSimulation(30,300)
```
```



8.5.5.4 Sample Size = 50, 300 Samples



8.6 Summary

In this chapter, we first calculated the sample distribution for a small population. We saw that once we calculated the sample distribution, we can make educated statements about the population. We noticed that as the sample size increases, the sample distribution approaches a normal distribution.

Most of the time, it is very hard to calculate the sample distribution for a specific population distribution. So instead of calculating the sample distribution, we can derive the sample distribution by randomly sampling a population distribution many times. We saw that the resulting sample distribution is approximately a normal distribution if the sample size is greater than 30.

We saw that we had doubts about the normality of a sample distribution from a sample size 5 from a lognormal distribution. We saw that the sample distribution of a lognormal distribution was approximately normal when the sample size became 10. We saw that the sample distribution of a uniform distribution was normal when the size of the sample distribution was 5.

We used *R* to demonstrate the central limit theorem.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O’Reilly Media, 2017.

Other Continuous Distributions

9

9.1 Introduction

We will now look at other continuous distributions that are commonly used in statistics. It is important to study all these distributions because later we will make statistical inferences based on the particular distribution we are working with. We should remember that the area under the density curve of each of these distributions is equal to 1.

This chapter is organized into the following sections. Section 9.2 explores *t*-distribution, Sect. 9.3 discusses chi-square distribution, and Sect. 9.4 discusses F-distribution. Section 9.5 talks about exponential distribution, and finally summarizes the chapter.

9.2 *t*-Distribution

The *t*-distribution is a symmetric distribution with a mean of 0.

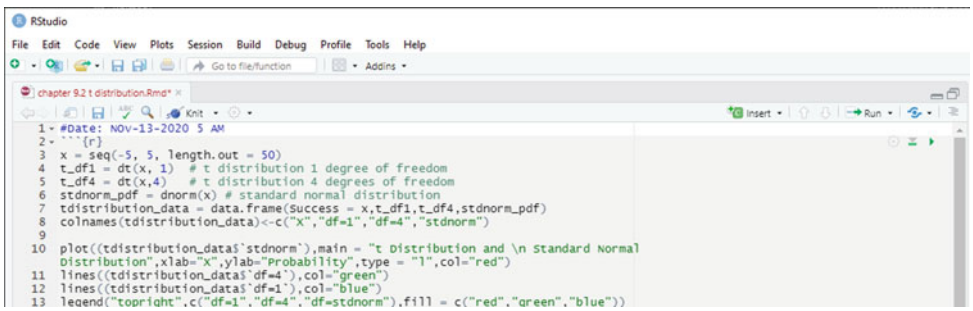
The mathematical formula for the *t*-distribution is

$$t = \frac{\overline{x} - \mu}{\frac{s_x}{\sqrt{n}}}$$

The *t*-distribution is influenced by the degrees of freedom. As the degrees of freedom increase, the *t*-distribution more closely resembles a standard normal distribution. In fact, the standard normal distribution is a *t*-distribution with an infinite number of degrees of freedom.

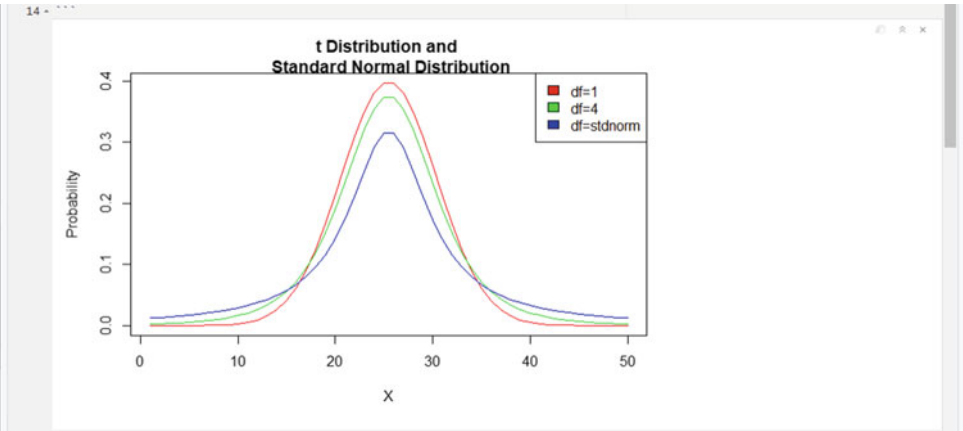
9.2.1 *t*-Distribution in *R*

The characteristics of the *t*-distribution are illustrated below. The *R* commands to produce the *t*-distribution and the standard normal distribution are given below:



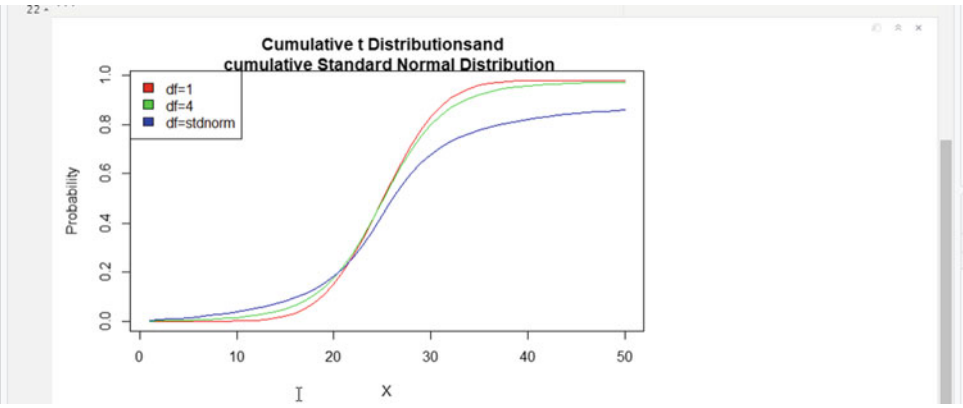
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_9



We will now calculate and plot the *t*-cumulative density function for both the one degree of freedom *t*-distribution and the four degrees of freedom *t*-distribution to illustrate that the total probability of a *t*-distribution equals 1.

```r
```{r}
gradSum = cumsum(tdistribution_data)/5
gradSum$Success <-x
plot((gradSum$`stdnorm`),main = "Cumulative t Distributionsand  \n cumulative Standard Normal
Distribution",xlab="X",ylab="Probability",type = "l",col="red")
lines((gradSum$`df=4`),col="green")
lines((gradSum$`df=1`),col="blue")
legend("topleft",c("df=1","df=4","df=stdnorm"),fill = c("red","green","blue"))
```
```



9.2.2 *t*-Distribution in *Python*

The *R* commands to produce the *t*-distribution and the standard normal distribution are discussed below:

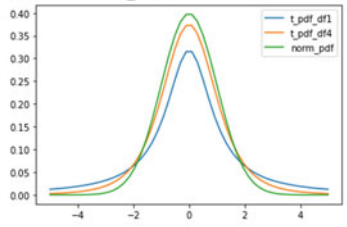
```python
[ ] from scipy.stats import t, norm
    import numpy as np
    import pandas as pd
```

```python
[ ] t_df1 = t.pdf(np.linspace(-5,5,50),df=1)
    t_df4 = t.pdf(np.linspace(-5,5,50),df=4)
    norm_pdf = norm.pdf(np.linspace(-5,5,50))
```

```python
[ ] # prepare dataframe
    pre_df = {'t_pdf_df1':t_df1,
              't_pdf_df4':t_df4,
              'norm_pdf':norm_pdf}
    df = pd.DataFrame(pre_df, index=np.linspace(-5,5,50))
```

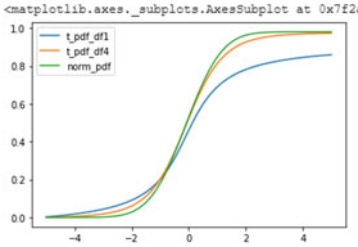
```
[ ] df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2a555f7dd8>
```



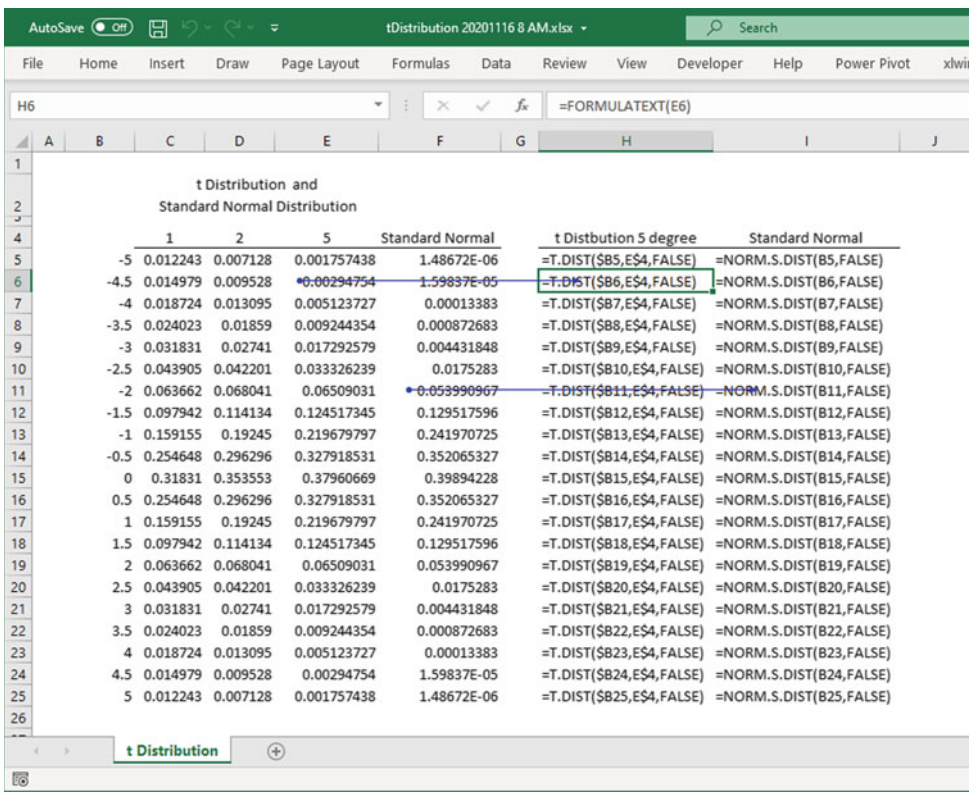

```
(df.cumsum()/5).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2a555f7978>
```



9.2.3 Student's *t*-Distribution in Excel

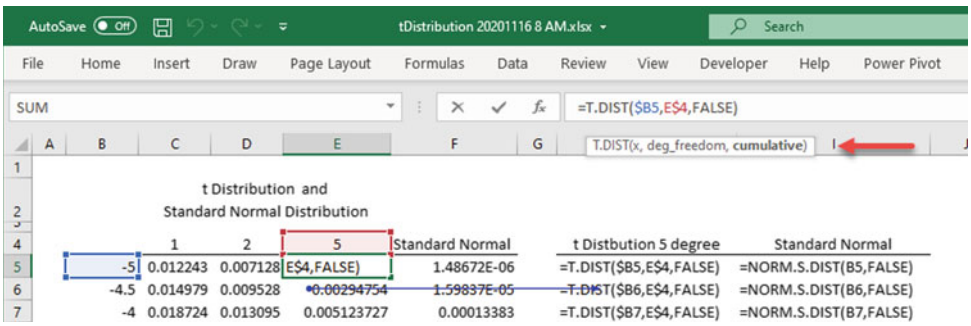
A *t*-distribution and standard normal distribution table are given below. Columns C, D, and E show a *t*-distribution with 1, 2, and 5 degrees of freedom, respectively. Column F shows the standard normal distribution, which has a mean of zero and a standard variance of 1. Column H shows the formula used to calculate the student's *t*-distribution in column E. Column I shows the formula used to calculate the standard normal distribution in column F.



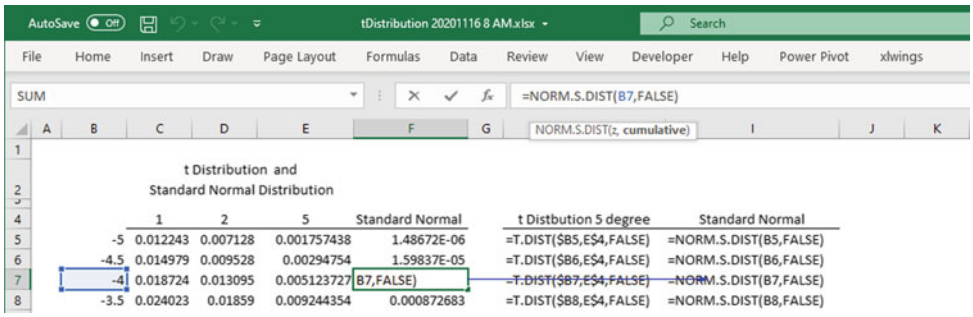
t Distribution and Standard Normal Distribution

	1	2	5	Standard Normal	t Distbution 5 degree	Standard Normal
-5	0.012243	0.007128	0.001757438	1.48672E-06	=T.DIST($B5,E$4,FALSE)	=NORM.S.DIST(B5,FALSE)
-4.5	0.014979	0.009528	0.00294754	1.59837E-05	=T.DIST($B6,E$4,FALSE)	=NORM.S.DIST(B6,FALSE)
-4	0.018724	0.013095	0.005123727	0.00013383	=T.DIST($B7,E$4,FALSE)	=NORM.S.DIST(B7,FALSE)
-3.5	0.024023	0.01859	0.009244354	0.000872683	=T.DIST($B8,E$4,FALSE)	=NORM.S.DIST(B8,FALSE)
-3	0.031831	0.02741	0.017292579	0.004431848	=T.DIST($B9,E$4,FALSE)	=NORM.S.DIST(B9,FALSE)
-2.5	0.043905	0.042201	0.033326239	0.0175283	=T.DIST($B10,E$4,FALSE)	=NORM.S.DIST(B10,FALSE)
-2	0.063662	0.068041	0.06509031	0.053990967	=T.DIST($B11,E$4,FALSE)	=NORM.S.DIST(B11,FALSE)
-1.5	0.097942	0.114134	0.124517345	0.129517596	=T.DIST($B12,E$4,FALSE)	=NORM.S.DIST(B12,FALSE)
-1	0.159155	0.19245	0.219679797	0.241970725	=T.DIST($B13,E$4,FALSE)	=NORM.S.DIST(B13,FALSE)
-0.5	0.254648	0.296296	0.327918531	0.352065327	=T.DIST($B14,E$4,FALSE)	=NORM.S.DIST(B14,FALSE)
0	0.31831	0.353553	0.37960669	0.39894228	=T.DIST($B15,E$4,FALSE)	=NORM.S.DIST(B15,FALSE)
0.5	0.254648	0.296296	0.327918531	0.352065327	=T.DIST($B16,E$4,FALSE)	=NORM.S.DIST(B16,FALSE)
1	0.159155	0.19245	0.219679797	0.241970725	=T.DIST($B17,E$4,FALSE)	=NORM.S.DIST(B17,FALSE)
1.5	0.097942	0.114134	0.124517345	0.129517596	=T.DIST($B18,E$4,FALSE)	=NORM.S.DIST(B18,FALSE)
2	0.063662	0.068041	0.06509031	0.053990967	=T.DIST($B19,E$4,FALSE)	=NORM.S.DIST(B19,FALSE)
2.5	0.043905	0.042201	0.033326239	0.0175283	=T.DIST($B20,E$4,FALSE)	=NORM.S.DIST(B20,FALSE)
3	0.031831	0.02741	0.017292579	0.004431848	=T.DIST($B21,E$4,FALSE)	=NORM.S.DIST(B21,FALSE)
3.5	0.024023	0.01859	0.009244354	0.000872683	=T.DIST($B22,E$4,FALSE)	=NORM.S.DIST(B22,FALSE)
4	0.018724	0.013095	0.005123727	0.00013383	=T.DIST($B23,E$4,FALSE)	=NORM.S.DIST(B23,FALSE)
4.5	0.014979	0.009528	0.00294754	1.59837E-05	=T.DIST($B24,E$4,FALSE)	=NORM.S.DIST(B24,FALSE)
5	0.012243	0.007128	0.001757438	1.48672E-06	=T.DIST($B25,E$4,FALSE)	=NORM.S.DIST(B25,FALSE)

The Excel function *T.DIST* calculates the student's *t*-distribution. The first parameter is the x-value. The second parameter is the degrees of freedom. The third parameter is to indicate if the function should calculate a cumulative *t*-distribution.

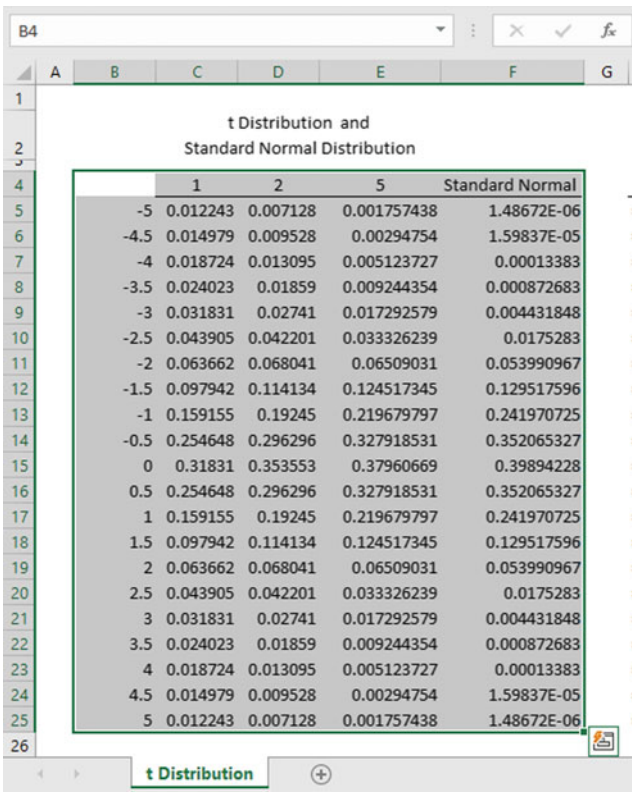


The Excel formula *NORM.S.DIST* calculates the standard normal distribution. The first parameter is the x-value. The second parameter is to indicate if the function should calculate a cumulative standard normal distribution.

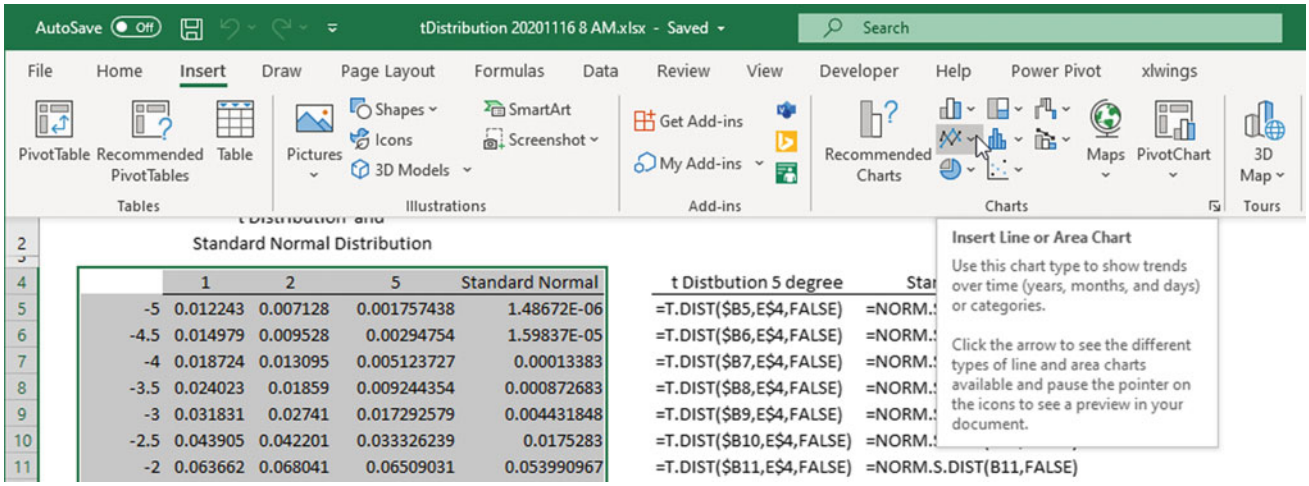


The following are the steps to graph the student's *t*-distribution and the standard normal distribution:

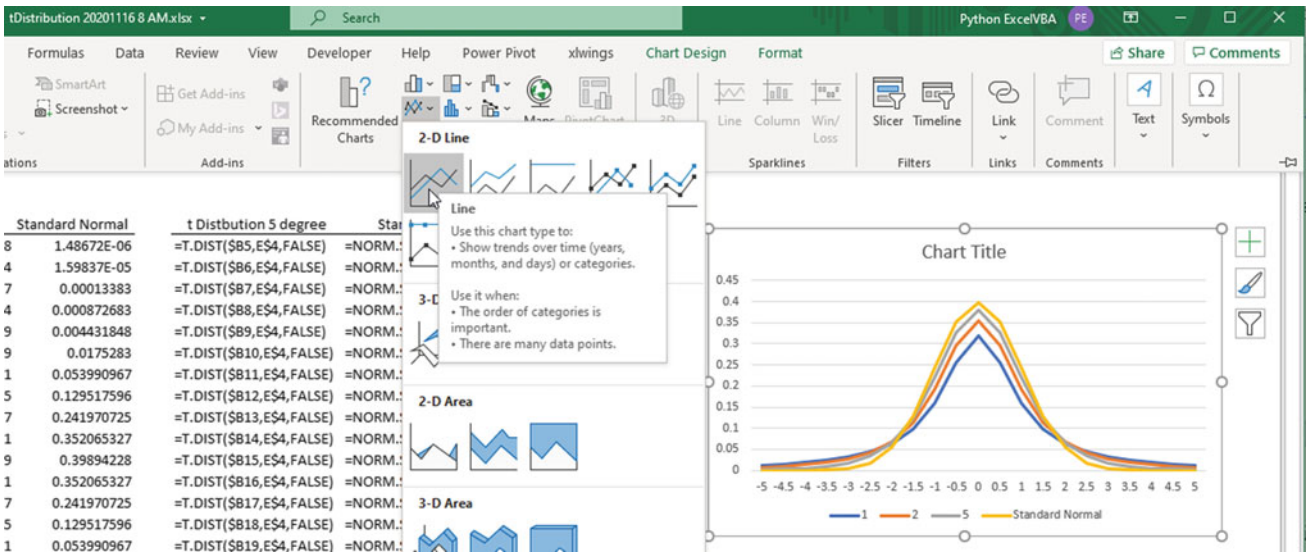
(1) Select cells B4 to F25.



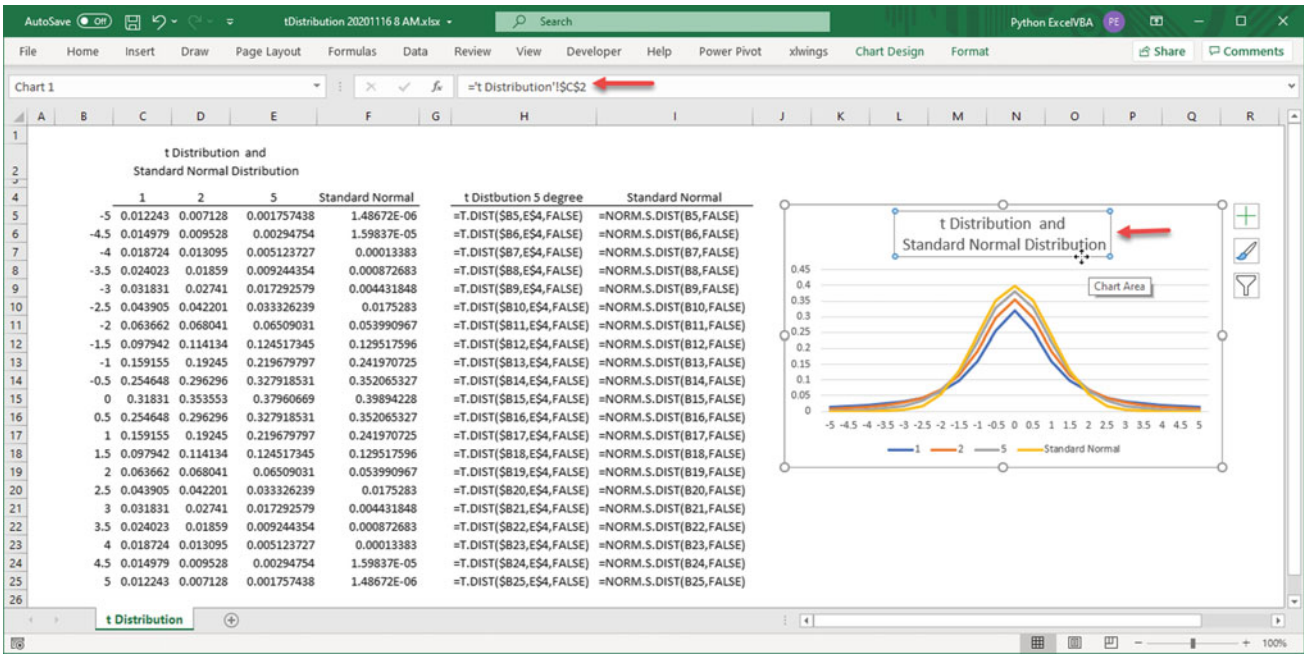
(2) Click on *Insert Line or Area Chart* from the *Insert* tab.



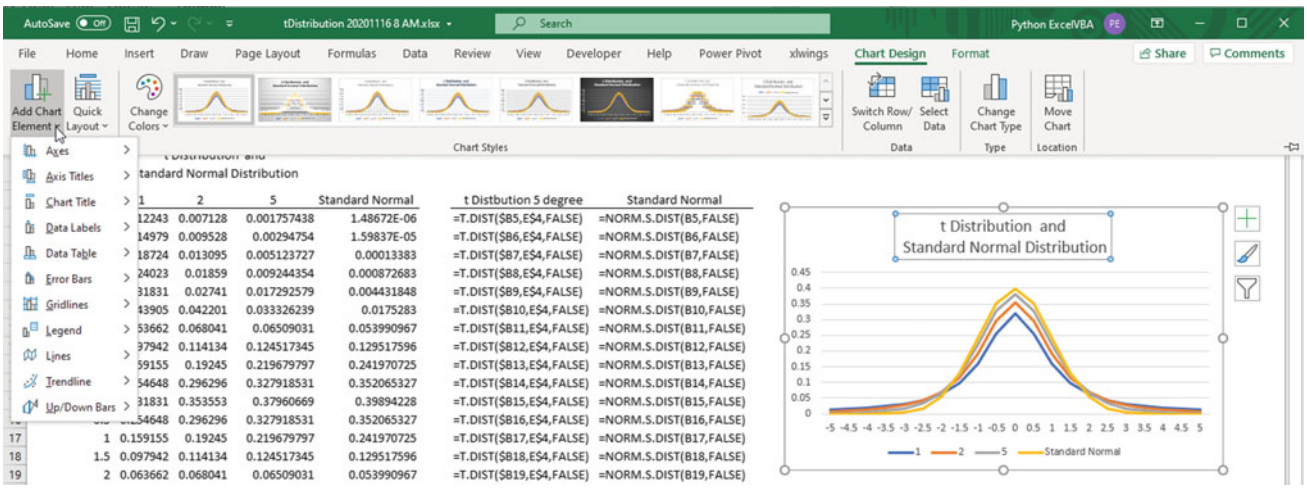
(3) Choose *Line* in the *2-D Line* row.



(4) Click on the borders of the chart title and in the formula bar, enter "= 't Distribution'!$C$2."



(5) With the chart selected, go to the *Add Chart Element* in the *Chart Design* tab to further format the chart.



9.3 Chi-Square (χ^2) Distribution

Another distribution often used in statistics is the chi-square (χ^2) distribution. The mathematical formula for the chi-square is

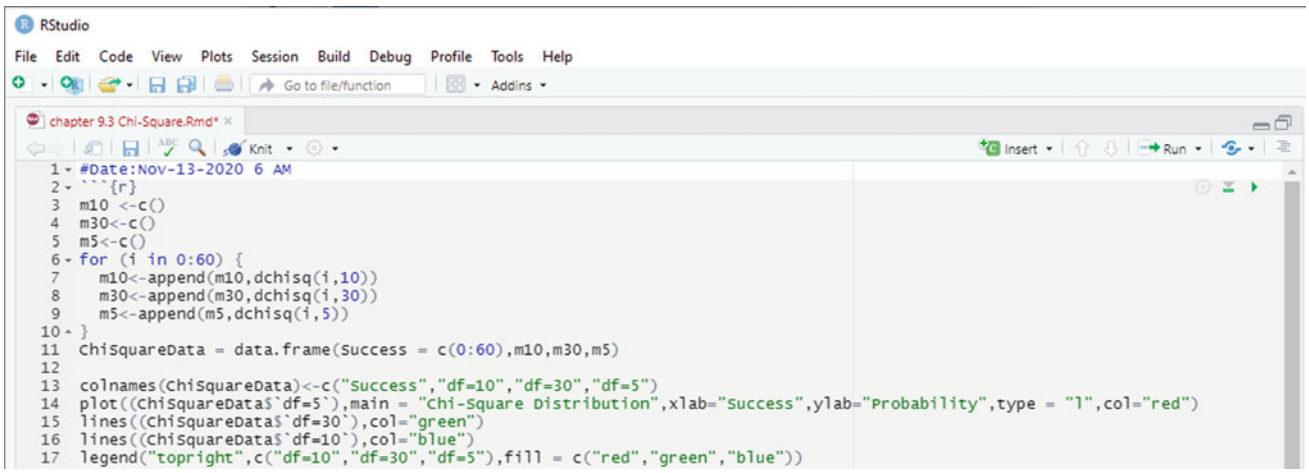
$$\chi^2 = \sum_{i=1}^{n} \left(\frac{X_i - \mu}{\sigma_X}\right)^2$$

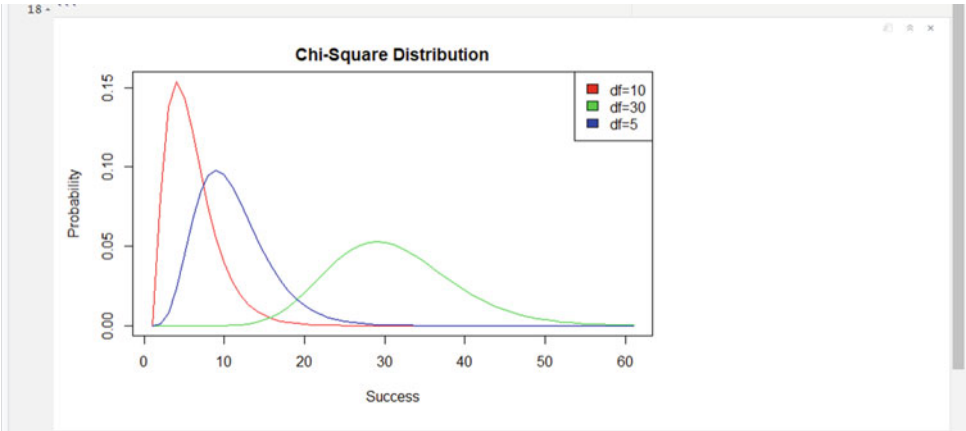
This probability density function is greatly influenced by the degrees of freedom.

The *R* commands below will show that as the degree of freedom increases, the distribution shifts to the right. We will create three chi-square distributions with 5, 10, and 30 degrees of freedom to show that the chi-square distribution does in fact shift to the right as the degree of freedom increases.

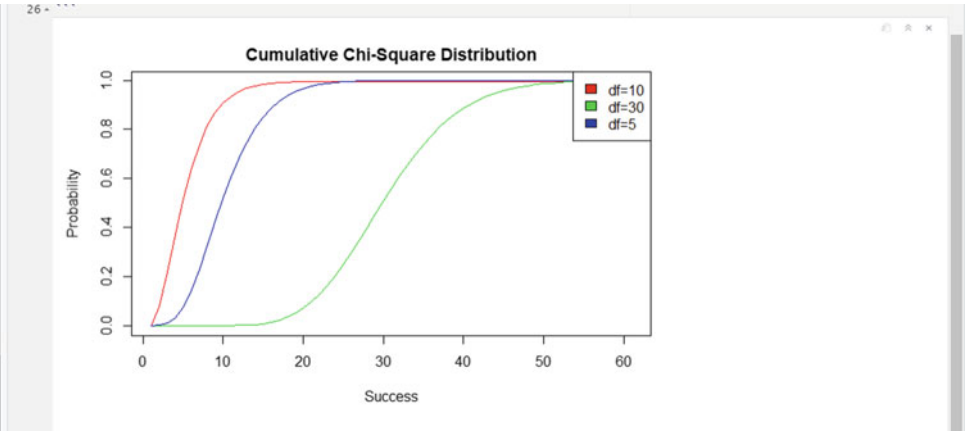
9.3.1 Chi-Square (χ^2) Distribution in *R*

The characteristics of the *Chi-Square* distribution are illustrated below in *R*:





The *R* commands to graph the cumulative chi-square distribution with degrees of freedom of 5, 10, and 30 are given below:



9.3.2 Chi-Square (χ^2) Distribution in *Python*

The characteristics of the *Chi-Square* distribution are illustrated below in *Python*:

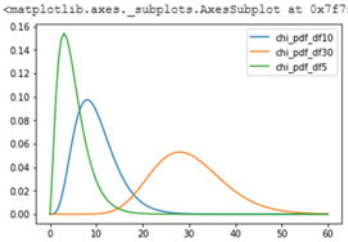
```
[ ] #Date: Apr-04-2021
    from scipy.stats import chi2
    import numpy as np
    import pandas as pd
```

```
[ ] chi_df10 = chi2.pdf(np.linspace(0,60,100),df=10)
    chi_df30 = chi2.pdf(np.linspace(0,60,100),df=30)
    chi_df5 = chi2.pdf(np.linspace(0,60,100),df=5)
```

```
[ ] # prepare dataframe
    pre_df = {'chi_pdf_df10':chi_df10,
              'chi_pdf_df30':chi_df30,
              'chi_pdf_df5':chi_df5}
    df = pd.DataFrame(pre_df, index=np.linspace(0,60,100))
```

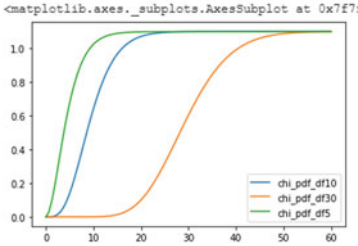
```
[ ] df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7fb98f8898>
```



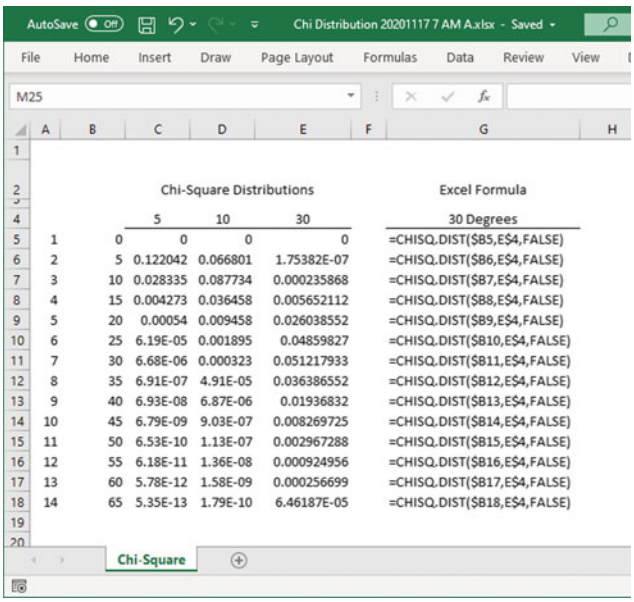

```
[ ] (df.cumsum()/1.5).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7fb7fbf518>
```

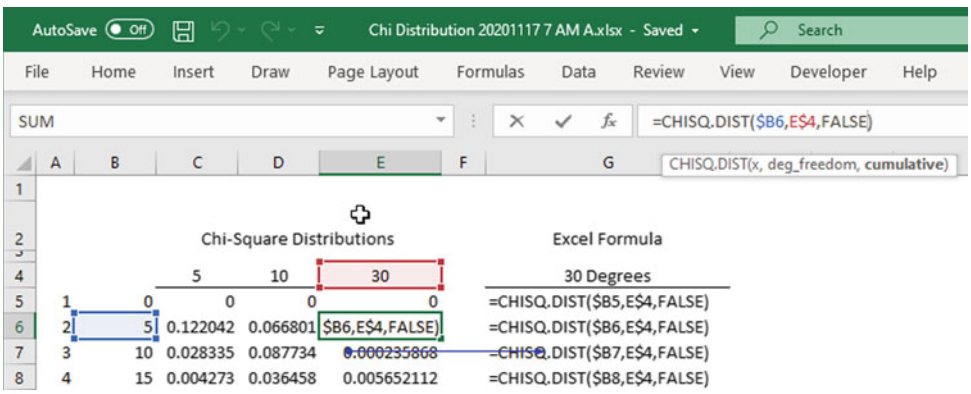


9.3.3 Chi-Square (χ^2) Distribution in Excel

A *chi-square* distribution table is given below. Columns C, D, and E show a *chi-square* distribution with 5, 10, and 30 degrees of freedom, respectively. Column G shows the formula used to calculate the *chi-square* distribution in column E.

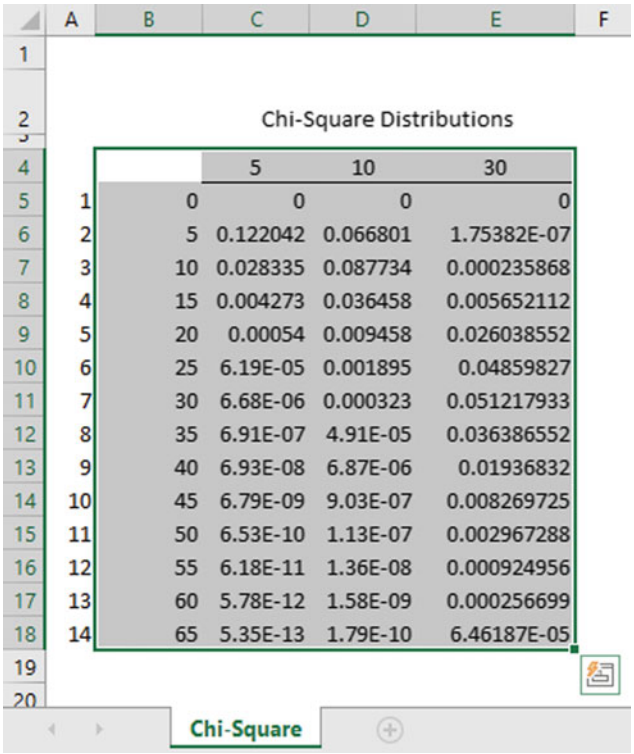


The Excel function *CHISQ.DIST* calculates the *chi-square* distribution. The first parameter is the x-value. The second parameter is the degrees of freedom. The third parameter is to indicate if the function should calculate a cumulative chi-square distribution.



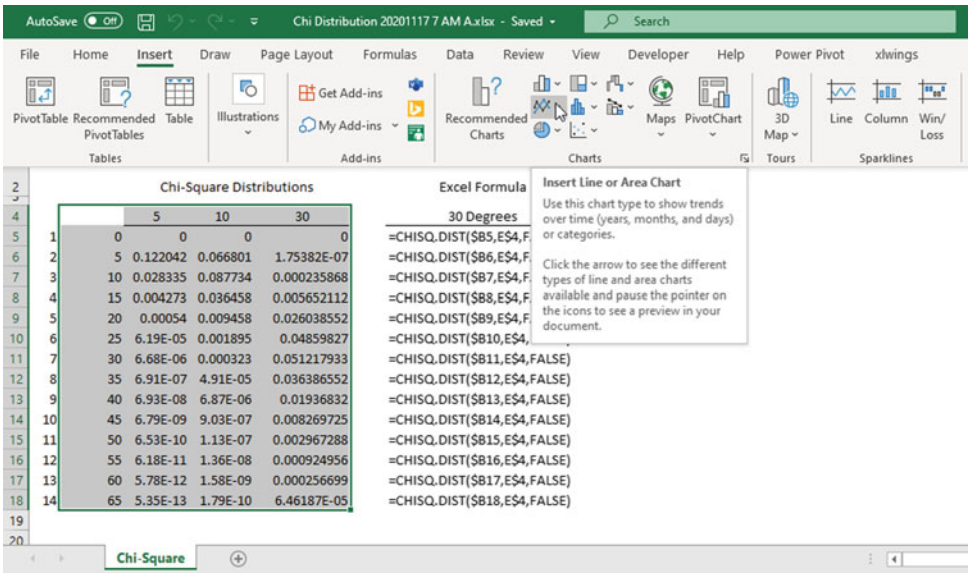
The following are the steps to graph the chi-square distribution:

(1) Select cells B4 to E18.

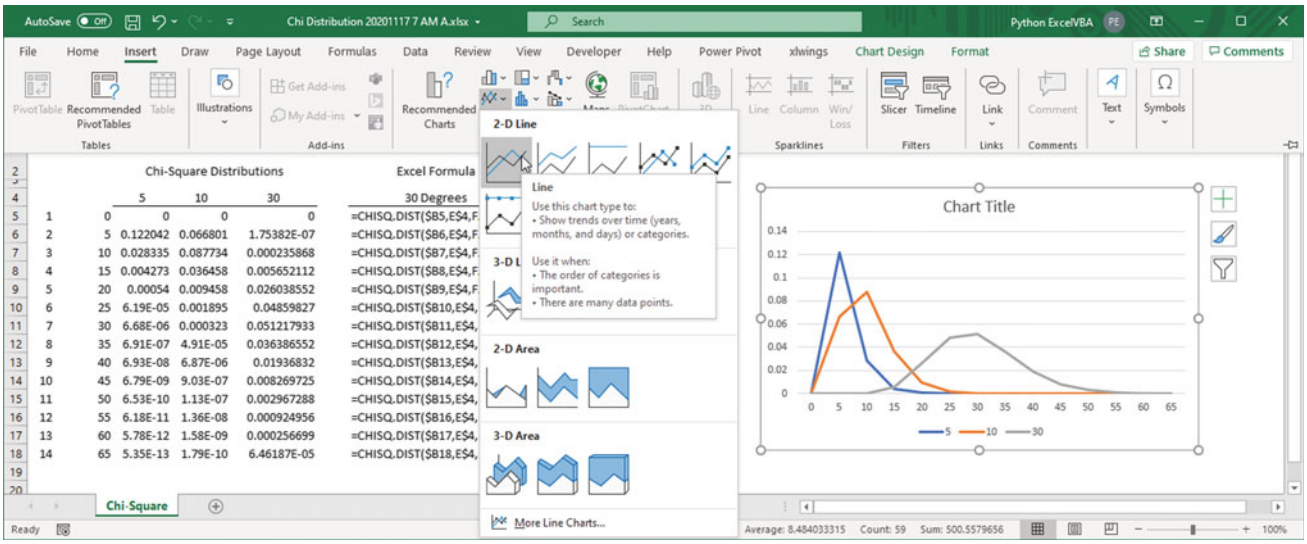


		5	10	30
1	0	0	0	0
2	5	0.122042	0.066801	1.75382E-07
3	10	0.028335	0.087734	0.000235868
4	15	0.004273	0.036458	0.005652112
5	20	0.00054	0.009458	0.026038552
6	25	6.19E-05	0.001895	0.04859827
7	30	6.68E-06	0.000323	0.051217933
8	35	6.91E-07	4.91E-05	0.036386552
9	40	6.93E-08	6.87E-06	0.01936832
10	45	6.79E-09	9.03E-07	0.008269725
11	50	6.53E-10	1.13E-07	0.002967288
12	55	6.18E-11	1.36E-08	0.000924956
13	60	5.78E-12	1.58E-09	0.000256699
14	65	5.35E-13	1.79E-10	6.46187E-05

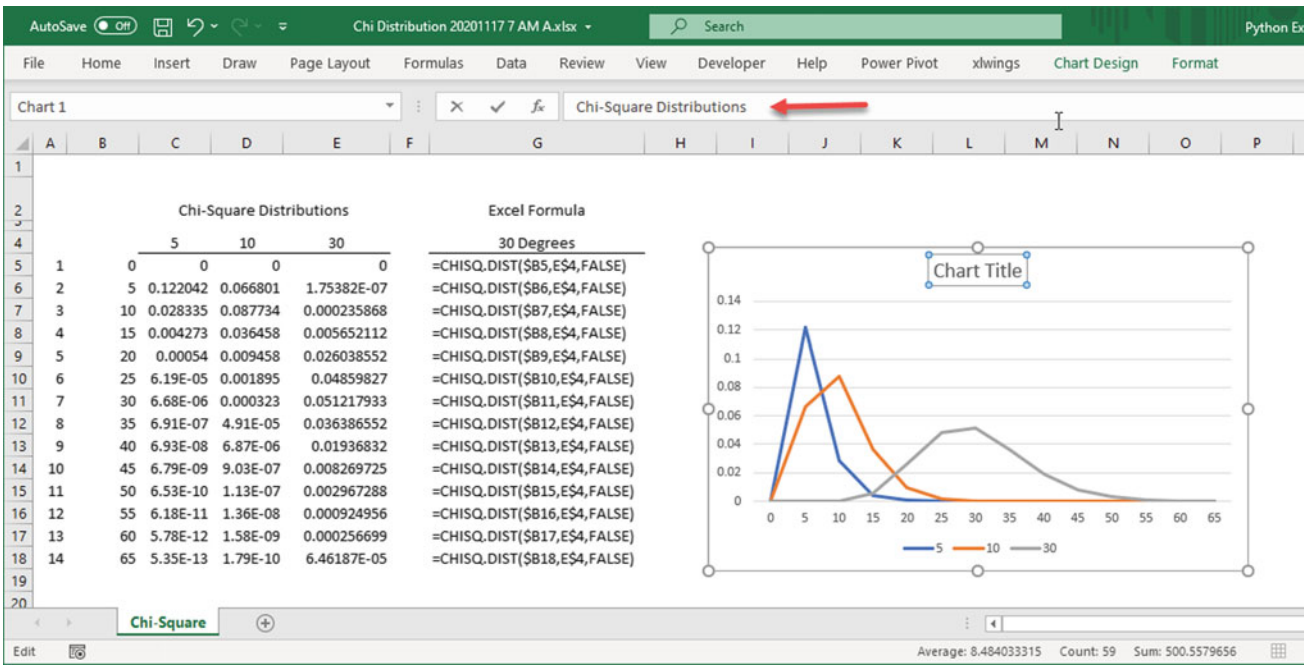
(2) Click on *Insert Line or Area Chart* from the *Insert* tab.



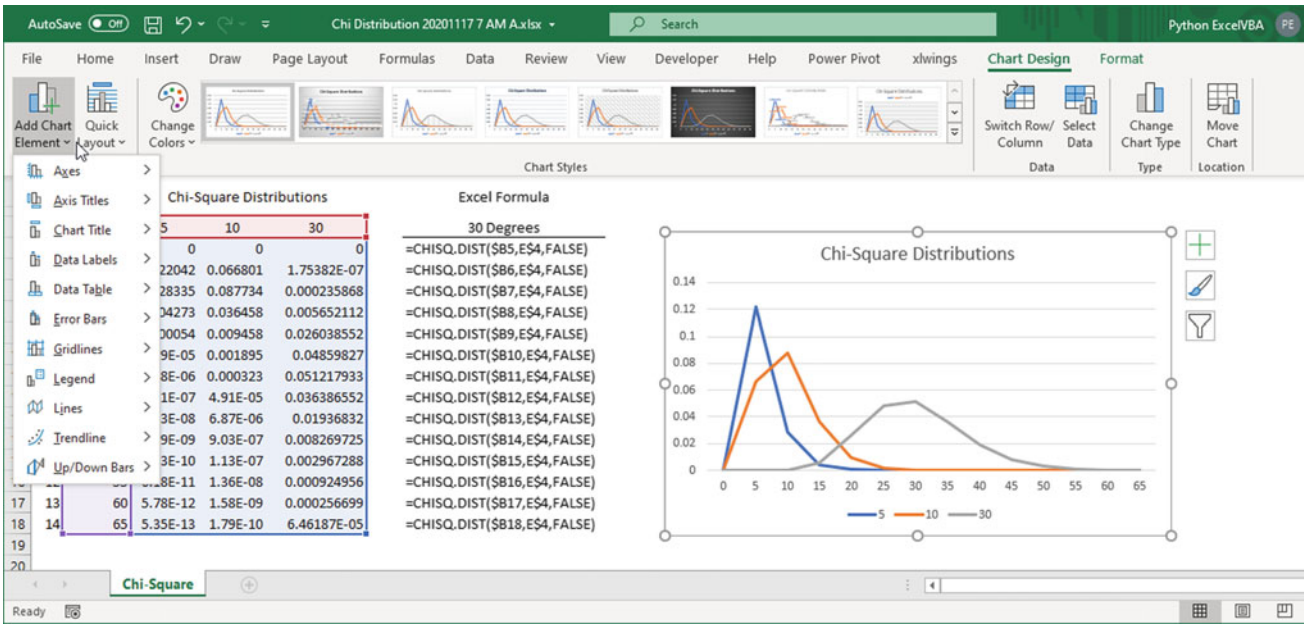
(3) Choose *Line* in the *2-D Line* row.



(4) Click on the borders of the chart title and in the formula bar, enter "Chi-Square."



(5) With the chart selected, go to the *Add Chart Element* in the *Chart Design* tab to further format the chart.



9.4 *F*-Distribution

Another distribution often used in statistics is the F-distribution. Its mathematical formula is

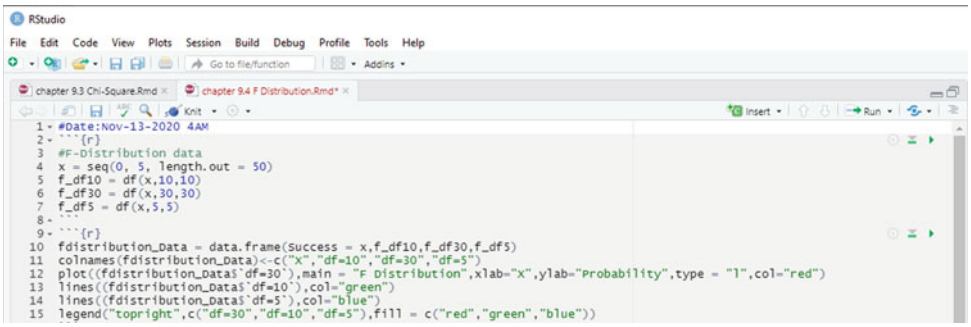
$$F = \frac{s_X^2/\sigma_X^2}{s_Y^2/\sigma_Y^2}$$

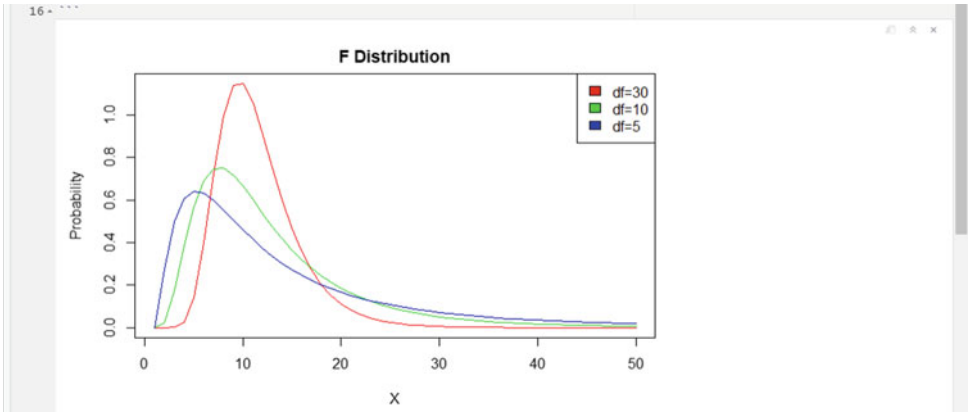
Each F-distribution is influenced by two different degrees of freedom. There is a degree of freedom for the numerator and a degree of freedom for the denominator of the formula.

The session commands below will show that as the two different degrees of freedom increase, the more the F-distribution shifts to the right and the more peaked the F-distribution becomes. We will generate and plot three F-distributions to illustrate this characteristic of the F-distribution.

9.4.1 *F*-Distribution in *R*

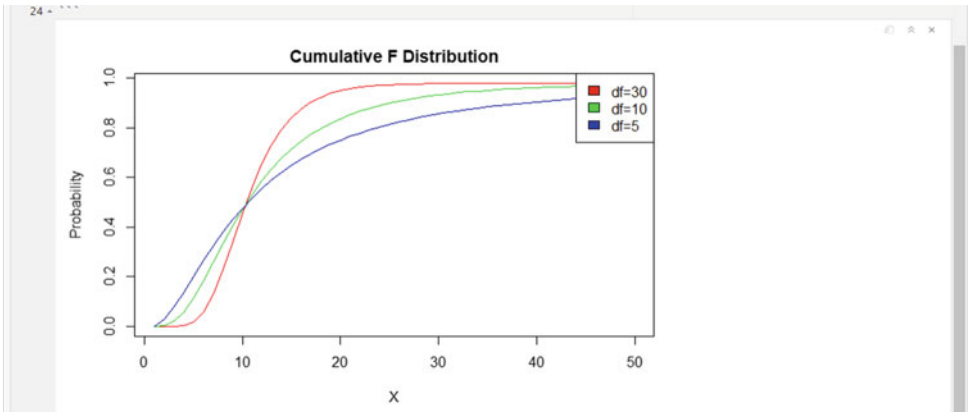
The characteristics of the *F-Distribution* distribution are illustrated below in *R*:





The commands to calculate the cumulative *F*-distribution for graphs with 5,5 and 10,10 and 30,30 degrees of freedom are given below:

```r
```{r}
gradSum = cumsum(fdistribution_Data)/10
gradSum$Success <-x
plot((gradSum$`df=30`),main = "Cumulative F Distribution",xlab="X",ylab="Probability",type = "l",col="red")
lines((gradSum$`df=10`),col="green")
lines((gradSum$`df=5`),col="blue")
legend("topright",c("df=30","df=10","df=5"),fill = c("red","green","blue"))
```
```



9.4.2 *F*-Distribution in *Python*

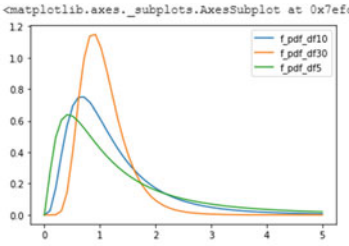
```
[ ]  #Date: Apr-04-2021
     from scipy.stats import f
     import numpy as np
     import pandas as pd

[ ]  f_df10 = f.pdf(np.linspace(0,5,50),dfn=10, dfd=10)
     f_df30 = f.pdf(np.linspace(0,5,50),dfn=30, dfd=30)
     f_df5 = f.pdf(np.linspace(0,5,50),dfn=5, dfd=5)

[ ]  # prepare dataframe
     pre_df = {'f_pdf_df10':f_df10,
               'f_pdf_df30':f_df30,
               'f_pdf_df5':f_df5}
     df = pd.DataFrame(pre_df, index=np.linspace(0,5,50))
```

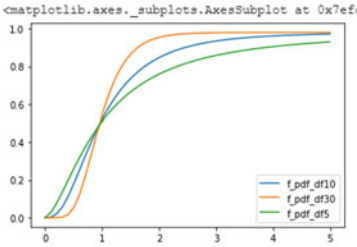
```
[ ] df.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efcfd6b9390>
```



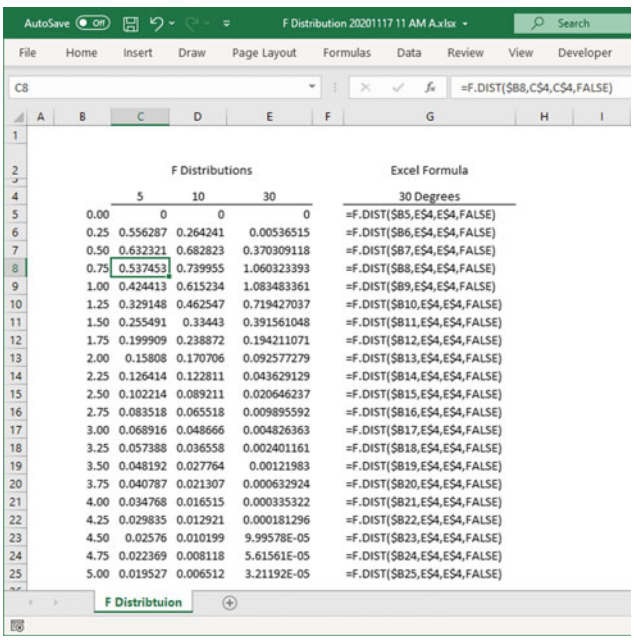

```
[ ] (df.cumsum()/10).plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efcfd668860>
```

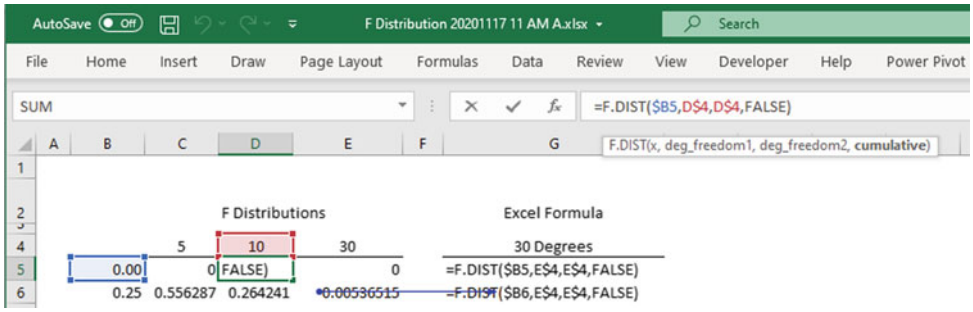


9.4.3 *F*-Distribution in Excel

An *F*-distribution table in Excel is shown below. Columns C, D, and E show an *F*-distribution with "5,5," "10,10," and "30,30" degrees of freedom, respectively. Column G shows the formula used to calculate the *F*-distribution in column E.

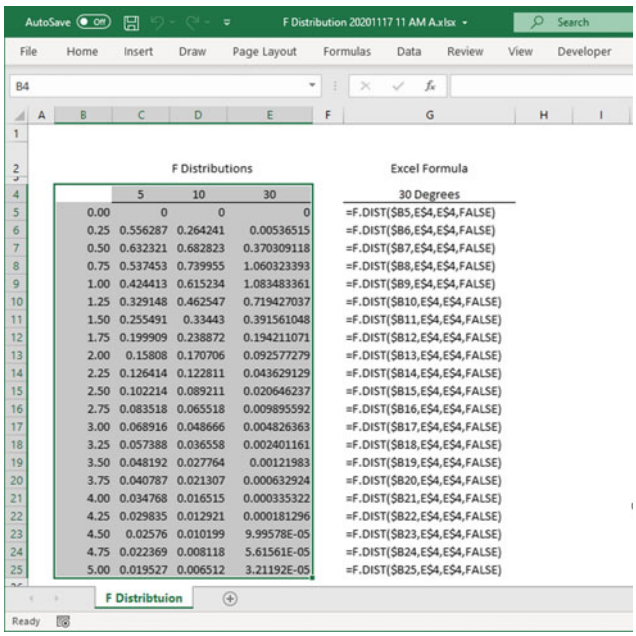


The Excel function *F.DIST* calculates the *F*-distribution. The first parameter is the x-value. The second parameter is the first degree of freedom. The third parameter is the second degree of freedom. The fourth parameter is to indicate if the function should calculate a cumulative *F*-distribution.

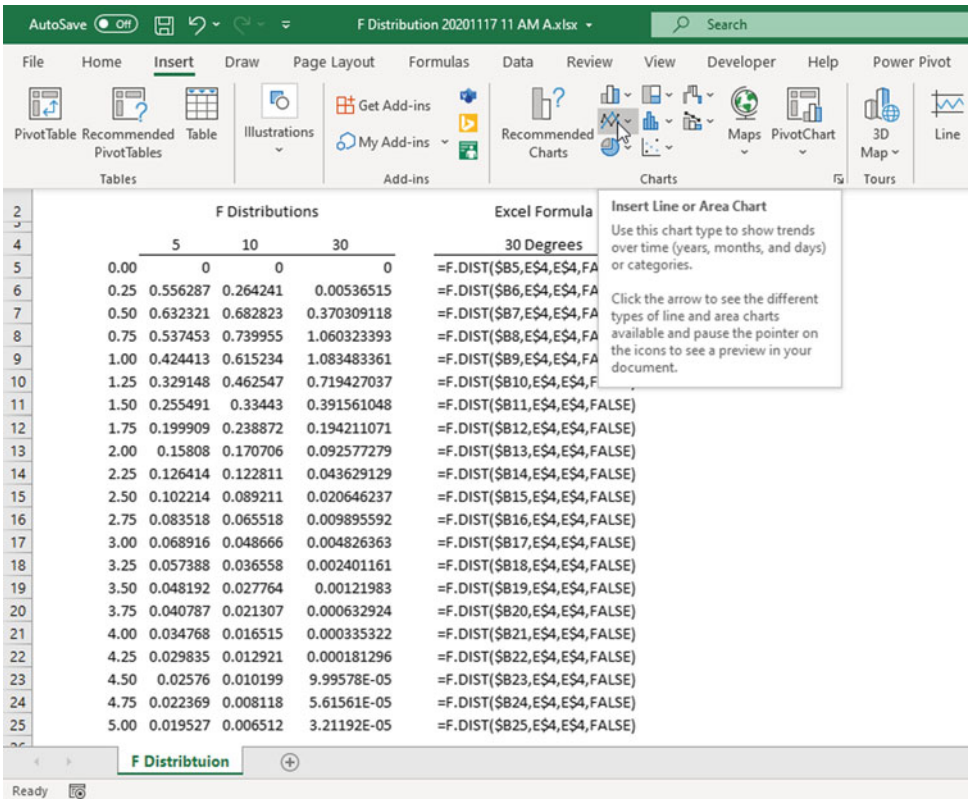


The following are the steps to graph the *F*-distribution:

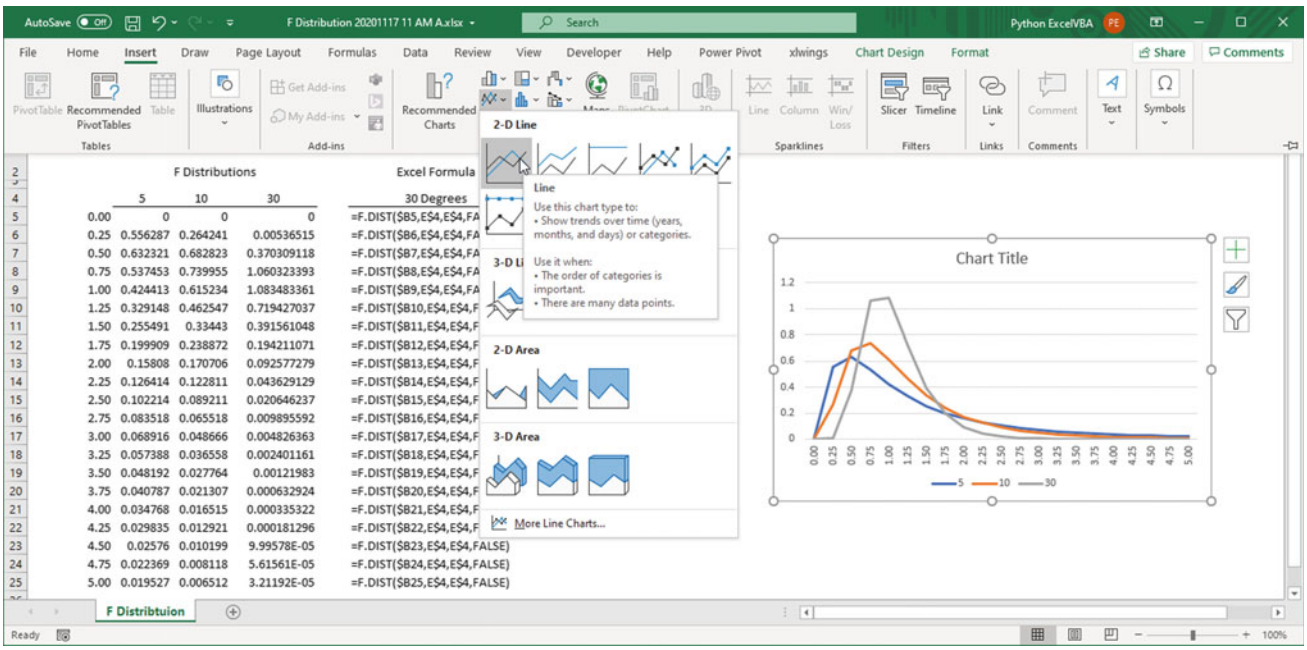
(1) Select cells B4 to E25.



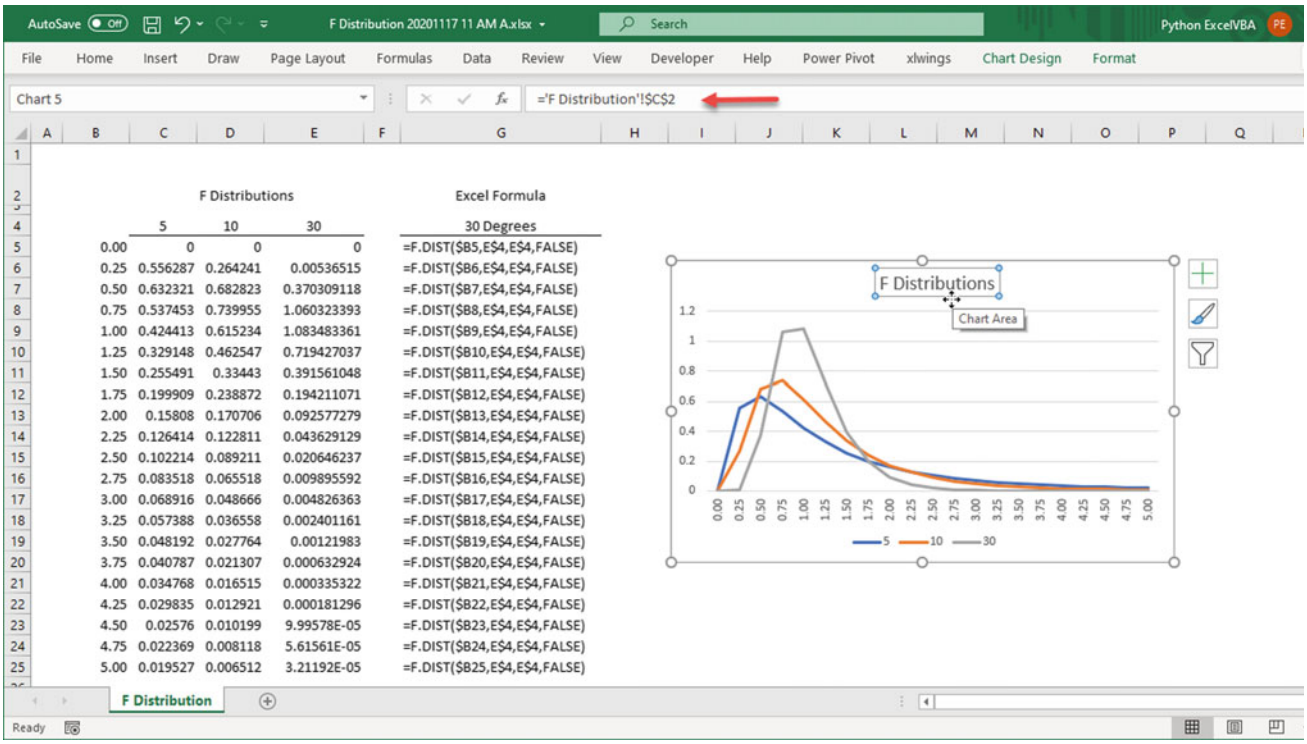
(2) Click on *Insert Line or Area Chart* from the *Insert* tab.



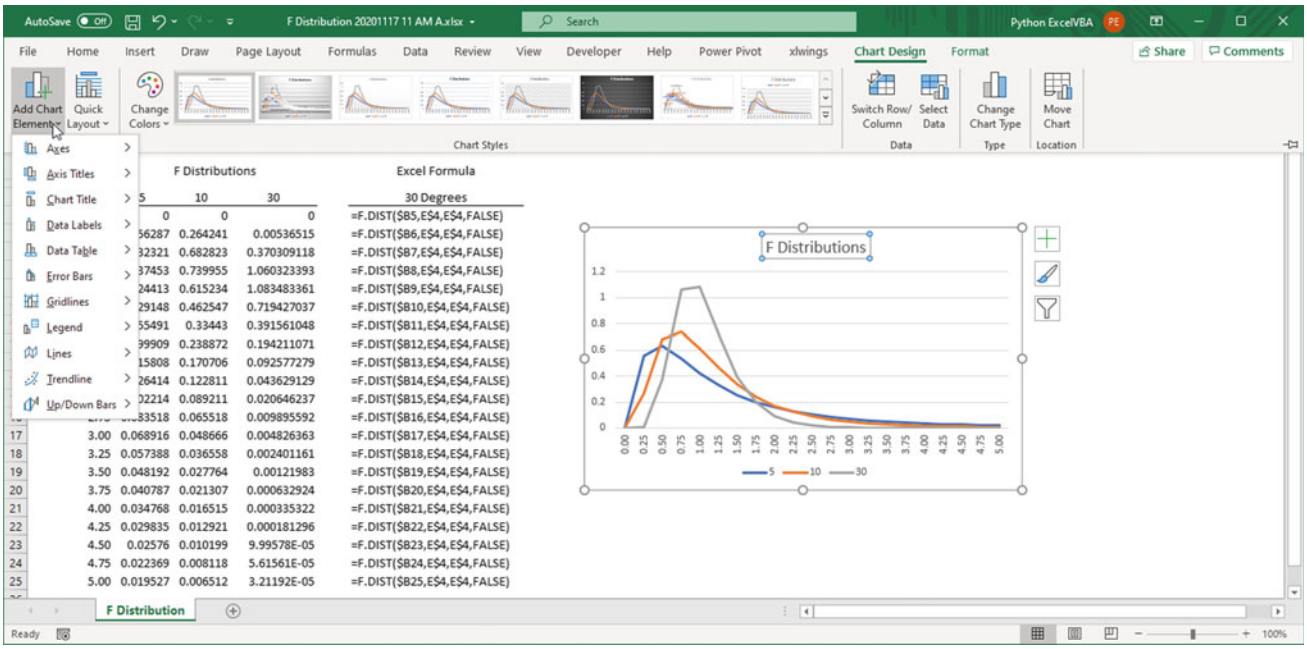
(3) Choose *Line* in the *2-D Line* row.



(4) Click on the borders of the chart title and in the formula bar, enter "= 'F Distribution."



(5) With the chart selected, go to the *Add Chart Element* in the *Chart Design* tab to further format the chart.



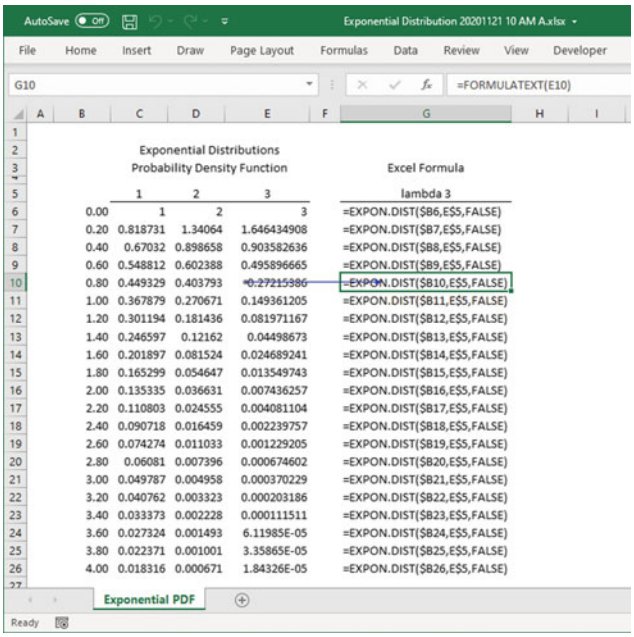
9.5 Exponential Distribution

We will now look at the exponential distribution. The probability density function of the exponential distribution is

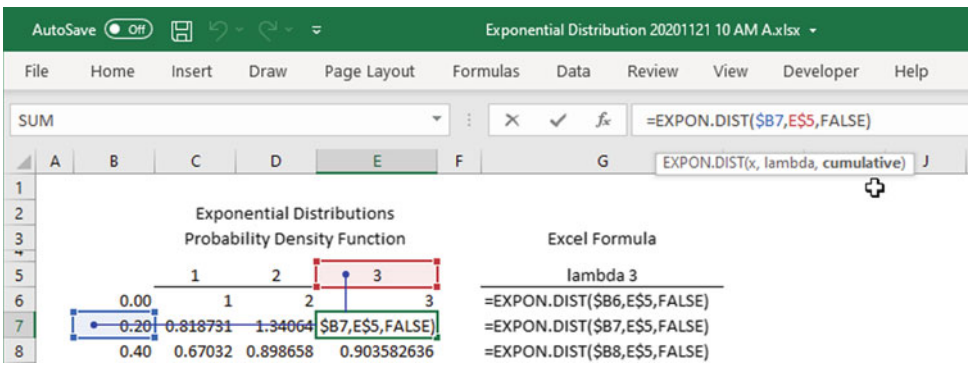
$$f(x) = \lambda e^{-\lambda x} \quad x > 0$$

9.5.1 Exponential Probability Density Function in Excel

An exponential probability density function table in Excel is shown below. Columns C, D, and E show an *exponential* distribution with lambda of 1, 2, and 3, respectively. Column G shows the formula used to calculate the *exponential* distribution in column E.

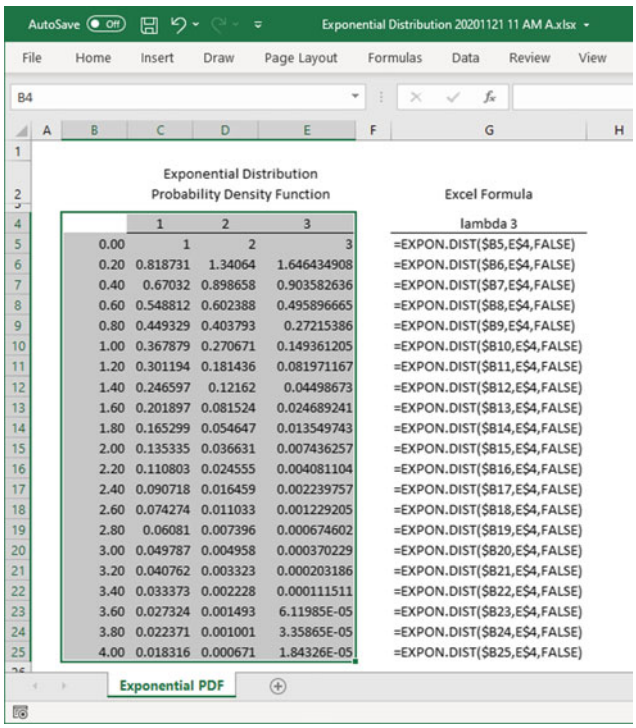


The Excel function *EXPON.DIST* calculates the *exponential* distribution. The first parameter is the x-value. The second parameter is the lambda. The third parameter is to indicate if the function should calculate a cumulative *exponential* distribution.

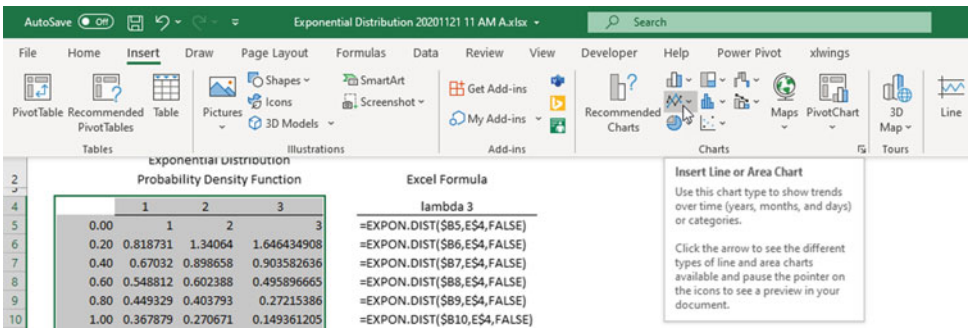


The following are the steps to graph the *exponential* distribution:

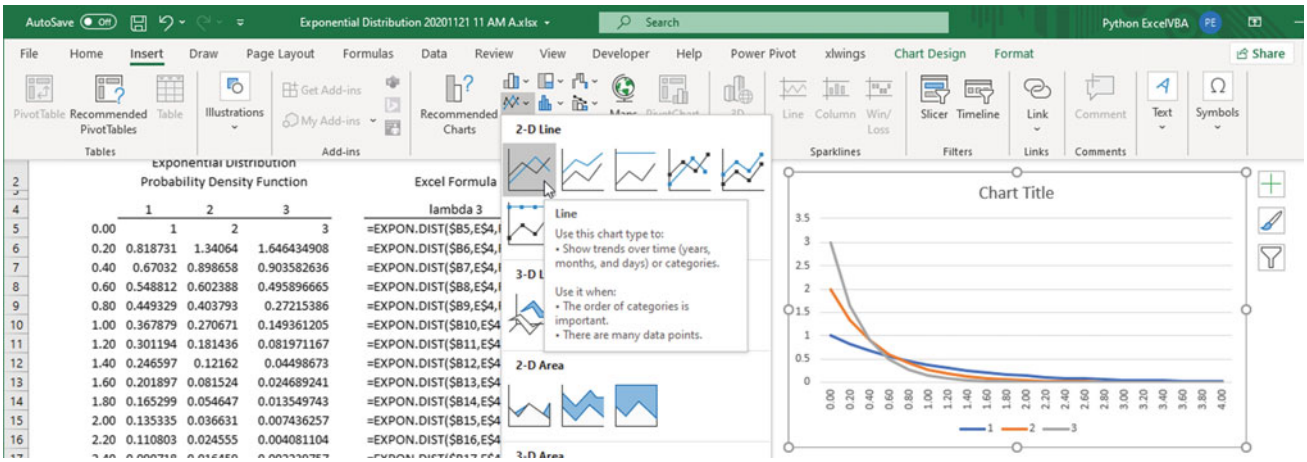
(1) Select cells B4 to E25.



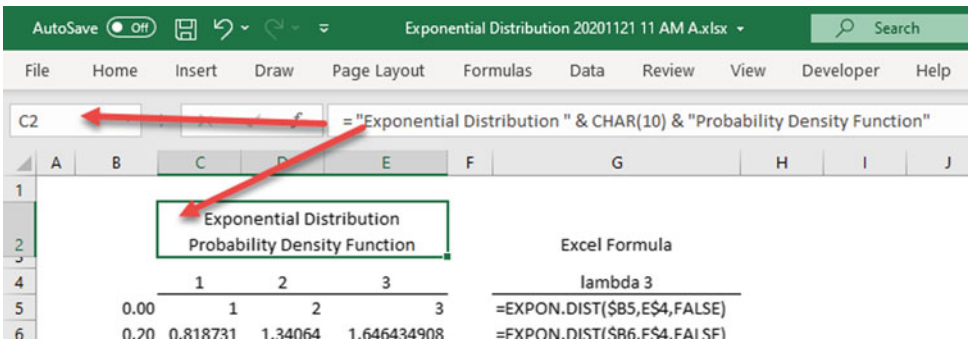
(2) Click on *Insert Line or Area Chart* from the *Insert* tab.



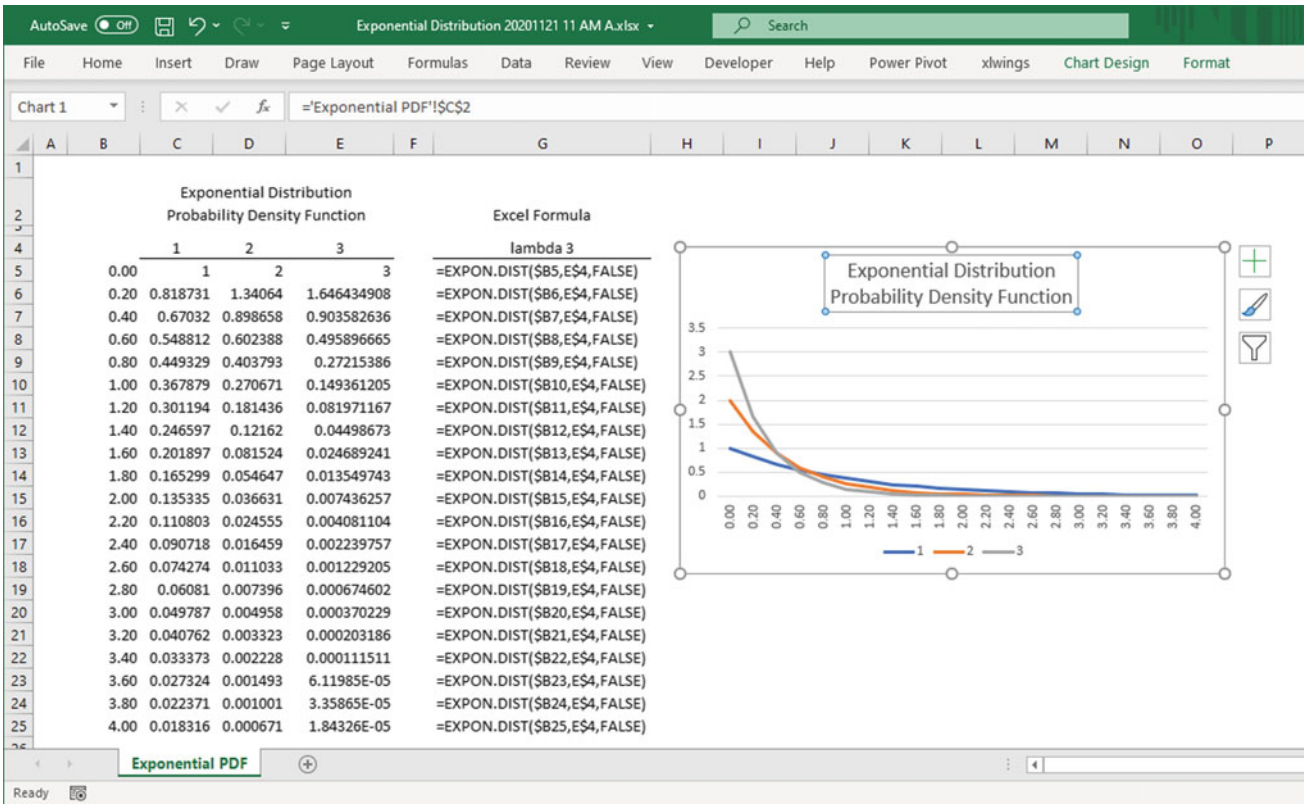
(3) Choose *Line* in the *2-D Line* row.



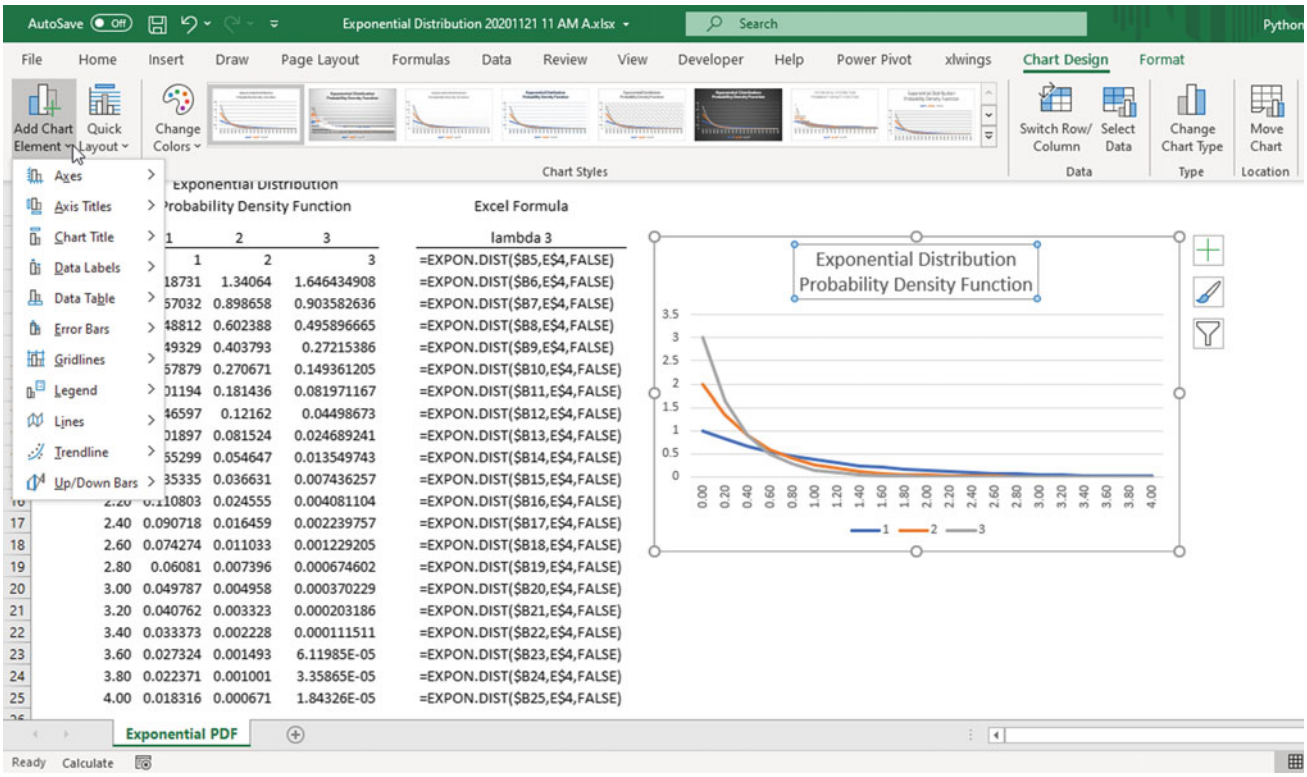
(4) In cell C2 enter "= "Exponential Distribution" & CHAR(10) & "Probability Density Function." It's the function *CHAR(10)* that causes the second.



(5) Click on the borders of the chart title and in the formula bar, enter "= 'Exponential PDF'!$C$2."

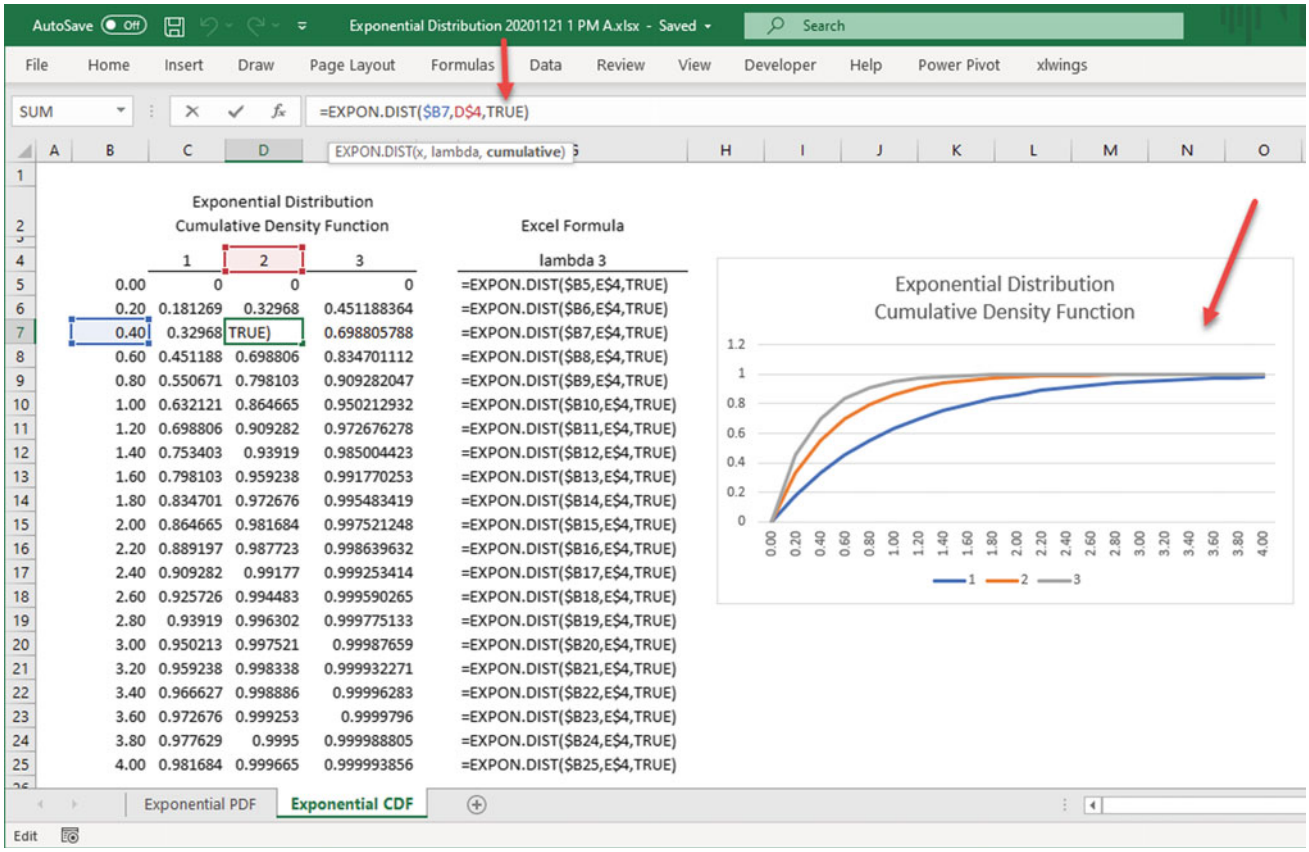


(6) With the chart selected, go to the *Add Chart Element* in the *Chart Design* tab to further format the chart.

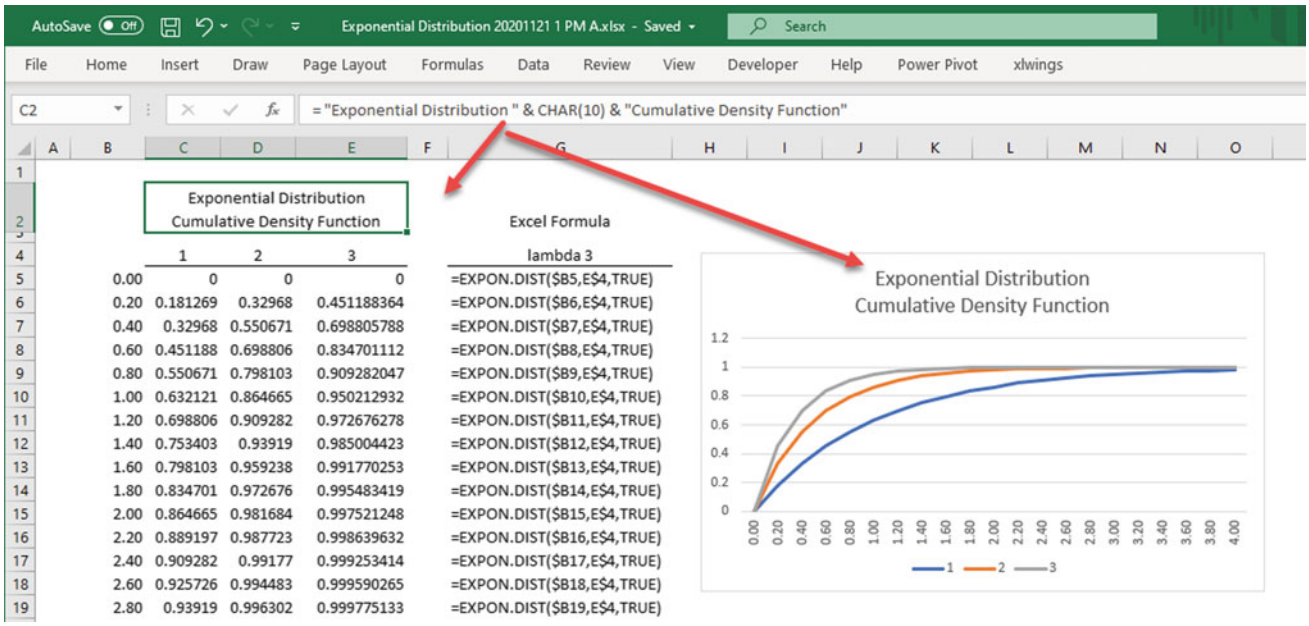


9.5.2 Exponential Cumulative Density Function in Excel

To create the *exponential* cumulative density function table, go to the *exponential* probability density function table, as shown in the previous section, and change the third parameter in the *EXPON.DIST* function from "false" to true.



In cell c2, enter the following to change the title, "= "Exponential Distribution" & CHAR(10) & "Cumulative Density Function."



Entering the title in cell c2 will automatically update the chart title.

9.6 Summary

In this chapter, we completed our survey of important continuous variables. We looked at the *t*, chi-square, *F,* and exponential distributions. We have demonstrated that the *t*-distribution approaches a standard normal distribution as the degrees of freedom increase. For the chi-square distribution, we have demonstrated that as the degrees of freedom increase, the farther right the density function shifts. For the *F*-distribution, we have demonstrated that as the degrees of freedom increase, the more peaked its density function becomes. For the exponential distribution, we have shown how the shape can be affected by the value of the parameter λ.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Estimation

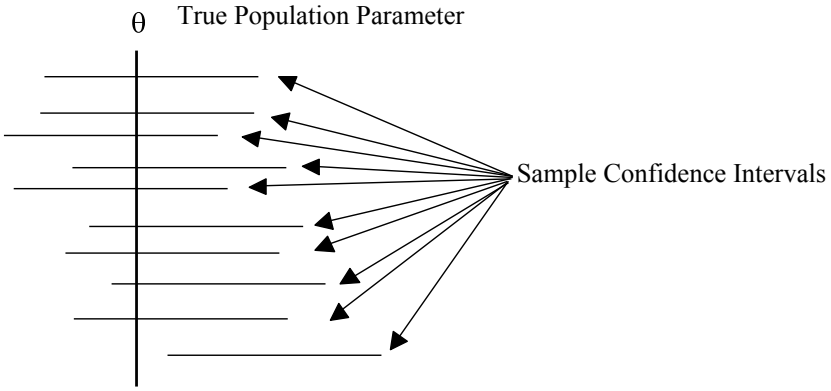
10

10.1 Introduction

This chapter uses the continuous variables that we studied in Chaps. 7 and 9 to estimate the mean, variance, and population proportion.

Often, we are interested in making inferences about a population parameter based on a sample parameter. One method for doing this is to make a confidence interval that might contain the population parameter.

Confidence intervals are intervals that have a chance of containing the population parameter. For example, a 90% confidence interval means that 90% of the confidence intervals we create will contain the true population parameter. The concept of confidence intervals is illustrated below.



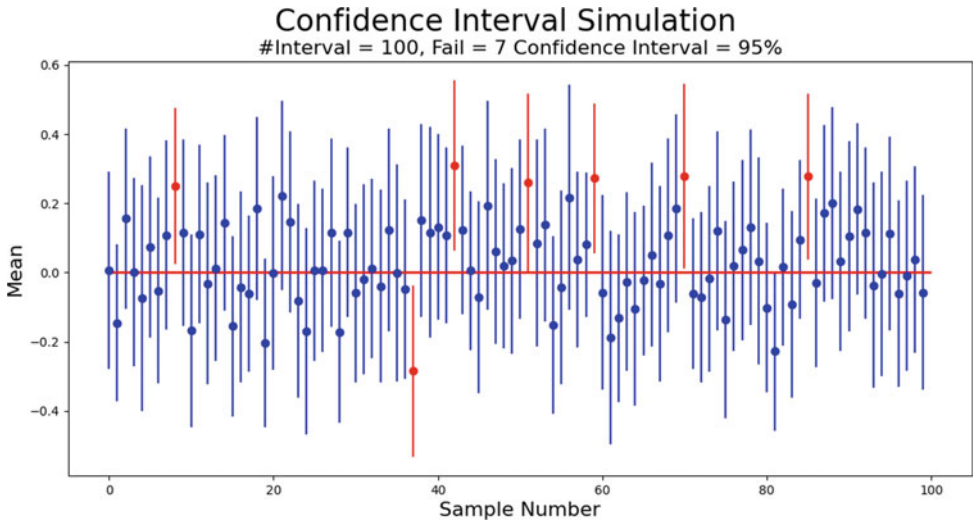
From this diagram of 90% confidence intervals, we can see that nine of the ten intervals contain the true population parameter, while one of them does not contain the true mean.

This chapter is broken down into the following sections. Section 10.2 discusses the confidence interval simulation in Python, Sect. 10.3 discusses interval estimates for when 2 is known, and Sect. 10.4 talks about the confidence intervals for μ when σ2 is unknown. Section 10.5 discusses the confidence intervals for the population proportion and Sect. 10.6 discusses the confidence intervals for the variance. Finally, Sect. 10.7 summarizes the chapter.

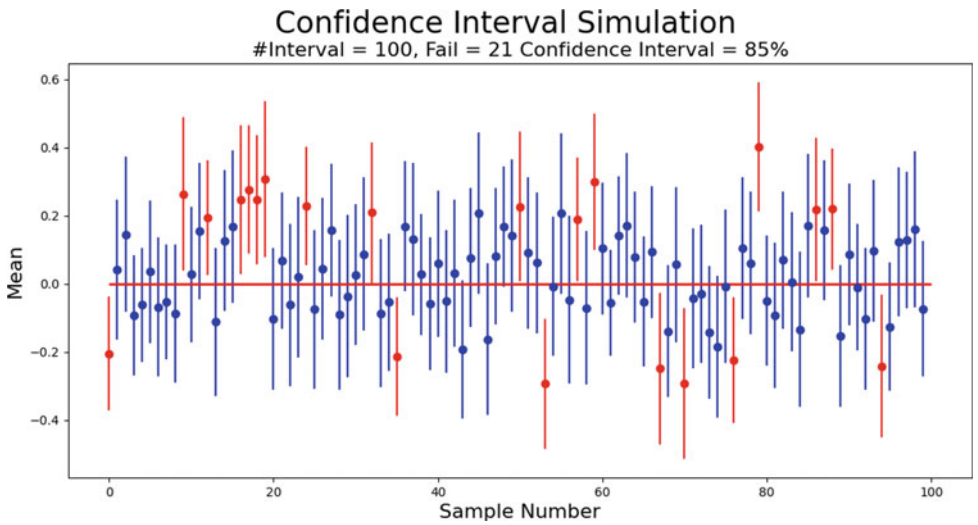
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_10

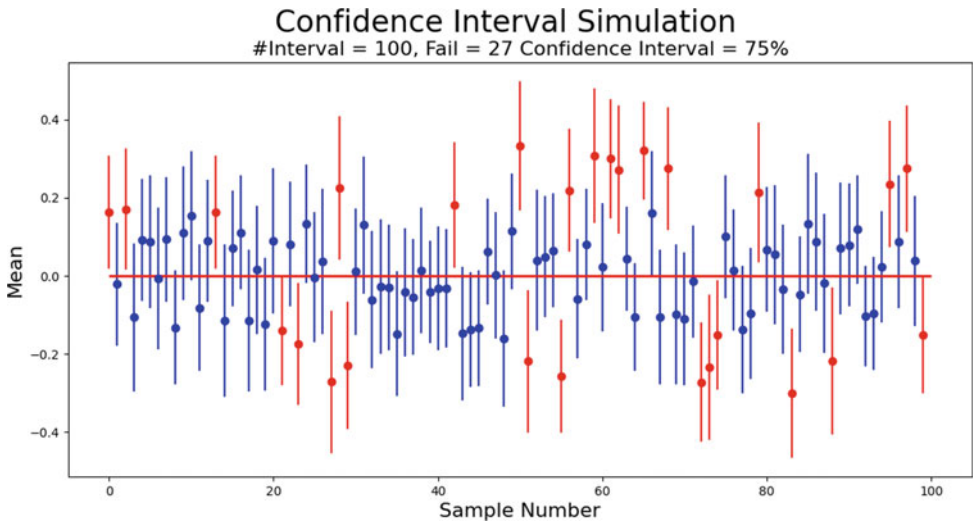
10.2 Confidence Interval Simulation in Python



Above is the result of simulating a 95% confidence interval in Python. In a 95% confidence interval simulation, we would expect that out of 100 confidence intervals, 5 of the confidence intervals will not contain the population mean. In the simulation, seven of the confidence intervals did not contain the population mean. The failed confidence intervals are shown in red.



Above is the result of simulating an 85% confidence interval. In an 85% confidence interval simulation, we would expect that out of 100 confidence intervals, 15 of the confidence intervals will not contain the population mean. In the simulation, 21 of the confidence intervals did not contain the population mean. The failed confidence intervals are shown in red.



Above is the result of simulating a 75% confidence interval. In a 75% confidence interval simulation, we would expect that out of 100 confidence intervals, 25 of the confidence intervals will not contain the population mean. In the simulation, 27 of the confidence intervals did not contain the population mean.

10.2.1 Python Code

Chapter10_ConfidenceIntervalSimulation 20201212 7 AM.py

```
#Date:Dec-12-2020 7AM
import numpy as np,pandas as pd,random,math,datetime
import scipy.stats as stats, matplotlib.pyplot as plt
from tabulate import tabulate

def CI_simulation(Num_of_interval, Sample_size, pop_mean, pop_std, CI_range):
  #generate population
  population = np.random.normal(pop_mean,pop_std,1000)
  intervals =[]
  Sample_data = []
  for sample in range(Num_of_interval):
    sample = np.random.choice(a= population, size = Sample_size)
    sample_mean = sample.mean()
    Sample_data.append(sample)
    t_critical = stats.t.ppf((1-CI_range)/2+CI_range,Sample_size)  # Get the t-critical value*
    margin_of_error = t_critical * (np.std(sample)/math.sqrt(Sample_size))# get marginal error
    confidence_interval = (sample_mean - margin_of_error,sample_mean,sample_mean)
    intervals.append(confidence_interval)

  plt.figure(figsize=(13,6))

```

```
    #record data
    top_data = []
    mean_data = []
    bot_data = []
    pass_fail = []
    Fail = 0 #count the fail times
    for i in range(Num_of_interval):
        (top,mean,bot)= intervals[i]
        top_data.append(top)
        mean_data.append(mean)
        bot_data.append(bot)

        if abs(mean-pop_mean)>abs(top-mean):
            plt.errorbar(i, mean, yerr=top-mean, fmt='o',color='red')
            Fail = Fail+1
            pass_fail.append('Fail')
        else:
            plt.errorbar(i, mean, yerr=top-mean, fmt='o',color='blue')
            pass_fail.append('Pass')

    plt.hlines(xmin=0, xmax=Num_of_interval,y=0, linewidth=2.0,color="red")
    plt.title('#Interval = '+str(Num_of_interval)+', Fail = '+str(Fail)+
              ' Confidence Interval = '+'{:.0%}'.format(CI_range),fontsize=16)
    plt.suptitle('Confidence Interval Simulation',fontsize=24)
    plt.xlabel('Sample Number', fontsize = 16)
    plt.ylabel('Mean', fontsize = 16)
    plt.show(block=False)
    return Sample_data,top_data,mean_data,bot_data,pass_fail

#************************************************************.....95%·confidence·interval·simulation



CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.95)

#last 4 columns creates interval and indicates if it contains population mean
CI_data = pd.DataFrame(CI_data)
CI_data.insert(50, "bottom",bot_data , True)
CI_data.insert(51, "mean",mean_data , True)
CI_data.insert(52, "top",top_data , True)
CI_data.insert(53, "result",pass_fail , True)

# saving 95 percent CI data to current folderas an Excel File
CI_data.to_excel('Chapter10_CI_95Percent.xlsx')

#************************************************************.....85%·confidence·interval·simulation
CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.85)

#last 4 columns creates interval and indicates if it contains population mean
CI_data = pd.DataFrame(CI_data)
CI_data.insert(50, "bottom",bot_data , True)
CI_data.insert(51, "mean",mean_data , True)
CI_data.insert(52, "top",top_data , True)
CI_data.insert(53, "result",pass_fail , True)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
print("Data for 85% Confidence Interval")

CI_data.to_excel('Chapter10_CI_85Percent.xlsx')

#************************************************************.....75%·confidence·interval·simulation
CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.75)

#print data set used in above graph
#last 4 columns creates interval and indicates if it contains population mean
CI_data = pd.DataFrame(CI_data)
CI_data.insert(50, "bottom",bot_data , True)
CI_data.insert(51, "mean",mean_data , True)
CI_data.insert(52, "top",top_data , True)
CI_data.insert(53, "result",pass_fail , True)

CI_data.to_excel('Chapter10_CI_75Percent.xlsx')

plt.show()
```

Above is the Python code that generated 95%, 85%, and 75% confidence in interval simulations. Lines 2–4 are the Python libraries used.

Lines 6–49 are the lines of the custom-defined Python function *CI_simulation*. All the confidence interval simulations use the *CI_simulation* function. Each simulation would pass the simulation's number of intervals, the same size of the simulation, the population mean, the population standard deviation, and the confidence interval.

The arguments of the *CI_simulation* function describe this

```
6  def CI_simulation(Num_of_interval, Sample_size, pop_mean, pop_std, CI_range):
```

Line 52 creates the 95% confidence interval simulation graph.

```
51  #*************************************************************.....95%·confidence·interval·simulation
52  CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.95)
```

Line 65 creates the 85% confidence interval simulation graph.

```
64  #*************************************************************.....85%·confidence·interval·simulation
65  CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.85)
```

Line 80 creates the 75% confidence interval simulation graph.

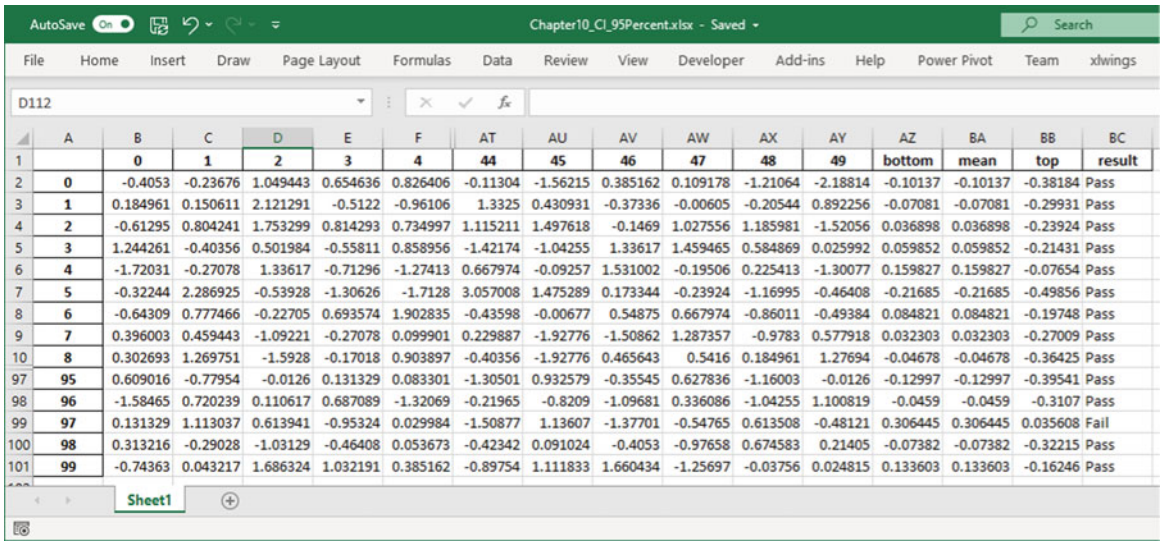
```
79  #*************************************************************.....75%·confidence·interval·simulation
80  CI_data,top_data,mean_data,bot_data,pass_fail = CI_simulation(100, 50, 0, 1, 0.75)
```

10.2.2 Confidence Interval Simulation Data

Lines 62, 77, and 90 save the simulation data to a Microsoft Excel file.

```
61  # saving 95 percent CI data to current folderas an Excel File
62  CI_data.to_excel('Chapter10_CI_95Percent.xlsx')
63
76
77  CI_data.to_excel('Chapter10_CI_85Percent.xlsx')
78
89
90  CI_data.to_excel('Chapter10_CI_75Percent.xlsx')
91
```

Python starts counting at 0 instead of 1; this is why the column representing sample size stops at 49 instead of 50 and the rows representing the number of samples stop at 99 instead of 100.



AutoSave On — Chapter10_CI_95Percent.xlsx - Saved — Search

File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins Help Power Pivot Team xlwings

D112

	A	B	C	D	E	F	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC
1		0	1	2	3	4	44	45	46	47	48	49	bottom	mean	top	result
2	0	-0.4053	-0.23676	1.049443	0.654636	0.826406	-0.11304	-1.56215	0.385162	0.109178	-1.21064	-2.18814	-0.10137	-0.10137	-0.38184	Pass
3	1	0.184961	0.150611	2.121291	-0.5122	-0.96106	1.3325	0.430931	-0.37336	-0.00605	-0.20544	0.892256	-0.07081	-0.07081	-0.29931	Pass
4	2	-0.61295	0.804241	1.753299	0.814293	0.734997	1.115211	1.497618	-0.1469	1.027556	1.185981	-1.52056	0.036898	0.036898	-0.23924	Pass
5	3	1.244261	-0.40356	0.501984	-0.55811	0.858956	-1.42174	-1.04255	1.33617	1.459465	0.584869	0.025992	0.059852	0.059852	-0.21431	Pass
6	4	-1.72031	-0.27078	1.33617	-0.71296	-1.27413	0.667974	-0.09257	1.531002	-0.19506	0.225413	-1.30077	0.159827	0.159827	-0.07654	Pass
7	5	-0.32244	2.286925	-0.53928	-1.30626	-1.7128	3.057008	1.475289	0.173344	-0.23924	-1.16995	-0.46408	-0.21685	-0.21685	-0.49856	Pass
8	6	-0.64309	0.777466	-0.22705	0.693574	1.902835	-0.43598	-0.00677	0.54875	0.667974	-0.86011	-0.49384	0.084821	0.084821	-0.19748	Pass
9	7	0.396003	0.459443	-1.09221	-0.27078	0.099901	0.229887	-1.92776	-1.50862	1.287357	-0.9783	0.577918	0.032303	0.032303	-0.27009	Pass
10	8	0.302693	1.269751	-1.5928	-0.17018	0.903897	-0.40356	-1.92776	0.465643	0.5416	0.184961	1.27694	-0.04678	-0.04678	-0.36425	Pass
97	95	0.609016	-0.77954	-0.0126	0.131329	0.083301	-1.30501	0.932579	-0.35545	0.627836	-1.16003	-0.0126	-0.12997	-0.12997	-0.39541	Pass
98	96	-1.58465	0.720239	0.110617	0.687089	-1.32069	-0.21965	-0.8209	-1.09681	0.336086	-1.04255	1.100819	-0.0459	-0.0459	-0.3107	Pass
99	97	0.131329	1.113037	0.613941	-0.95324	0.029984	-1.50877	1.13607	-1.37701	-0.54765	0.613508	-0.48121	0.306445	0.306445	0.035608	Fail
100	98	0.313216	-0.29028	-1.03129	-0.46408	0.053673	-0.42342	0.091024	-0.4053	-0.97658	0.674583	0.21405	-0.07382	-0.07382	-0.32215	Pass
101	99	-0.74363	0.043217	1.686324	1.032191	0.385162	-0.89754	1.111833	1.660434	-1.25697	-0.03756	0.024815	0.133603	0.133603	-0.16246	Pass

Sheet1

10.3 Interval Estimates for μ When σ^2 is Known

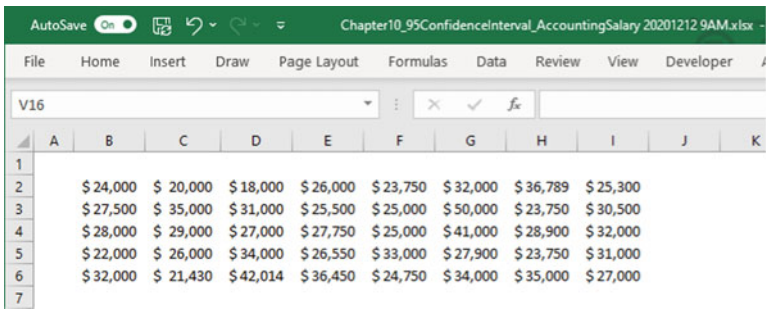
In general, if the sample size is greater than 30, we know that the sample distribution will approximate a normal distribution. We can substitute the sample variance for the population variance when the sample size is greater than 30. We can do this because of the central limit theorem. Below is the formula to create the confidence interval for the population mean:

$$\bar{x} - z_{\alpha/2}(\sigma/\sqrt{n}) < \mu < \bar{x} + z_{\alpha/2}(\sigma/\sqrt{n})$$

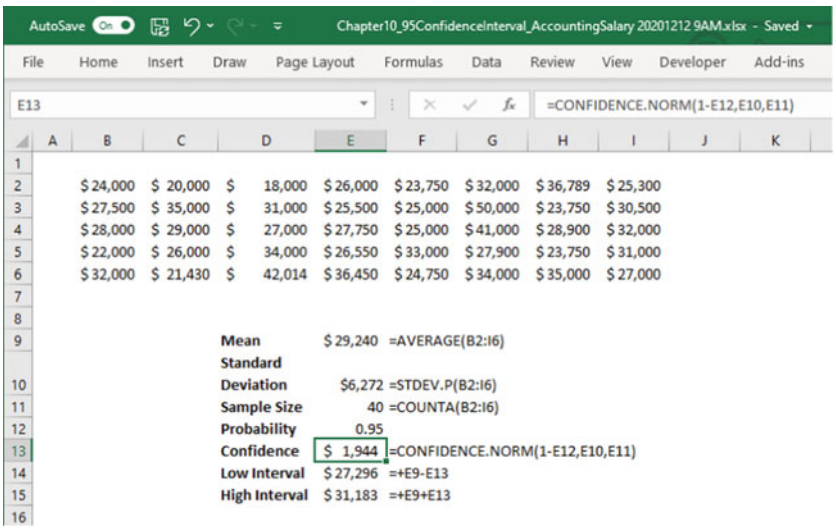
10.3.1 Z Confidence Intervals

10.3.1.1 Example A

Suppose we are interested in creating a 95% confidence interval for the following starting salaries for accountants.

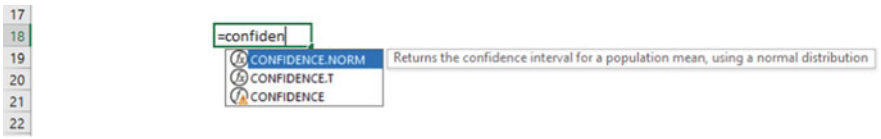


Below shows how to create the 95% confidence interval in Excel.



In cell E10, we used the population standard deviation Excel function instead of the sample population standard deviation Excel function.

In cell E13, we used the Excel function *CONFIDENCE.NORM* to create a confidence interval.

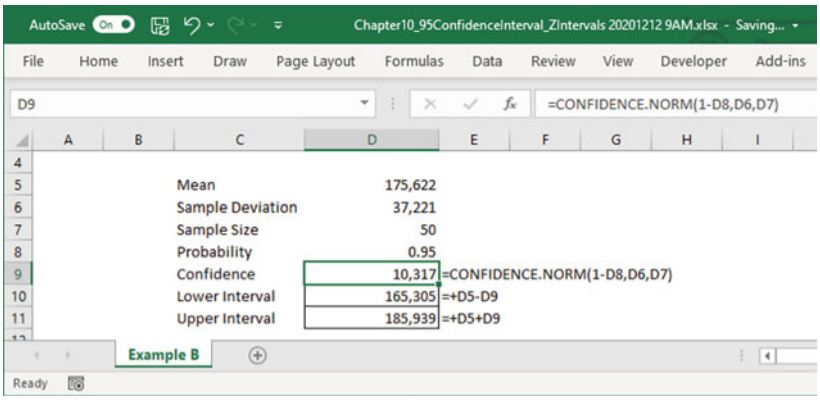


Excel calculated that the 95% confidence interval for the accounting salaries is between $27,296 and $31,183 and that the mean is $29,240.

Often we are given the sample size, mean, and standard deviation but not the data set.

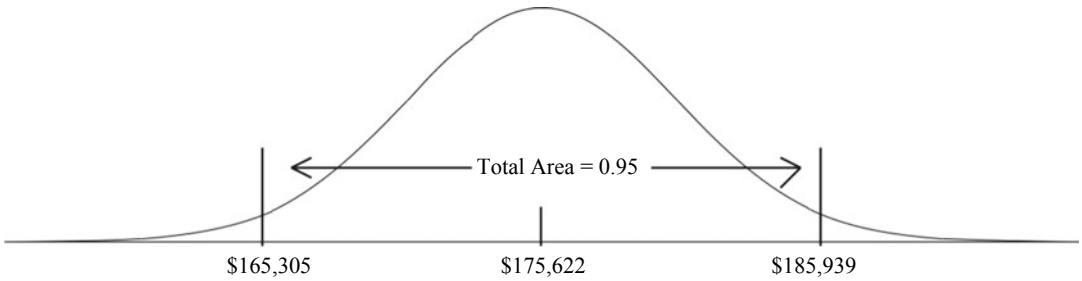
10.3.1.2 Example B

Suppose a real estate agent in Connecticut is interested in the state's mean home price. A random sample of 50 homes shows a mean price of $175,622, assuming a population sample standard deviation of $37,221. What is the 95% confidence interval for the mean home price for Connecticut?



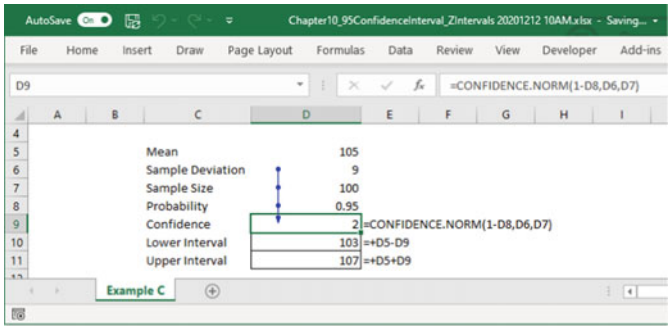
Excel tells us that the 95% confidence interval for Connecticut's mean home price is from $165,305 to $185,939.

The confidence interval is graphically illustrated below.



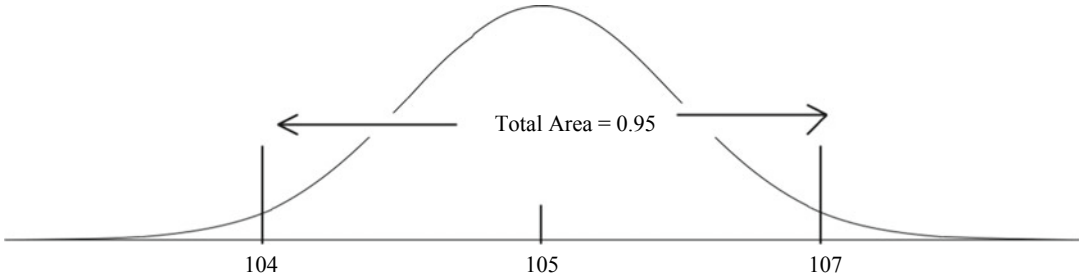
10.3.1.3 Example C

Suppose a machine dispenses sand into bags. The population standard deviation is 9.0 pounds and the weights are normally distributed. A random sample of 100 bags is taken. The sample mean is 105 pounds. Calculate a 95% confidence interval for the mean weight of the bags.



We can see that the 95% confidence interval for the bags' mean weight is 103–107 pounds from the output.

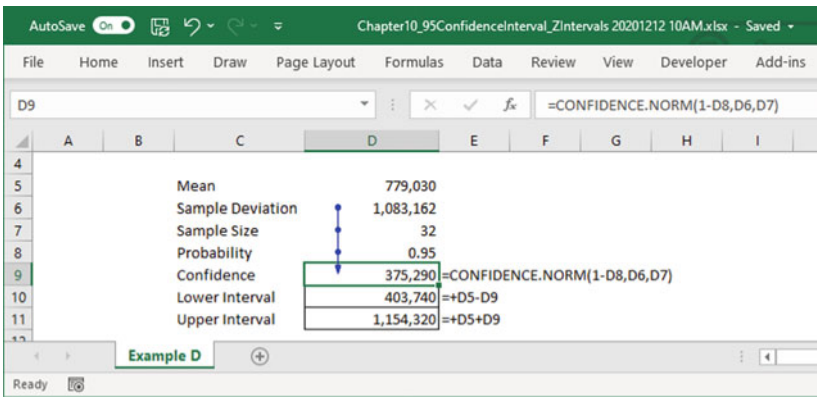
The confidence interval is graphically illustrated below.



10.3.1.4 Example D

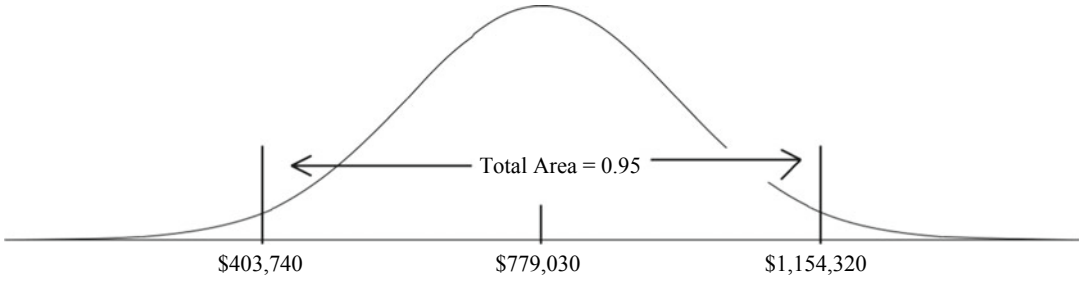
To study the effect of internal audit departments on external audit fees, W. A. Wallace conducted a survey of the audit department of 32 diverse companies (*Harvard Business Review*, March–April 1984). She found that the mean annual external audit paid by the 32 companies was $779,030 and the standard deviation was $1,083,162. Find the 95% confidence interval for the mean external audit fees.

The calculation in Excel is shown below.



From the output, we can see that the 95% confidence interval for the mean external audit fees is from $403,740 to $1,154,320.

The confidence interval is graphically illustrated below.



10.4 Confidence Intervals for μ When σ^2 is Unknown

When the sample size is small, less than 30, and the data have a normal distribution, we will have to use the *t*-test. The *t*-test uses a *t* distribution. The formula to create the confidence interval would be

$$\overline{x} - t_{n-1,a/2}(s/\sqrt{n}) < \mu < \overline{x} + t_{n-1,a/2}(s/\sqrt{n})$$

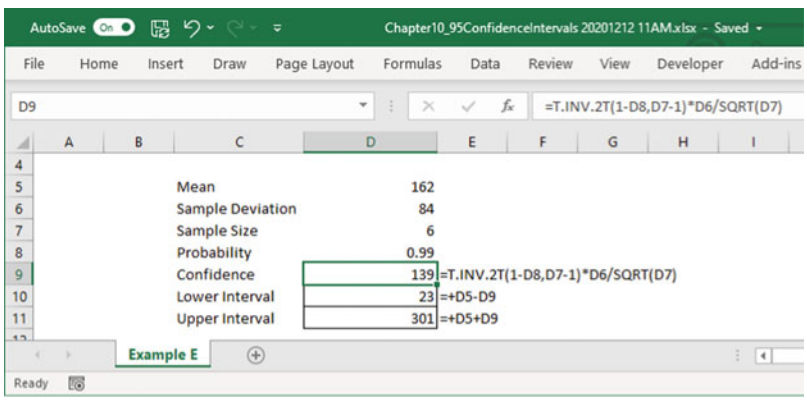
10.4.1 T Confidence Intervals

10.4.1.1 Example E

Suppose managers at the Smooth Ride Car Rental Company are interested in the mean number of miles that customers drive per day. A random sample of six-car rentals shows that the customers drove the following numbers of miles:

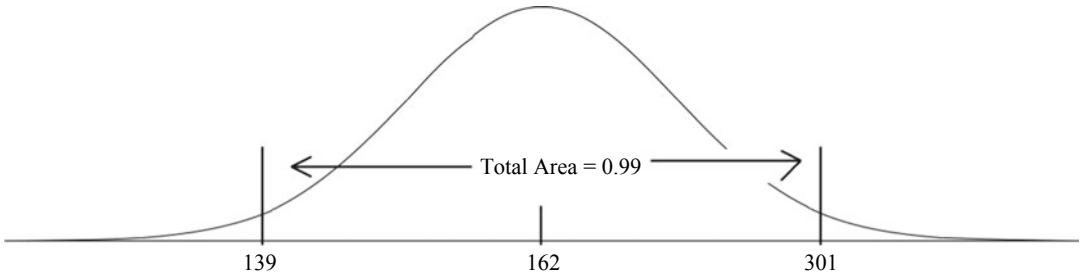
$$152, 222, 300, 84, 90, 122$$

Construct a 99% confidence interval for the mean number of miles driven.



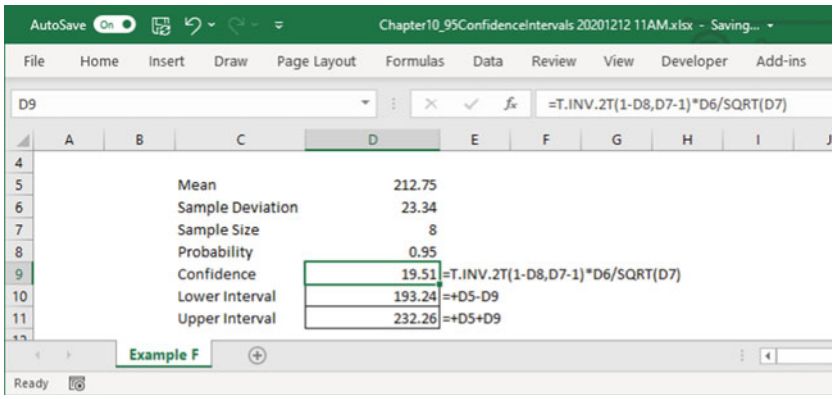
From the output, we can see that the 99% confidence interval for the mean number of miles that customers drive per day is from139 miles to 301 miles.

The confidence interval is graphically illustrated below.



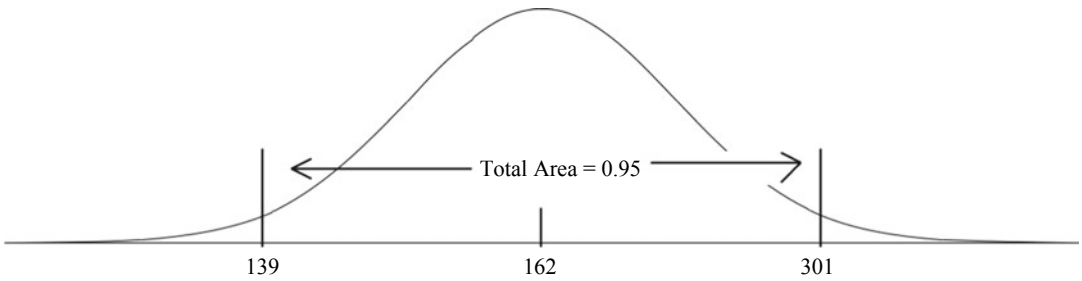
10.4.1.2 Example F

Suppose after randomly selecting eight football players, we found that the mean weight was 212.75 and the sample standard deviation was 23.34. Calculate the 95% confidence interval.



From the output, we can see that the 95% confidence interval for the average weight of football players is from 193.23 miles to 232.26 miles.

The confidence interval is graphically illustrated below.



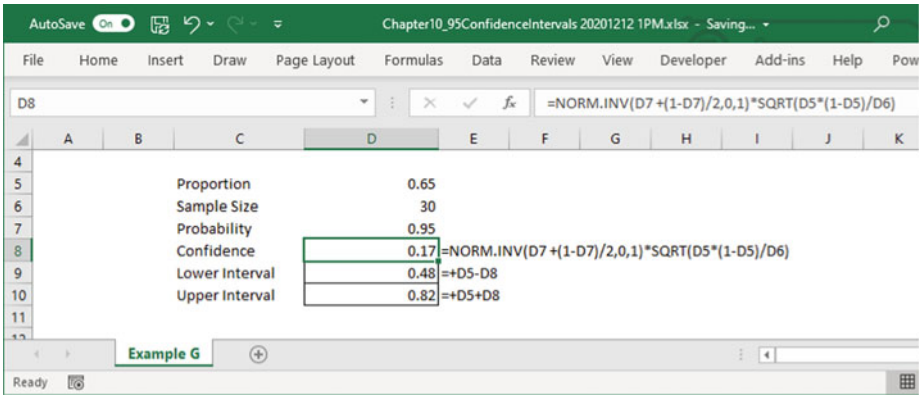
10.5 Confidence Intervals for the Population Proportion

Because of the central limit theorem and if the sample size is large enough, we can calculate the confidence interval for the population proportion by using the normal distribution. The formula to calculate the confidence interval for the population proportion is

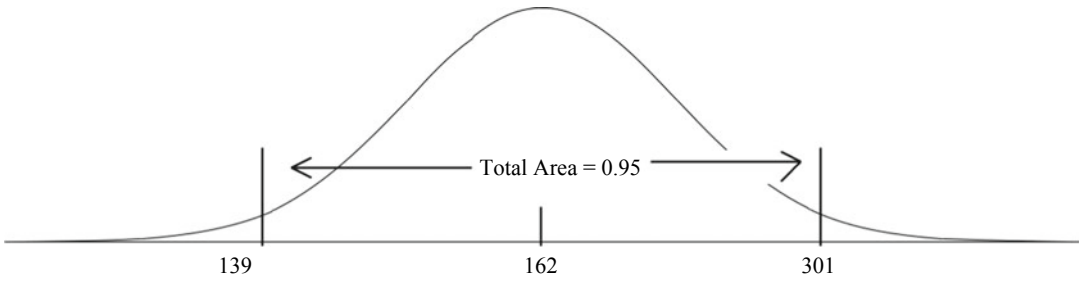
$$\widehat{P} - z_{\alpha/2}\sqrt{\frac{\widehat{P}(1-\widehat{P})}{n}} < P < \widehat{P} + z_{\alpha/2}\sqrt{\frac{\widehat{P}(1-\widehat{P})}{n}}$$

10.5.1 Example G

A marketing firm discovers that 65% of the 30 customers who participated in a blind taste test prefer brand A to brand B. Develop a 95% confidence interval for the number of people who prefer brand A.

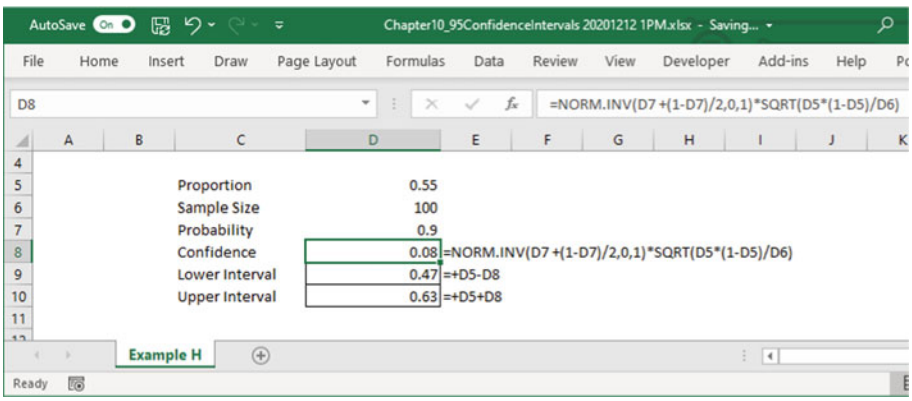


From the above calculation, the confidence interval for those who preferred brand A lies between 48 and 82%. The confidence interval is graphically illustrated below.



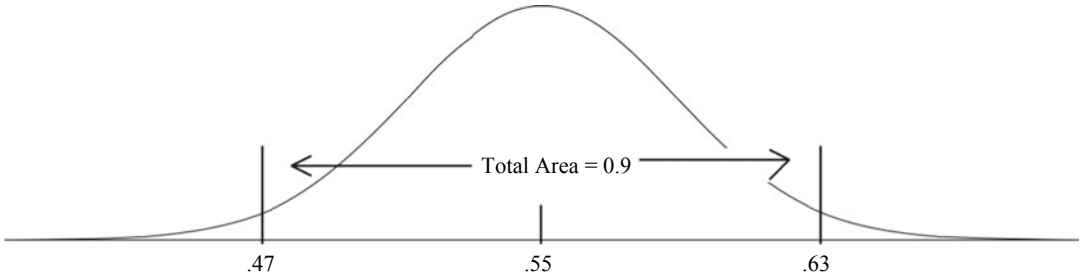
10.5.2 Example H

Suppose that a random sample of 100 voters is taken, and 55% of the sample supports the incumbent candidate. Construct a 90% confidence interval for the proportion that supports the incumbent candidate.



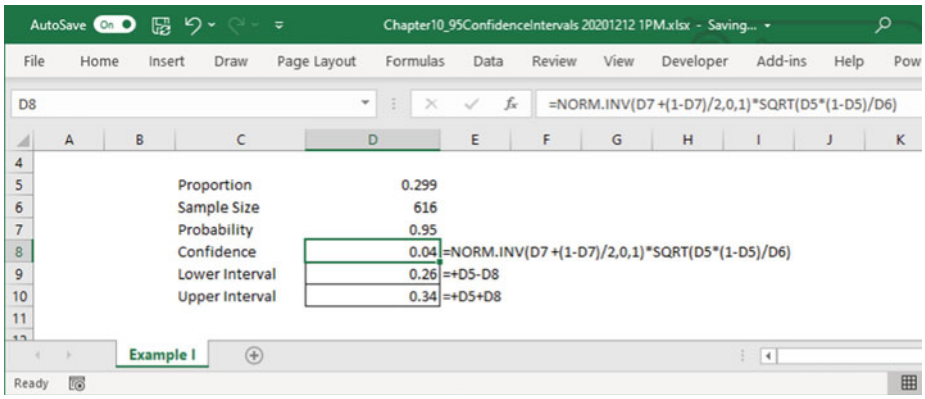
From the above calculation, we learn that the confidence interval for those who support the incumbent candidate is from 47 to 63%.

The confidence interval is graphically illustrated below.



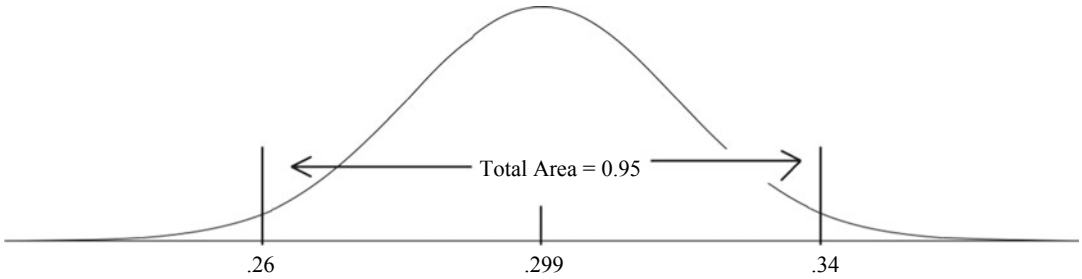
10.5.3 Example I

A study in the *Journal of Advertising Research* (April/May 1984) to find the proportion of working adults using computer equipment on the job employed the random sample approach to survey 616 working adults. The survey revealed that 184 of the adults, or 29.9%, regularly used computer equipment on the job. Find the 95% confidence interval for working adults' computer usage.



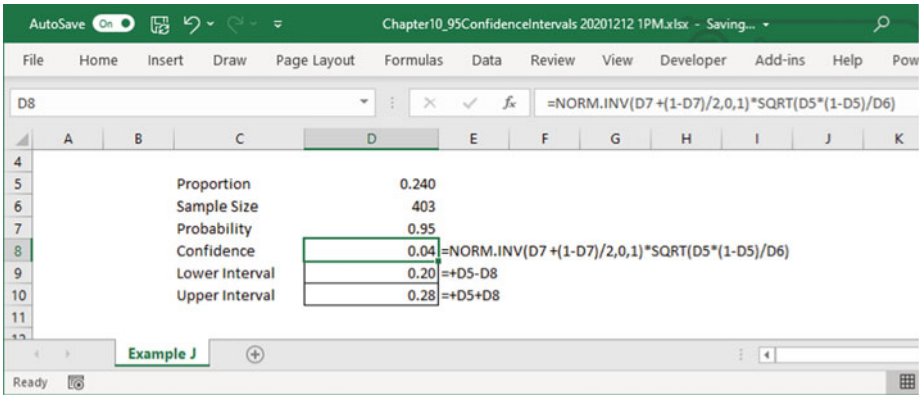
From the above calculation, we can see that the 95% confidence interval for working adults' computer usage was from 26 to 33%.

The confidence interval is graphically illustrated below.



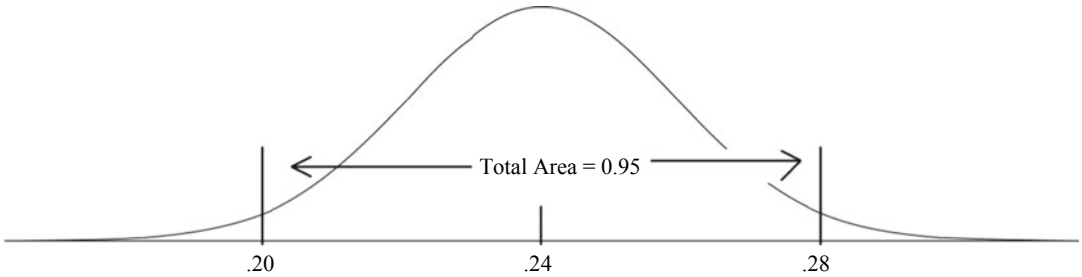
10.5.4 Example J

Harris et al. (1979) studied the attitudes of shoppers toward shoplifting and devices for its prevention. They sampled 403 shopping center patrons. Twenty-four percent of this sample expressed awareness of and discomfort with the use of TV cameras as a device to prevent shoplifting. Find the 95% confidence interval of the population in order to find the proportion that dislikes the use of TV devices to prevent shoplifting.



From the above calculations, we can see that the 95% confidence interval proportion that dislikes using TV cameras to prevent shoplifting is from 20 to 28%.

The confidence interval is graphically illustrated below.



10.6 Confidence Intervals for the Variance

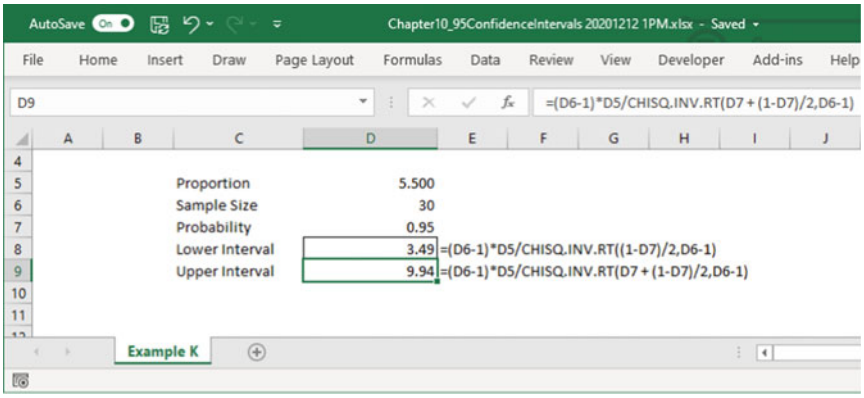
The confidence interval for variances uses a chi-square distribution. The formula for the confidence interval for a variance is

$$\frac{(n-1)s_x^2}{\chi^2_{v,\alpha/2}} < \sigma^2 < \frac{(n-1)s_x^2}{\chi^2_{v,1-\alpha/2}}$$

10.6.1 Example K

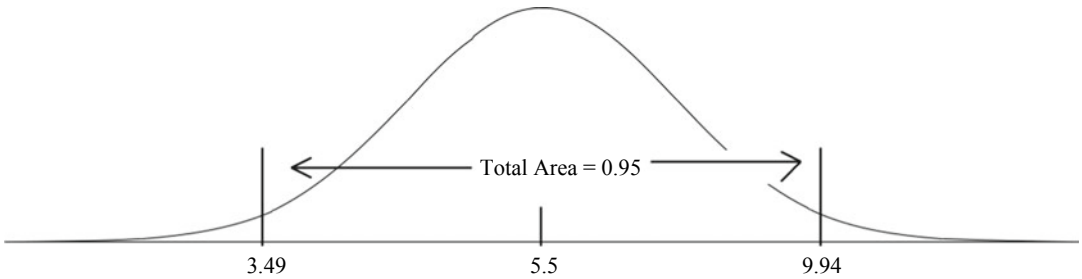
Suppose a random sample of 30 bags of sand is taken, and the sample variance of weights is 5.5. Find a 95% confidence interval for the population variance.

Below shows how to set up Excel to calculate a confidence interval for a population variance.



From the above calculation, we learn that the confidence interval for the sample variance of sandbag weight is between 3.49 and 9.94 pounds.

The confidence interval is graphically illustrated below.



10.7 Summary

In this chapter, we created a confidence interval simulation in Python to demonstrate the confidence interval concept.

We created the confidence interval for the population mean when the population variance was known and unknown. We also found the confidence interval for the population proportion and the confidence interval for the population variance.

In this chapter, we calculated the confidence interval for the population mean. Most of the time, the population variance is unknown but because of the central limit theorem, we can use the sample variance to approximate the population variance and use the normal distribution to calculate the confidence interval. We can substitute the sample variance for the population variance when the sample size is 30 or greater.

If the sample size is smaller than 30, we will need to use the *t* distribution to calculate the confidence interval for the population mean.

Also, we can calculate the confidence interval for the population proportion using the normal distribution because of the central limit theorem.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Harris, J. R., Guffey, H. J, and Laumer, J. F. "The windshield method of questionnaire distribution: A viable alternative." *Journal of the Academy of Marketing Science*, v. 7(3) (1979), pp. 184–191.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Check for updates

11 Hypothesis Testing

11.1 Introduction

In the last chapter, we made inferences about a population parameter by creating a confidence interval from a sample. We will now look at another method, called *hypothesis testing*, for making inferences about a population parameter. In hypothesis testing, we infer that the stated null hypothesis (H_0) is true until there is convincing but *not perfect* evidence that the null hypothesis is not true. Our evidence is from the sample that we obtain. We conclude that there is convincing evidence when the *p*-value is less than the alpha value. The *p*-value will be discussed later in this chapter.

Because the sample evidence is not perfect, we can make four kinds of decisions:

1. We can make a correct decision by rejecting H_0 when, in fact, the true H_0 is false.
2. We can make a correct decision by failing to reject H_0 when, in fact, the true H_0 is true.
3. We can make an incorrect decision by rejecting H_0 when, in fact, the true H_0 is true.
4. We can make an incorrect decision by accepting H_0 when, in fact, the true H_0 is false.

In statistics, we call situation 3 a type I error.

We will make a type I error alpha percent of the time. In statistics, we call situation 4 a type II error.

Below is a diagram of the four possible resulting decisions of hypothesis testing:

	Actual H_0 True	Actual H_0 False
Fail to Reject H_0	Correct Decision	Type II Error
Reject H_0	Type I Error	Correct Decision

This chapter is organized as follows. Section 11.2 discusses one-tailed tests of mean for large samples, Sect. 11.3 discusses Z-test, and Sect. 11.4 discusses hypothesis testing and the *p*-value. Section 11.5 talks about one-tailed tests of mean for large samples and Sect. 11.6 discusses two-tailed tests of mean for large samples. Meanwhile, Sect. 11.7 talks about one-tailed tests of mean for small samples. In Sect. 11.8, we use hypothesis testing for a population proportion. The power of a test and power function is discussed in Sect. 11.9. Power and sample size is discussed in Sect. 11.10, and power and alpha size is discussed in Sect. 11.11. We compare the average EPS of AAPL and MSFT in Python in Sect. 11.12. Finally, in Sect. 11.13, we summarize the chapter.

11.2 One-Tailed Tests of Mean for Large Samples

We begin by examining the case where only one sample is drawn and that sample is large. Using a large sample offers two important advantages. First, we can apply the central limit theorem. Second, the large sample enables us, through our choice of significance level, alpha, to reduce our chance of making a type II error.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_11

11.2.1 Example 11.1

Suppose an accountant is interested in finding out if the average starting salary for accountants is greater than $25,000 or not. Test at the 5% alpha level. The accountant collected the following sample of 40 salaries:

$24,000	$20,000	$18,000	$26,000	$23,750	$32,000	$36,789	$25,300
$27,500	$35,000	$31,000	$25,500	$25,000	$50,000	$23,750	$30,500
$28,000	$29,000	$27,000	$27,750	$25,000	$41,000	$28,900	$32,000
$22,000	$26,000	$34,000	$26,550	$33,000	$27,900	$23,750	$31,000
$32,000	$21,430	$42,014	$36,450	$24,750	$34,000	$35,000	$27,000

The null and alternative hypotheses would then be

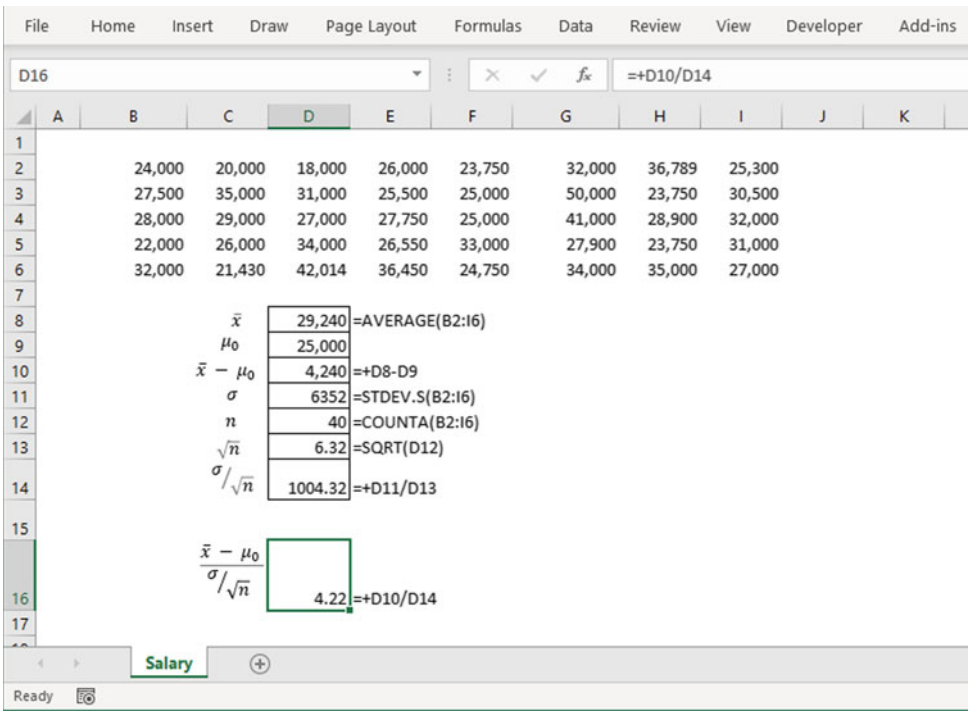
$$H_0 = \$25,000$$
$$H_1 > \$25,000$$

11.3 Z-Test

The test statistic for this would be

$$\frac{\bar{x} - \mu_0}{\sigma/\sqrt{n}} > Z_\alpha$$

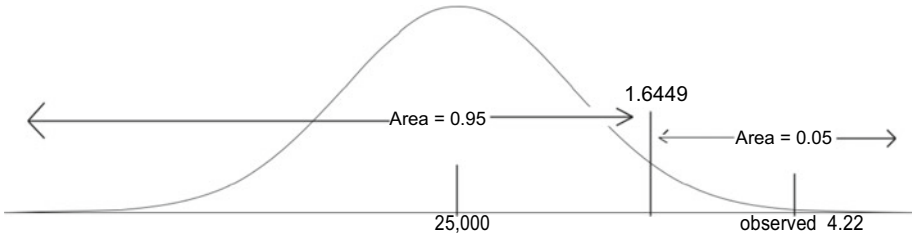
Below shows the calculation of the *Z*-score in Excel:



The *Z-score* indicates that the observed sample mean is 4.22 standard deviation away from the hypothesized mean.

We are testing at a 5% alpha level. For a one-sided right-tailed test, the 5% alpha level begins at the value 1.6449. We will reject the null hypothesis when the calculated *Z*-value is greater than 1.6449 or when the calculated *Z*-value lies in the 5% alpha area. The test statistic which we calculated, 4.22, lies in the reject region.

Graphically we are seeing the following:

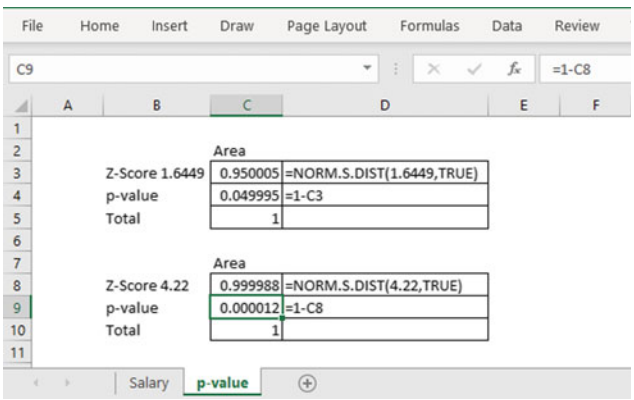


11.4 Hypothesis Testing and the *p*-Value

Another way to test the null hypothesis is to use the *p*-value. A *p*-value is the probability of getting a value more extreme than the calculated statistic. If we see that the *p*-value is less than the alpha value, we know that there is significant evidence to reject the null hypothesis because the *p*-value would lie in the region of the alpha value, as shown in the above graph. In our example, the *p*-value is the area to the right of the calculated value of 4.22. This area to the right of 4.22 is the probability of getting a value more extreme than the calculated statistic.

With the advent of computer statistical software, the use of the *p*-value method for accepting or rejecting the null hypothesis is easier than using the test statistic method. In the latter method, we would also have to look up the value that begins the alpha value. In our example, the value 1.6449 is the value where an alpha value of 5% begins. We then have to see if the calculated value is greater than the value of 1.6449. If it is, we conclude that there is enough evidence to reject the null hypothesis. If not, we conclude there is not enough evidence to reject the null hypothesis.

In the *p*-value method, we only need to see if the *p*-value is greater or less than the alpha. If it is greater or equal to the alpha value, we fail to reject the null hypothesis. If it is less than the alpha value, we reject the null hypothesis.



To calculate the *p*-value of the calculated test statistic, we first calculate the area to the left of the calculated test statistics. To do this, we use Excel's *NORM.S.DIST* function. To get the area to the right of the test statistics, we subtract by 1 because the total area of the standard normal distribution is 1. The *p*-value of the test statistics 4.22 is 0.000012. Since this *p*-value is less than the alpha value of 5%, there is enough evidence to reject the null hypothesis that the mean accounting salary is $25,000 and accept the alternative hypothesis that the mean accounting salary is greater than $25,000.

We should remember that we have strong but not perfect evidence to reject the null hypothesis. There is a 5% chance that we should not be rejecting the null hypothesis. As stated above, this kind of error in judgment is called a type I error.

11.4.1 Example 11.2

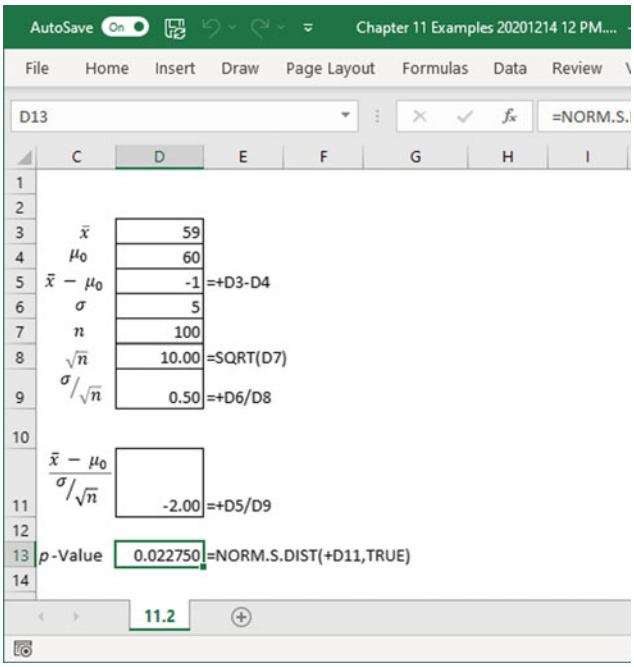
Say we want to test whether the average weight of 60-oz bags of cat food is equal to or smaller than 60 oz at significance level $\alpha = 0.05$. The null and alternative hypotheses can be stated as

$$H_0 \mu = 60$$
$$H_1 \mu < 60$$

In addition, suppose we know that the sample size n = 100, sample mean X = 59, and standard deviation s_x = 5. The test statistic for this would be

$$\frac{\overline{x} - \mu_0}{\sigma/\sqrt{n}} < Z_\alpha$$

The calculation is shown below. Based on the calculation shown below, we would reject the null hypothesis because, for a left-tail test, we reject the null hypothesis when the p-value is less than the alpha value.



11.5 One-Tailed Tests of Mean for Large Samples: Two-Sample Test of Means

Another important issue is how to test the difference between two population means, μ_1 and μ_1, of two normally distributed populations with variances ${\sigma_1}^2$ and ${\sigma_2}^2$. Because we will use a large sample, the assumption of normality is not necessary.

The test statistic for a right-tailed test testing two-sample test of means would be

$$Z_\alpha > \frac{(\overline{x}_1 - \overline{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

11.5.1 Example 11.3

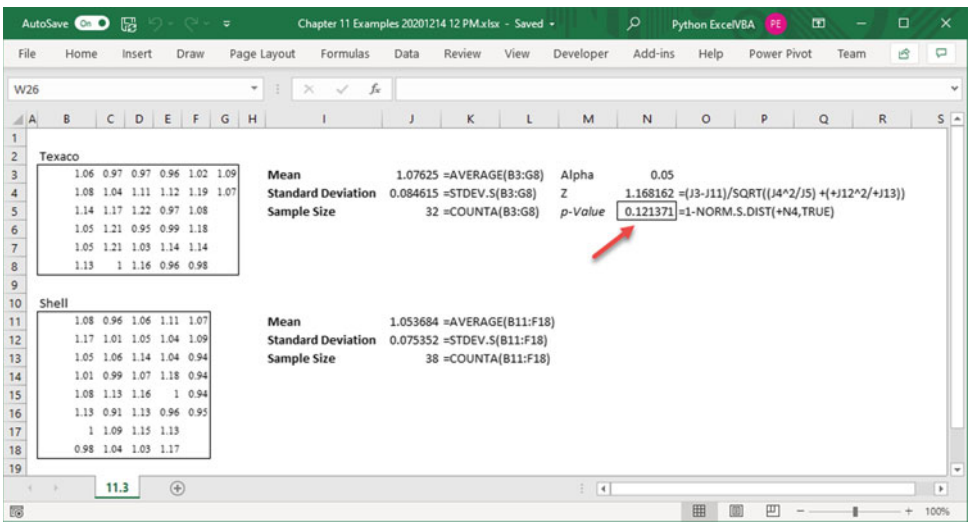
David Smith conducts a market survey to compare the prices of unleaded gasoline at Texaco stations and Shell stations. A random sample of 32 Texaco stations and 38 Shell stations in central New Jersey is used. The cost of 1 gallon of unleaded gasoline is recorded and the resulting data are summarized here.

Sample A (Texaco)					
1.06	0.97	0.97	0.96	1.02	1.09
1.08	1.04	1.11	1.12	1.19	1.07
1.14	1.17	1.22	0.97	1.08	
1.05	1.21	0.95	0.99	1.18	
1.05	1.21	1.03	1.14	1.14	
1.13	1.00	1.16	0.96	0.98	

Sample B (Shell)				
1.08	0.96	1.06	1.11	1.07
1.17	1.01	1.05	1.04	1.09
1.05	1.06	1.14	1.04	0.94
1.01	0.99	1.07	1.18	0.94
1.08	1.13	1.16	1.00	0.94
1.13	0.91	1.13	0.96	0.95
1.00	1.09	1.15	1.13	
0.98	1.04	1.03	1.17	

Test the following hypothesis at an alpha level of 0.05:

$$H_{0:}\mu_1 = \mu_2$$
$$H_{1:}\mu_1 > \mu_2$$



From the above calculations done in Excel, *p*-value is greater than alpha value, we can see that we cannot reject the null hypothesis that the average price per gallon of unleaded gasoline from Texaco stations is the same as that from Shell stations.

11.6 Two-Tailed Tests of Mean for Large Samples

In a two-tailed test, we are testing the following hypotheses:

$$H_{0\,\mu} = D$$
$$H_{1\,\mu} \neq D$$

Then we do the same calculations for the two-tailed test as for the one-tailed test. The main difference is that we make a decision based on half of the stated alpha. There are two reasons for this:

1. We are testing based on a large sample, and because of the central limit theorem, we can use the normal distribution.
2. Because we can use the normal distribution, we need to split the alpha into half because the normal distribution is symmetrical.

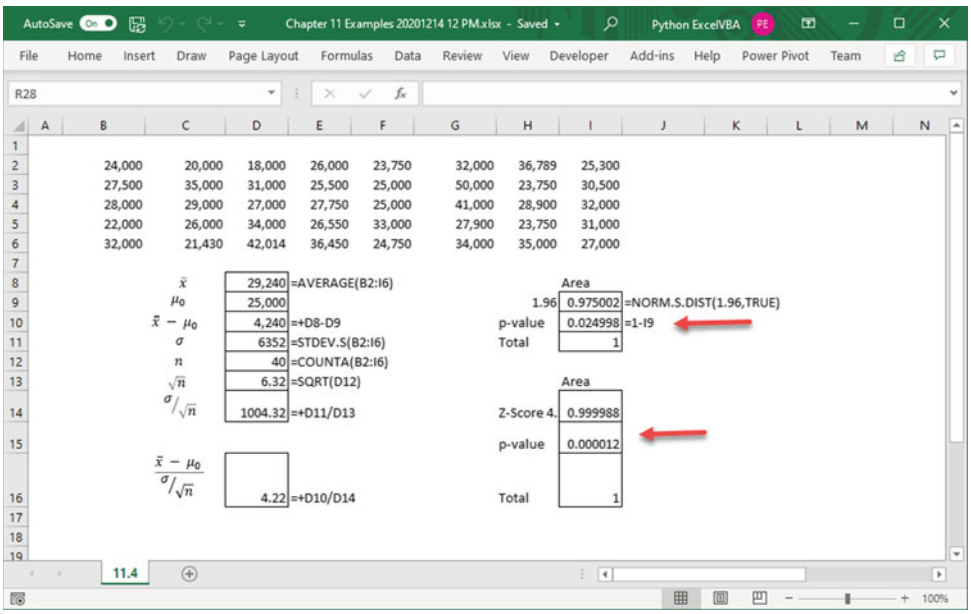
11.6.1 Example 11.4

Instead of checking if the salary was larger than $25,000, we can test to see if the mean is equal to $25,000 or not. The null and the alternative hypotheses would then be

$$H_0 = \$25,000$$
$$H_1 \neq \$25,000$$

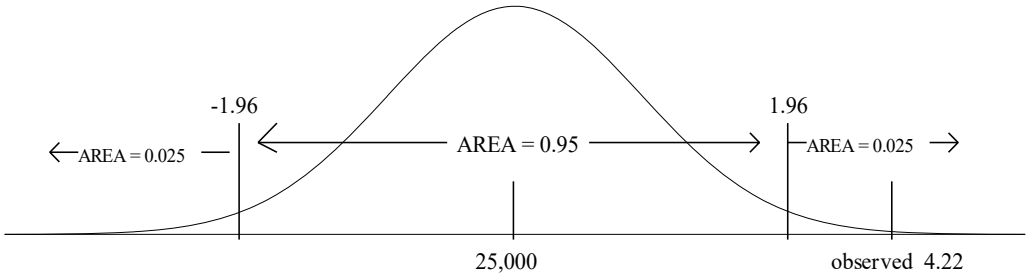
This would represent a two-tailed test. The mathematical formula to calculate this would be

$$\frac{\overline{x} - \mu_0}{\sigma/\sqrt{n}} > Z_{\alpha/2} \text{ or } \frac{\overline{x} - \mu_0}{\sigma/\sqrt{n}} < -Z_{\alpha/2}$$



The calculated *Z*-score for the data set is 4.22. We will reject the null hypothesis if the sample *Z*-score is greater than the standard *Z*-score of 1.96. Since this is the case, we can reject the null hypothesis that the mean salary is $25,000 and conclude that the mean salary is different from $25,000.

Using the *p*-value method, we can see that the calculated *p*-value is 0.000012. Since this is less than the alpha value of 0.05, indicating that the *p*-value is in the alpha area, we would reject the null hypothesis and accept the alternative hypothesis.



11.6.2 Example 11.5

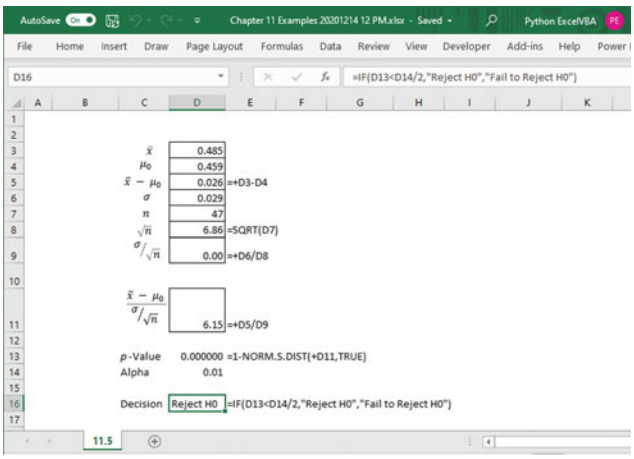
C. S. Patterson's study of a sample of 47 large public electric utilities with revenues of $300 million or more according to Moody's Manual (*Financial Management*, Summer 1984). Patterson's study focused on the financing practices and policies of these regulated utilities. Compilation of the actual debt ratios or long-term debt divided by total capital of the companies yielded the following results:

$$\overline{X} = 0.485, \quad s_x = 0.29$$

Test the following two-tailed hypothesis:

$$H_{0:\mu} = .459$$
$$H_{1:\mu} \neq .459$$

Below show the calculations done in Excel. From the calculation below, we would reject the null hypothesis.



Note that in the above calculation for the two-tail test, the format is the same as the one-tail test. The only difference in the worksheet format is in the decision-making process made in cell D16. Note that in cell D16, the *IF* function divides cell D14, which contains the alpha value, by 2.

11.7 One-Tailed Tests of Mean for Small Samples

For one-tailed tests of the mean for samples of less than 30, the *t*-test should be used. The *t*-test, which uses a *t*-distribution, looks like the following:

$$\frac{x - \mu_0}{s/\sqrt{n}} = 1$$

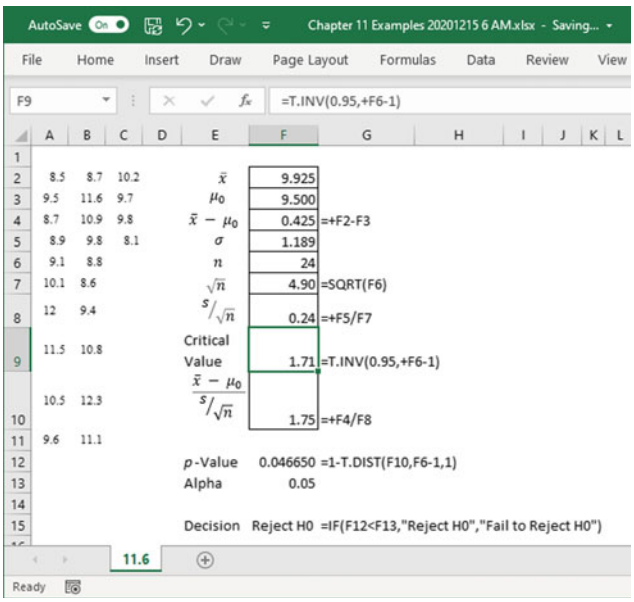
11.7.1 Example 11.6

United Van Lines Company is considering purchasing a large, new moving van. The sales agency has agreed to lease the truck to United Van Lines for 4 weeks (24 working days) on a trial basis. The main concern of United Van Lines is the miles per gallon (mpg) of gasoline that the van obtains on a typical moving day. The mpg values for the 24 trial days are.

8.5	9.5	8.7	8.9	9.1	10.1	12	11.5	10.5	9.6
8.7	11.6	10.9	9.8	8.8	8.6	9.4	10.8	12.3	11.1
10.2	9.7	9.8	8.1						

They want to know if the average was greater than 9.5. Our hypothesis would then be.

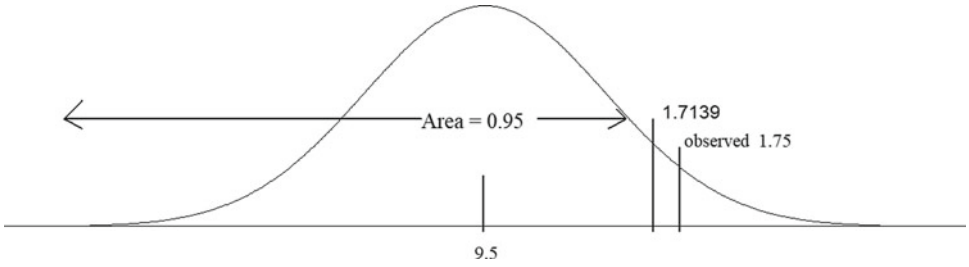
$$H_0 = 9.5$$
$$H_0 = 9.5$$
$$H_1 > 9.5$$
$$H_1 > 9.5$$



The critical value for 95% was 1.7109. The calculated value for the sample was 1.75. Since the calculated value was larger than the critical value, we reject the null hypothesis and accept the alternative hypothesis.

Using the *p*-value method, we can see that the calculated *p*-value of 0.047 is less than the alpha value of 0.05. Therefore, we can conclude that there is enough evidence to reject the null hypothesis and accept the alternative hypothesis.

Graphically this situation would look like the following:



11.8 Hypothesis Testing for a Population Proportion

If the sample size is large, the *Z*-test can be used to test the hypothesis for a population proportion. If the sample size is small, then the *t*-test should be used.

The test statistic for a two-tailed population proportion is

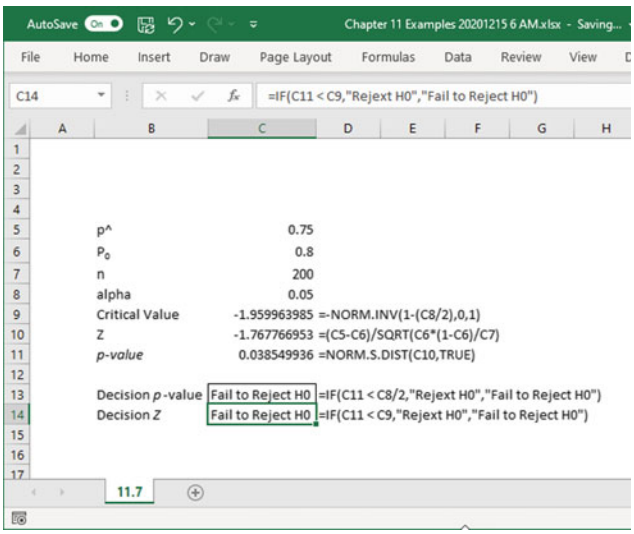
$$-z_{\alpha/2} > \frac{\widehat{P} - P_0}{\sqrt{P_0(1-P_0)/n}} \text{ or } z_{\alpha/2} > \frac{\widehat{P}S - P_0}{\sqrt{P_0(1-P_0)/n}}$$

11.8.1 Example 11.7

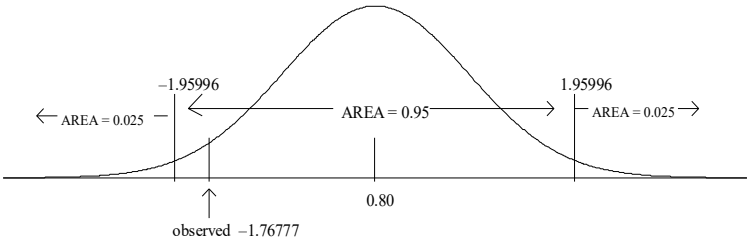
Francis Company is evaluating the promotability of its employees. That is, it is determining the proportion of employees whose ability, training, and supervisory experience qualify them for promotion to the next level of management. The Human Resources Director of Francis Company tells the president that 80% of the employees in the company are "promotable." However, a special committee appointed by the president finds that only 75% of the 200 employees who have been interviewed are qualified for promotion. Use this information to do a two-tailed hypothesis test at $\alpha = 5\%$.

The null and alternative hypotheses would be

$$H_0 : p = 0.80$$
$$H_1 : p \neq 0.80$$



Since the sample proportion is less than the hypothesized proportion, the test statistic will be negative. Because it will be negative, we will use the lower calculated Z-value. We cannot reject the null hypothesis because the lower Z-value for the sample is greater than the Z-value for the given alpha.



11.9 The Power of a Test and Power Function

The *power of a test* is the probability of rejecting H_0 when it is false. The probability is equal to $(1 - \beta)$, where β denotes the probability of a Type II error. Other things being equal, the greater the power of the test, the better the test.

Below are the power functions for a right-tail, a left-tail, and a two-tail test.

Right-Tail Power Function:

$$Power = P\left(Z > \frac{\mu_o - \mu_1}{\sigma_x/\sqrt{n}} + z_a\right)$$

Left-Tail Power Function:

$$Power = P\left(Z > \frac{\mu_o - \mu_1}{\sigma_x/\sqrt{n}} - z_a\right)$$

Two-Tail Power Function:

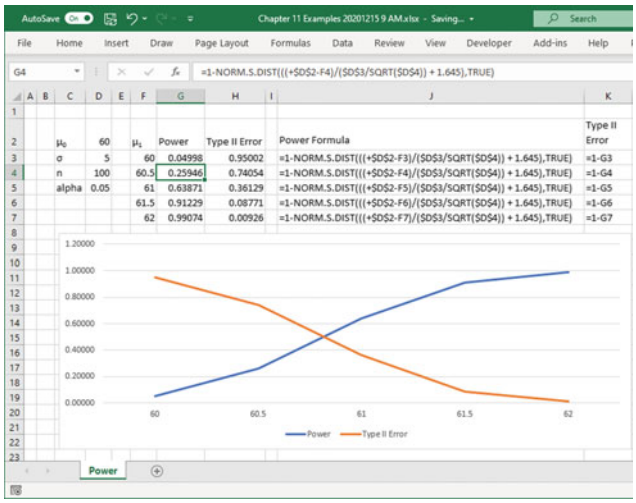
$$Power = P\left(Z > \frac{\mu_o - \mu_1}{\sigma_x/\sqrt{n}} + z_{a/2}\right) + P\left(Z > \frac{\mu_o - \mu_1}{\sigma_x/\sqrt{n}} - z_{a/2}\right)$$

11.9.1 Example 11.8

Say we want to test whether the average weight of 60-oz bags of cat food is equal to or smaller than 60 oz at significance level $\alpha = 0.05$. The null and alternative hypotheses can be stated as

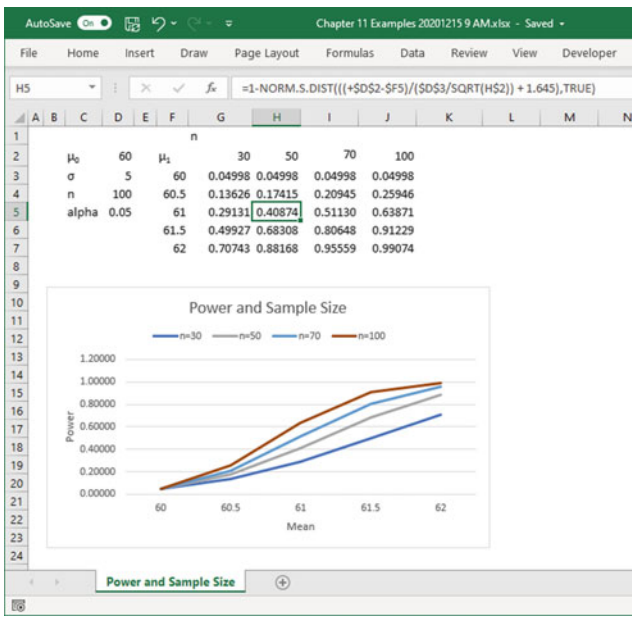
$$H_0 \mu = 60$$
$$H_1 \mu > 60$$

In addition, suppose we know that the sample size $n = 100$, sample mean $X = 59$, and standard deviation $s_x = 5$. The calculation for the power of μ_1 when $\mu_1 = 60, 60.5, 61, 61.5, 62$ is shown below:



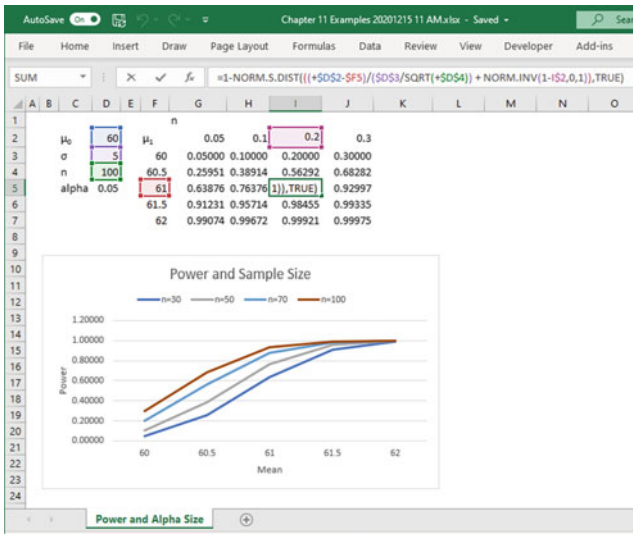
11.10 Power and Sample Size

In this section, we will demonstrate how the sample size affects the power of a test. Below shows how the sample size affects the power of the test. Below shows that as the sample size increases, the better the power test gets.



11.11 Power and Alpha Size

In this section, we will demonstrate how the alpha size affects the power of a test. Below shows how the alpha size affects the power of the test. Below shows that as the alpha size increases, the better the power of the test. The value of cell I5 is 0.87664.



11.12 Comparing the Average EPS of AAPL and MSFT in Python

In this section, we will see if the average EPS of two stocks are equal or not. We will use Python and the *t*-test to perform the this study.

```
[ ] import json
    from urllib.request import urlopen

def usr_input():
    ticker_1 = input('Enter 1st ticker:(defalut AAPL) ') or 'AAPL'
    ticker_2 = input('Enter 2nd ticker:(defalut MSFT) ') or 'MSFT'
    return ticker_1,ticker_2

[ ] #Data source for EPS of ticks
    URL = "https://www.alphavantage.co/query?function=EARNINGS&"
    API = "T2X24BNDA3Y3TK49"

[ ] ticker_1, ticker_2 = usr_input()

    Enter 1st ticker:(defalut AAPL) AAPL
    Enter 2nd ticker:(defalut MSFT) MSFT
```

```
[ ] def get_jsonparsed_data(url):
        response = urlopen(url)
        data = response.read().decode("utf-8")
        return json.loads(data)

[ ] def get_dictdata(ticker):
      url = URL + "symbol=" + ticker + "&apikey=" + API
      data = get_jsonparsed_data(url)['annualEarnings']
      date = []
      eps = []
      for ele in data:
        date.append(ele['fiscalDateEnding'])
        eps.append(ele['reportedEPS'])
      return {ticker+' Date':date,ticker+' EPS':eps}
```

```
[ ] import pandas as pd
    import matplotlib.pyplot as plt
    import numpy as np
    from scipy import stats
    from tabulate import tabulate

pd.set_option('display.max_rows', None, 'display.max_columns', None)
pd.set_option('max_colwidth', 1000)
test1 = get_dictdata(ticker_1)
test2 = get_dictdata(ticker_2)
framedata1 = pd.DataFrame(test1)
framedata2 = pd.DataFrame(test2)
print(framedata1)
print('')
print(framedata2)
```

	AAPL Date	AAPL EPS
0	2021-03-31	1.68
1	2020-09-30	3.27
2	2019-09-30	2.98
3	2018-09-30	2.97
4	2017-09-30	2.3
5	2016-09-30	2.0675
6	2015-09-30	2.3
7	2014-09-30	1.6075
8	2013-09-30	1.415
9	2012-09-30	1.5775
10	2011-09-30	0.9875
11	2010-09-30	0.54
12	2009-09-30	0.2225
13	2008-09-30	0.38
14	2007-09-30	0.55
15	2006-09-30	0.33
16	2005-09-30	0.2
17	2004-09-30	0.05
18	2003-09-30	0.0112
19	2002-09-30	0.0198
20	2001-09-30	0.0061
21	2000-09-30	0.1168
22	1999-09-30	0.09
23	1998-09-30	0.06
24	1997-09-30	-0.1114
25	1996-09-30	-0.1082

[ ]

	MSFT Date	MSFT EPS
0	2021-03-31	3.85
1	2020-06-30	5.75
2	2019-06-30	4.75
3	2018-06-30	3.88
4	2017-06-30	3.31
5	2016-06-30	2.76
6	2015-06-30	2.46
7	2014-06-30	2.63
8	2013-06-30	2.67
9	2012-06-30	2.73
10	2011-06-30	2.69
11	2010-06-30	2.1
12	2009-06-30	1.7
13	2008-06-30	1.88
14	2007-06-30	1.49
15	2006-06-30	1.27
16	2005-06-30	1.3
17	2004-06-30	1.26
18	2003-06-30	1.05
19	2002-06-30	0.93
20	2001-06-30	0.91
21	2000-06-30	0.87
22	1999-06-30	0.71
23	1998-06-30	0.46
24	1997-06-30	0.33
25	1996-06-30	0.215

```
# changing the format of eps
eps_1 = test1[ticker_1 + " EPS"]
eps_2 = test2[ticker_2 + " EPS"]
eps_1 = [float(i) for i in eps_1]
eps_2 = [float(i) for i in eps_2]
print('Two sample T test result '+ticker_1+' vs '+ticker_2)
print('')
stat,pval = stats.ttest_ind(eps_1,eps_2)
Number_1 = len(eps_1)
Number_2 = len(eps_2)
mean_1 = np.mean(eps_1)
mean_2 = np.mean(eps_2)
std_1 = np.std(eps_1)
std_2 = np.std(eps_2)
metric = {'stock Name':[ticker_1,ticker_2],'N':[Number_1,Number_2],'Mean':[mean_1,mean_2],'standard deviation':[std_1,std_2]}
metric = pd.DataFrame(metric)
print(tabulate(metric,headers=metric.columns,tablefmt="github",showindex=False))
print('')
print('T test for hypothesis for equal mean on eps for '+ticker_1+' and '+ticker_2)
print('T = 0.05, P = {}, DF = {}, with assumed same std'.format(np.round(pval,4),Number_1+Number_2-2))
if pval<0.05:
  print('we reject the Null hypothesis because T > P')
else:
  print('we can not reject the Null hypothesis because T <= P')
```

```
Two sample T test result AAPL vs MSFT

| stock Name   |   N |     Mean |   standard deviation |
|--------------|-----|----------|----------------------|
| AAPL         |  26 | 0.981223 |              1.07556 |
| MSFT         |  26 | 2.07519  |              1.3765  |

T test for hypothesis for equal mean on eps for AAPL and MSFT
T = 0.05, P = 0.0029, DF = 50, with assumed same std
we reject the Null hypothesis because T > P
```

From the Python code and t-test, we reject the null hypothesis that average mean EPS for AAPL and MSFT are equal.

11.13 Summary

In this chapter, we made inferences about a population parameter by checking if there is enough evidence to reject the null hypothesis. A Type I mistake is rejecting the null hypothesis when, in fact, it is true. A Type II mistake is accepting the null hypothesis when, in fact, it is false.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

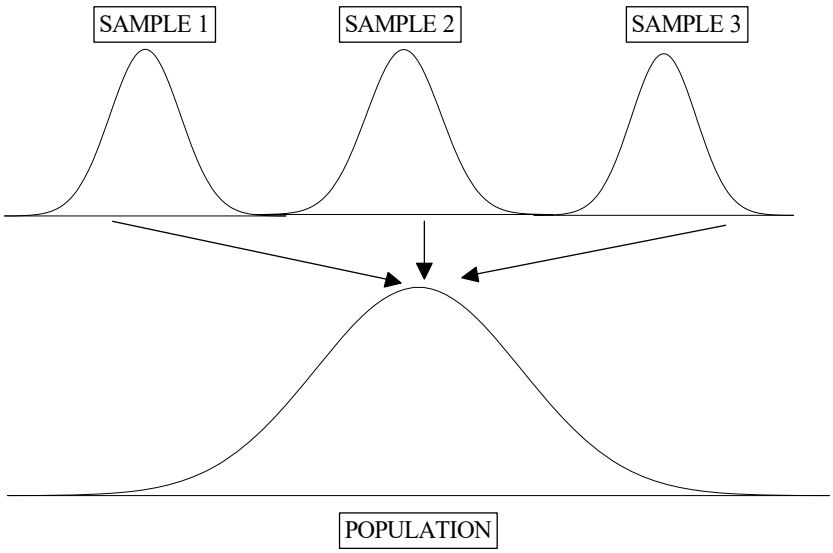
Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O'Reilly Media, 2017.

Analysis of Variance and Chi-Square Tests

12

12.1 Introduction

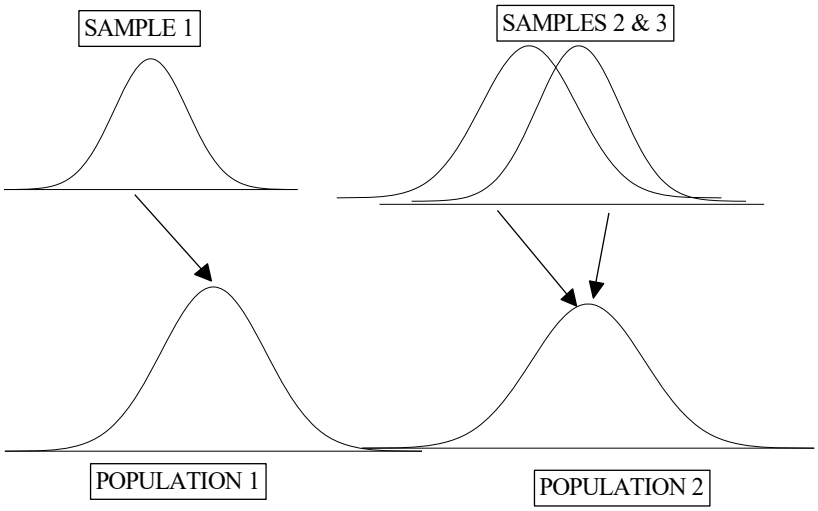
Often in statistics, we are interested in whether two or more samples have the same mean from the same population. Statistics uses the concept of analysis of variance to answer this question.



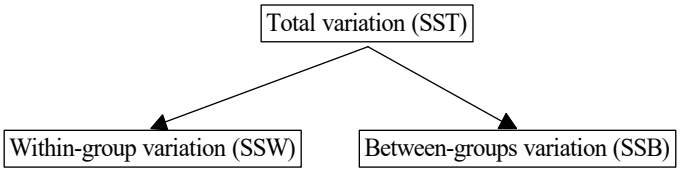
We must find out whether the group of samples comes from the same population, as shown in the above diagram or whether the group of samples does not come from the same population, as shown below:

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

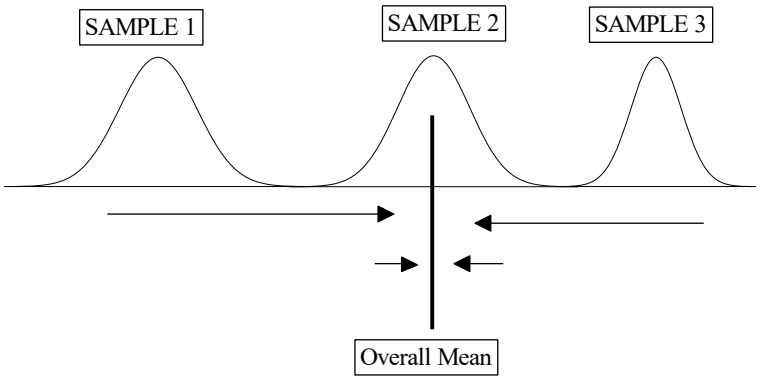
J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_12



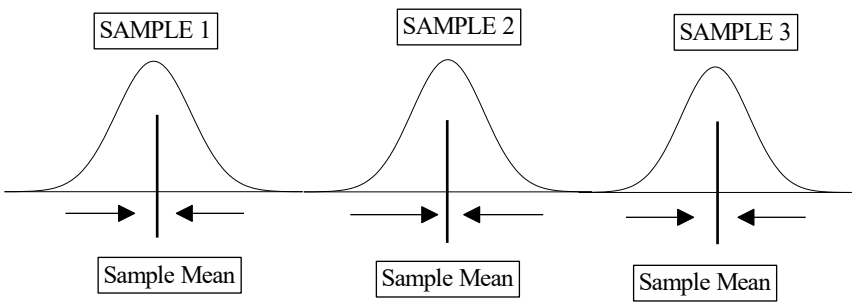
Analysis of variance, which is based on the concept of variation among samples, can be separated into two components, between-group variation and within-group variation. This is shown below:



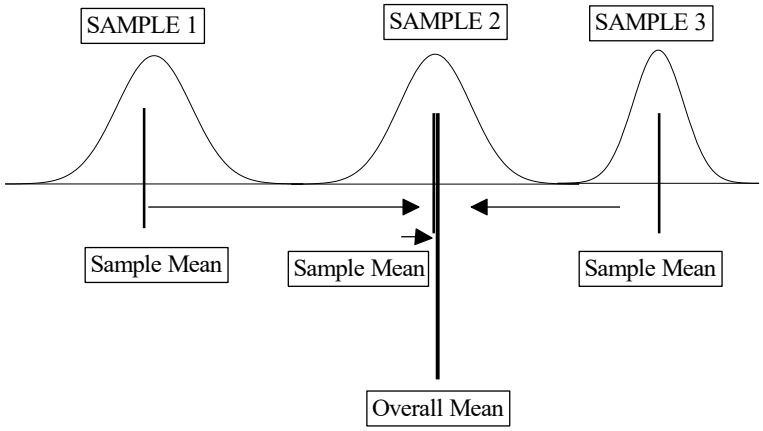
Total variation is the variation of all the samples about the overall mean of the samples.



Within-group variation is variation about the mean of each individual sample.



Between-group variation is a variation of group means about the overall mean.



If the between-group variation is significantly larger than the within-group variation, we can conclude that all the samples did not come from the same population. If the between-group variation is not significantly larger, we cannot conclude that all the samples did not come from the same population.

Therefore, based on the above concepts, the test statistic for the analysis of variance is

$$\frac{\text{between-groups variation}}{\text{within-groups variation}} = \frac{\text{between-groups mean square}}{\text{within-groups mean square}}$$

The above ratio follows an F distribution.

In the analysis of variance, the general null and alternative hypotheses are

H_0: all the samples have the same mean.
H_1: not all the samples have equal means.

This chapter is broken down into the following sections. In Sect. 12.2, we discuss a one-way analysis of variance, and in Sect. 12.3 we look at a two-way analysis of variance. In Sect. 12.4, we discuss the chi-square test. In Sect. 12.5, we look at goodness of fit and in Sect. 12.6 we discuss test of independence. In Sect. 12.7, we use the chi-square test and Python to determine if rate of return of Apple Inc. is normal distribution. Finally, in Sect. 12.8, we summarize the chapter.

12.2 One-Way Analysis of Variance

One-way analysis of variance studies the mean of two or more samples based on one classification.

12.2.1 Example 12.1

A research institution says that gasoline is gasoline. There is no difference among different brands of gasoline in terms of mileage per gallon. An independent consumer rights organization did an experiment on three different brands of gasoline. Twenty cars of the same condition were divided into three groups, and each group was filled with a different brand of gasoline. The test results show

Gasoline		
A	B	C
34	29	32
28	32	34
29	31	30
37	43	42
42	31	32
27	29	33
29	28	

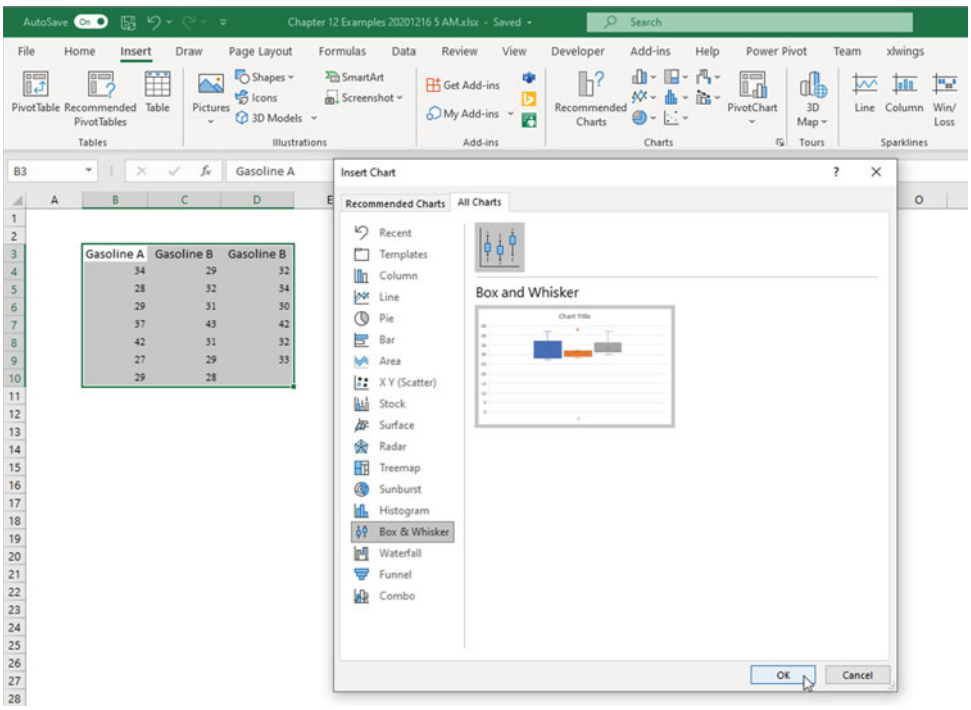
Is there any difference between the three brands of gasoline? Is there any significant difference at the 5% alpha level?

We will use the analysis of variance to see if there are any significant differences among the means. The null hypothesis and the alternative hypothesis would be used to perform the studies.

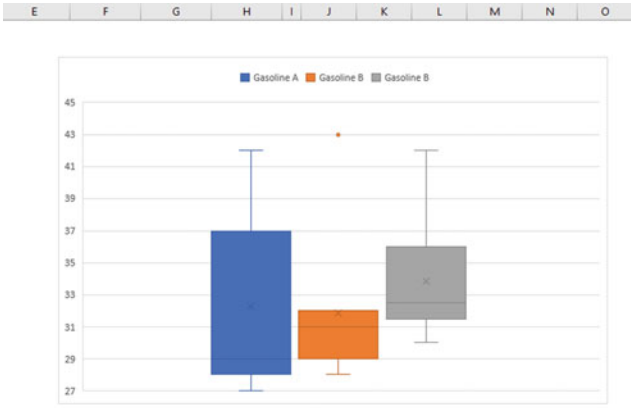
12.2.1.1 Box Plot

Let's first have a look at the box plots of the three types of gasoline to get an overall picture of the data.

To create a box plot, first, select the data and then choose Insert → Recommended Charts → All Charts → Box & Whisker.



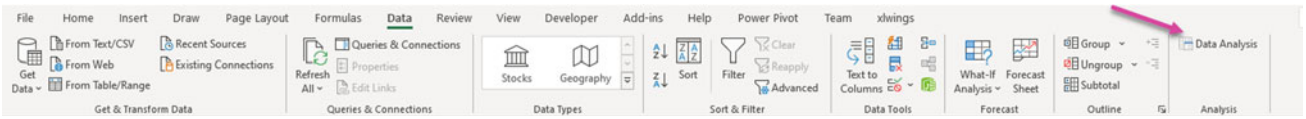
After formatting the box plot, the box plot looks like the following:



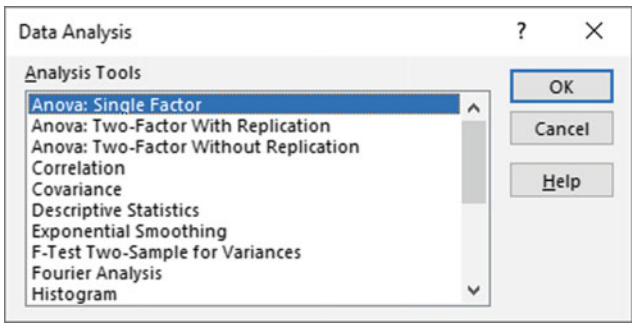
From the box plot, we can see that the three types of gasoline get similar mileage. This suggests to us that the three gasoline brands might even have the same mean. We will test this by doing a one-way analysis of variance test.

12.2.1.2 One-Way Analysis of Variance

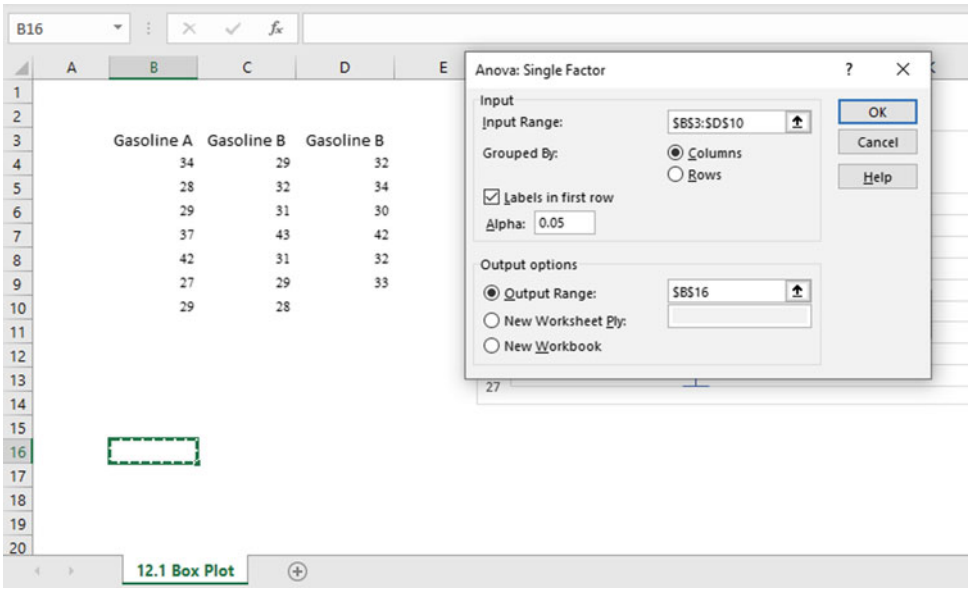
To do the one-way analysis of variance, we will use the *Data Analysis* in the *Data* tab.



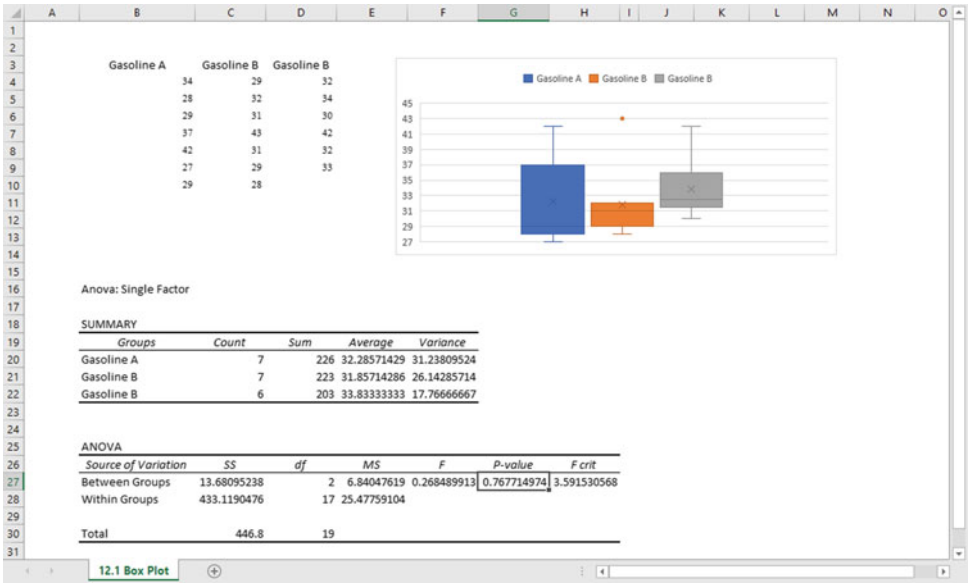
The *Data Analysis* menu item brings up the following dialog box:



We will choose the *ANOVA: Single Factor* item to do the one-way analysis of variance. Fill out the dialog box, as shown below:



The One-Way Analysis of Variance report is shown below:

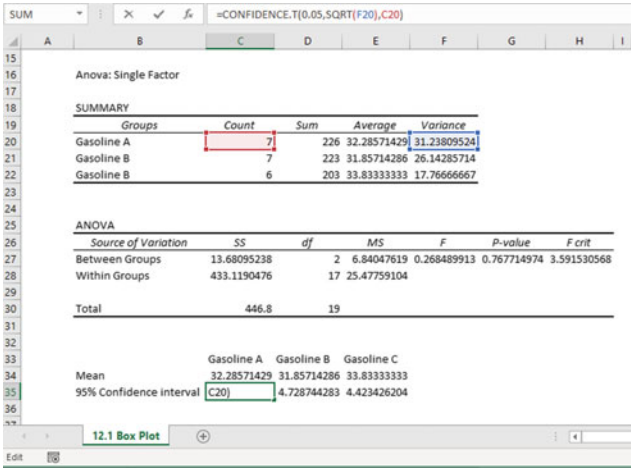


From the Excel output, we can see that the *p-value* for the one-way analysis is 0.7677. Since this *p-value* is greater than an alpha value of 0.05, we cannot reject the null hypothesis that the three gasoline brands have the same mean.

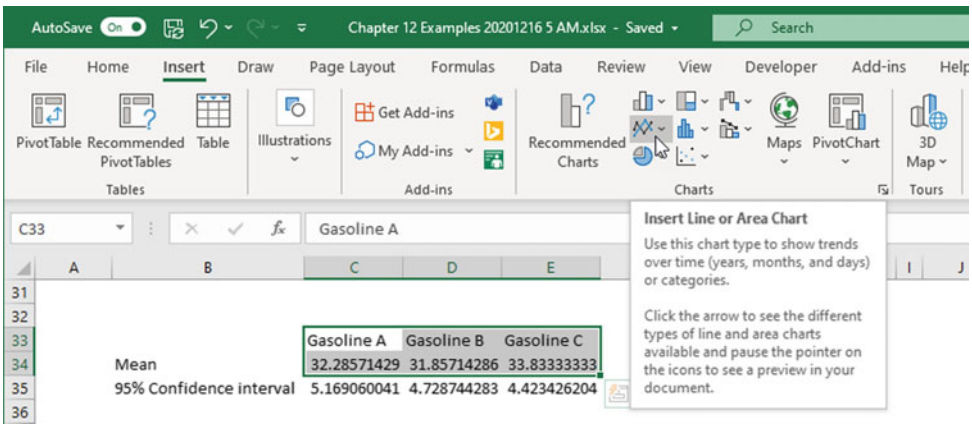
12.2.1.3 95% Confidence Interval for Mean

With Analysis of Variance reports, there are usually a 95% confidence interval graphs for the means. The following shows the step in creating a 95% confidence interval graph for the mean.

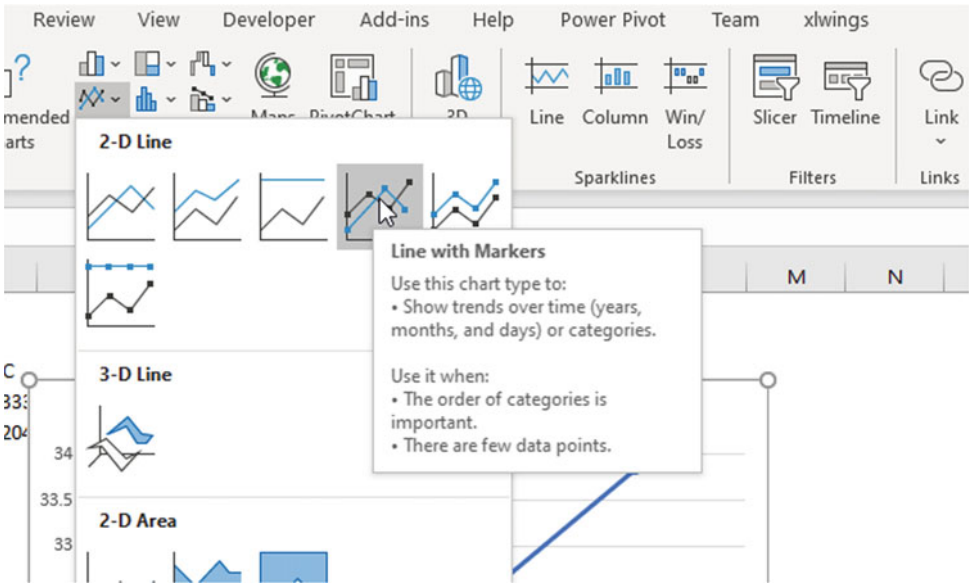
We first create a 95% confidence interval for the means for the gasoline. We use the Excel function *CONFIDENCE.T* to do this; this is done in row 35. The value for cell c35 is 5.169.



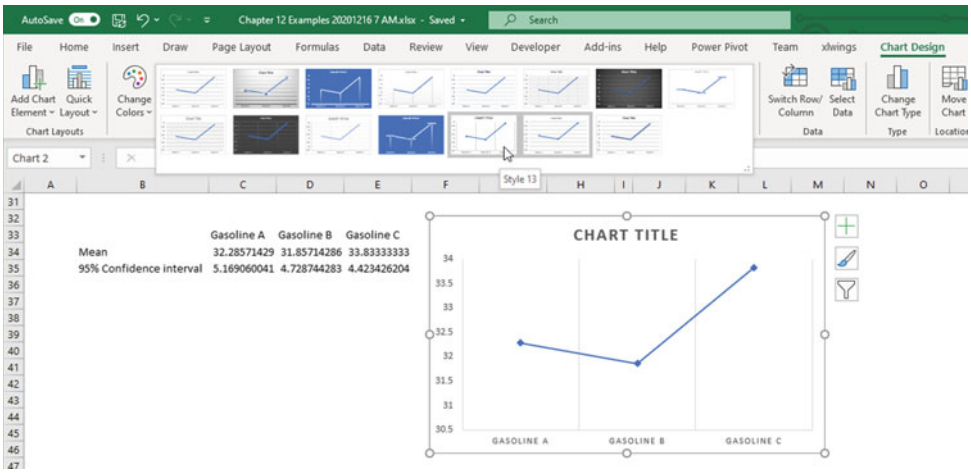
Next, we select cells c33 to e34 and then select *Insert Line or Area Chart.*



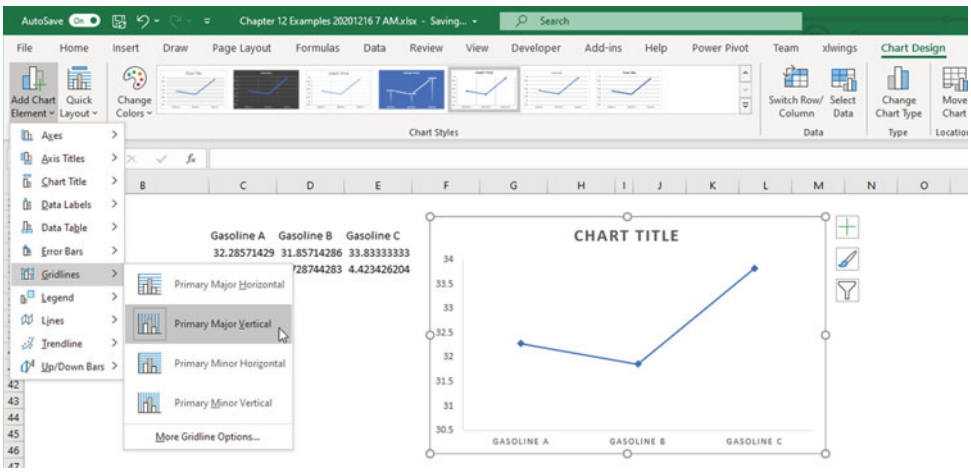
Next, choose the *Line with Markers* line type.



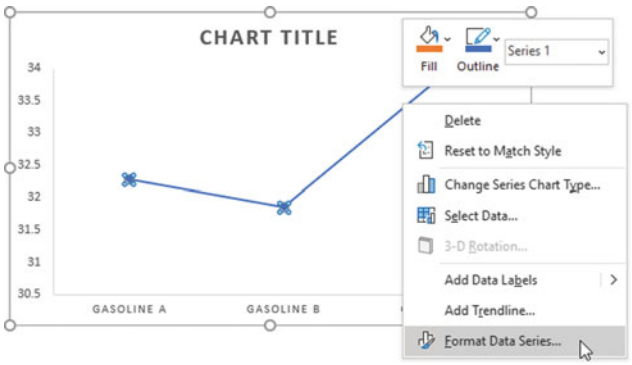
Next, in the *Chart Design* tab, choose style 13.



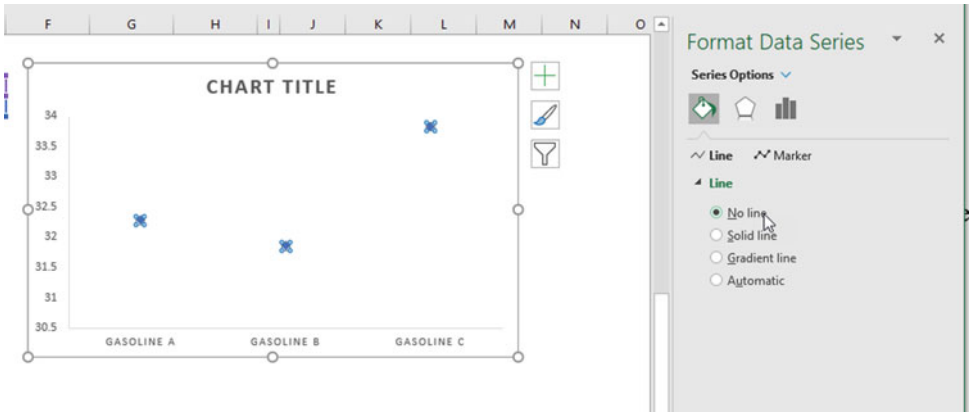
Next, while selecting the chart, do Chart Design → Add Chart Element → Gridlines → Primary Major Vertical to remove the vertical lines from the chart.



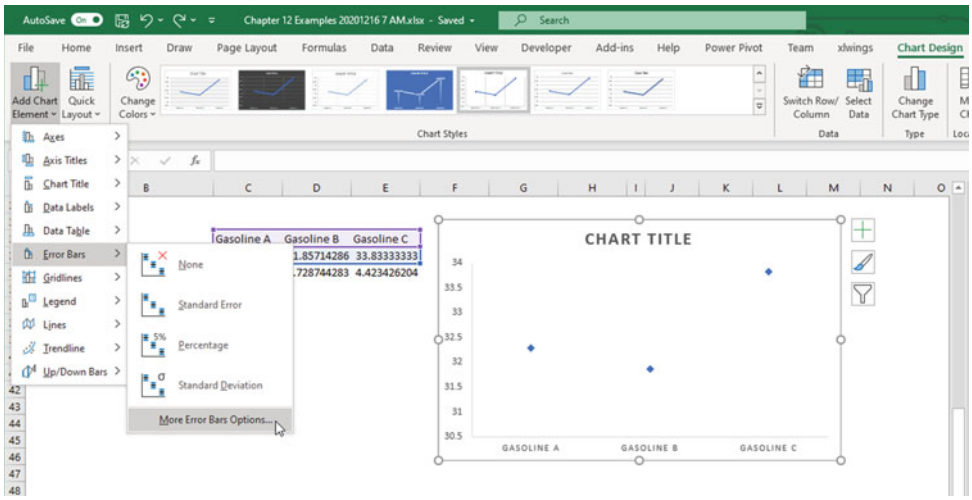
Next, we are going to remove the solid line by selecting the line and right-mouse click and select the *Format Data Series* option.



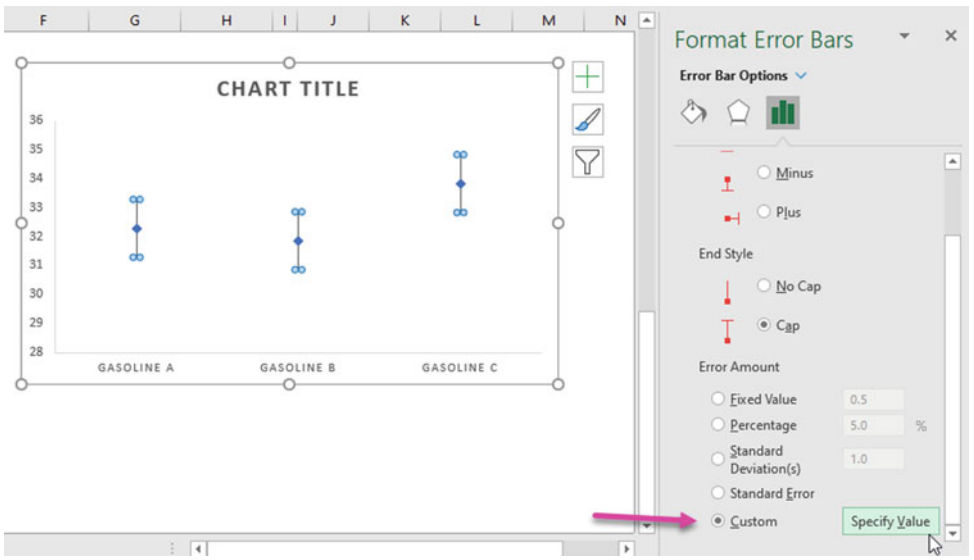
Next, select the *No Line* option.



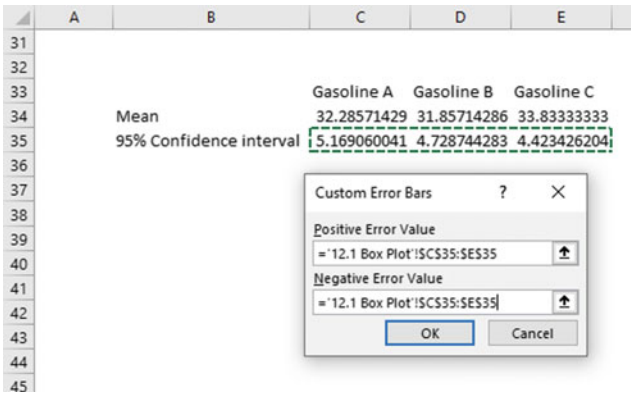
Next, choose Chart Design → Add Chart Element → Error Bars → More Error Bars Options.



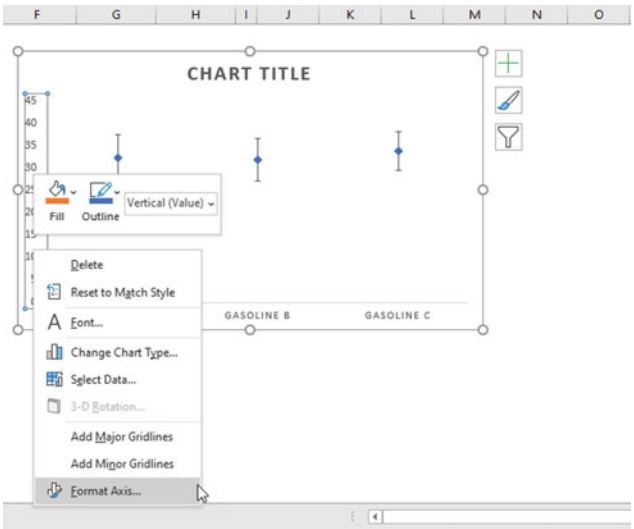
Next, choose Format Error Bars → Custom.



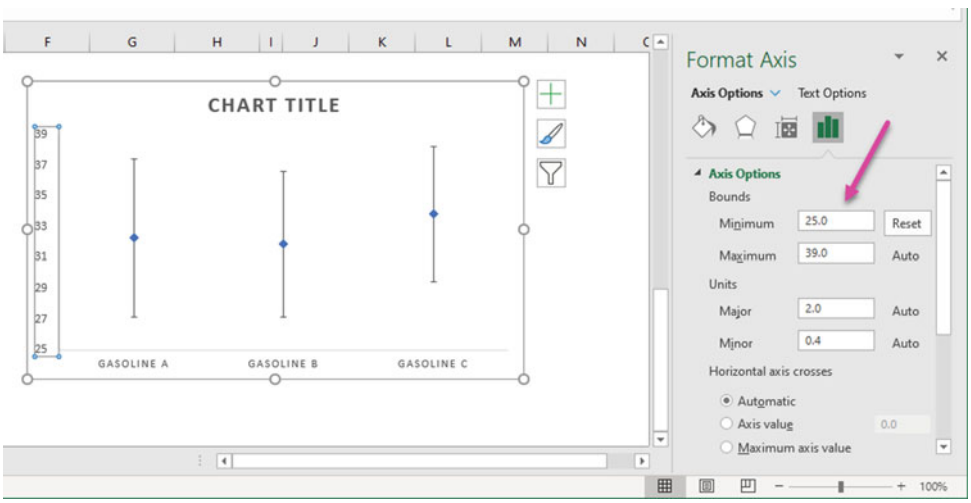
Next, select cells c35 to E35 for both the *Positive Error Value* and the *Negative Error Value*.



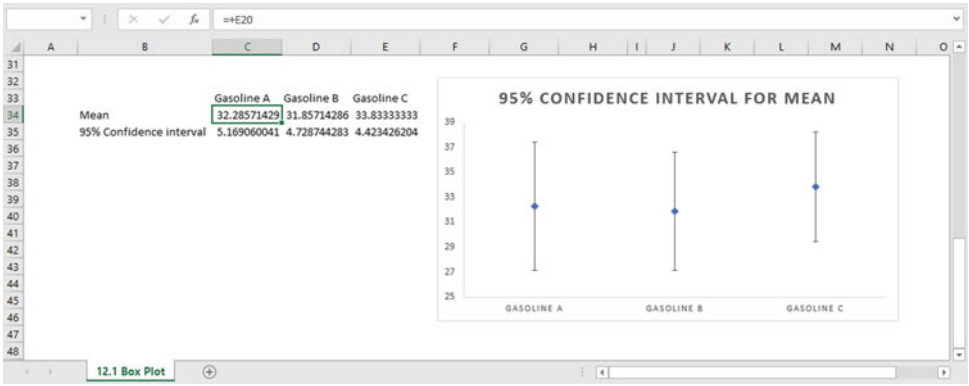
Next, right-mouse click on the axis and then choose *Format Axis*.



Next, set the Minimum to 25.

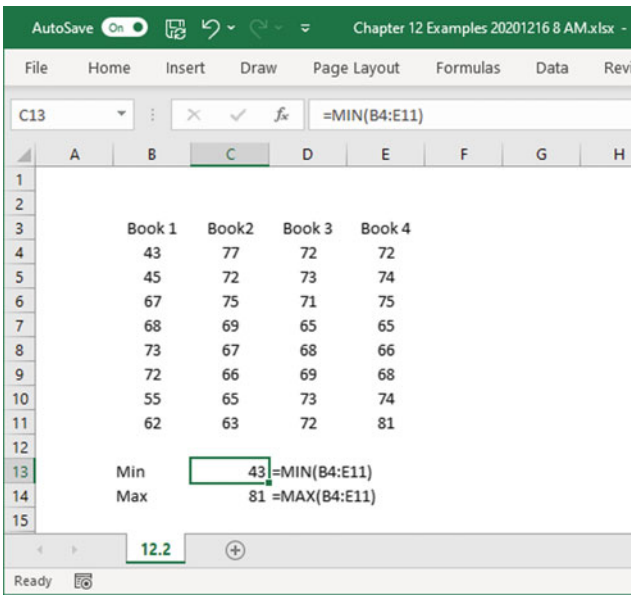


Below is the 95% confidence interval graph for the gasoline means.

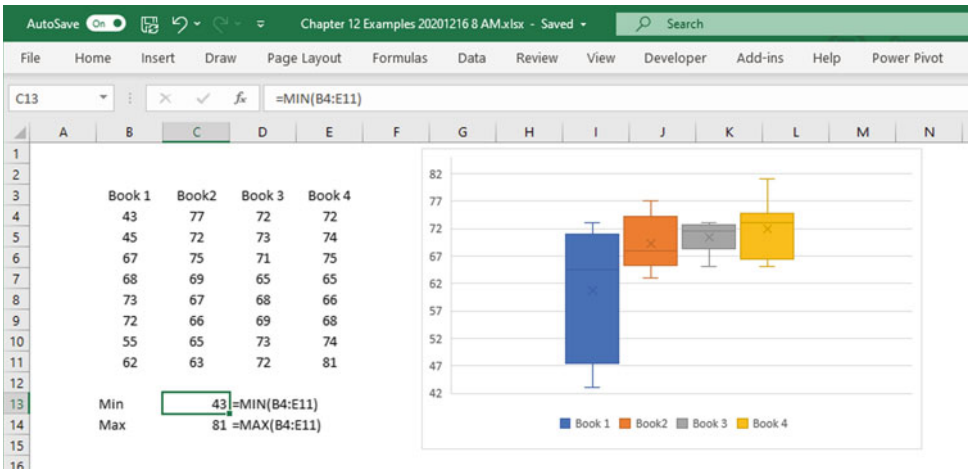


12.2.2 Example 12.2

There are many books to help people learn computer software packages. An instructor checked these books and found that they are all of similar quality. He picked four books and used a different book in each of his four classes. If the students all have the same average grades, he will use the cheapest book. The test results for students using the four books are



12.2.2.1 Box Plot



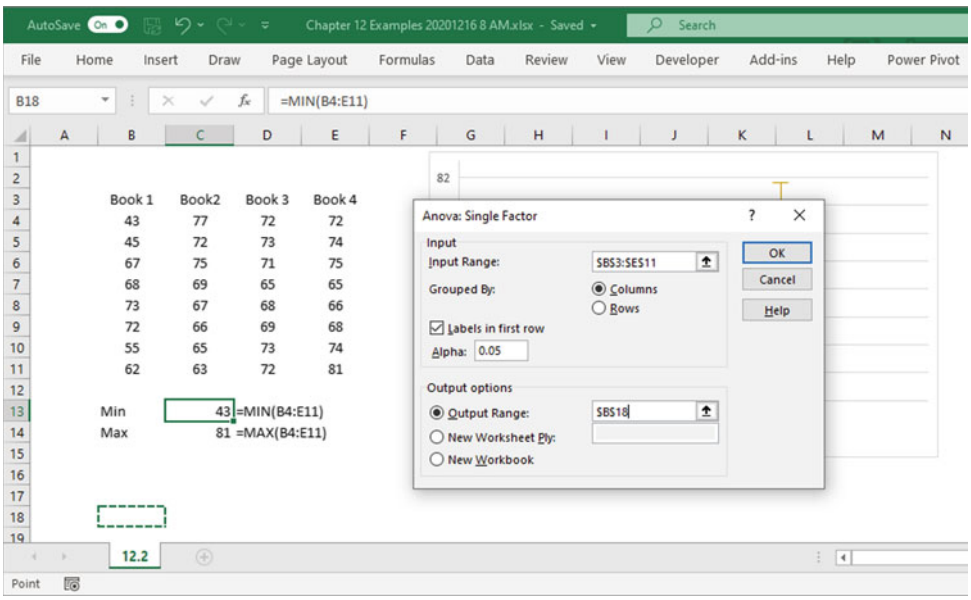
From the box plot of the four books, we can see that the plots are somewhat similar. This suggests to us that the four classes might have the same average grade. We will test this by doing a one-way analysis of variance calculation.

12.2.2.2 One-Way Analysis of Variance

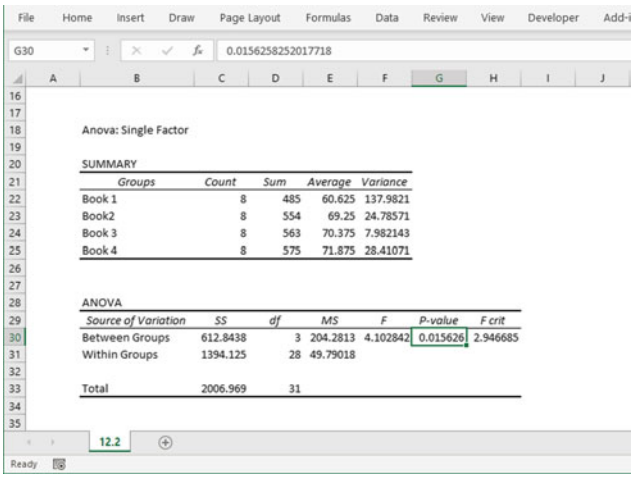
Our null and alternative hypotheses would be

H_0: book 1 grade average = book 2 grade average = book 3 grade average = book 4 grade average.
H_1: not all book grades averages are equal.

Choose Data → Data Analysis → Anova: Single Factor.



The ANOVA report is shown below:



Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Book 1	8	485	60.625	137.9821
Book2	8	554	69.25	24.78571
Book 3	8	563	70.375	7.982143
Book 4	8	575	71.875	28.41071

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	612.8438	3	204.2813	4.102842	0.015626	2.946685
Within Groups	1394.125	28	49.79018			
Total	2006.969	31				

From the output, we can see that the calculated p-value is 0.0156. Therefore, since the p-value is less than the alpha value of 0.05, we reject the null hypothesis that every class has the same average grade.

12.2.2.3 95% Confidence Interval for Mean

The 95% confidence interval for the mean grade for the books is given below:

	Book 1	Book 2	Book 3	Book 4
Mean	60.625	69.25	70.375	71.875
95% Confidence Interval	9.820387	4.162151	2.361984	4.456134

From the graph, we conclude that the book with a statistically different mean grade is book 1.

So far, we have looked at examples based on a single classification. The first example analyzed the mean based on gasoline. The second example analyzed the mean based on books. These classifications are called *factors*. In the gasoline example, we had three different types of gasoline. In the classroom example, we had four classes. The individual types of gasoline and types of books are called *levels.*

12.3 Two-Way Analysis of Variance

Doing an analysis of variance with one factor is called a one-way analysis of variance. We will now expand the analysis of variance and test to see if the means from different samples are the same or not based on two factors. Analysis of variance of two factors is called a two-way analysis of variance.

We can think of factors as influencing the value of the mean of each sample. For example, in the gasoline example, we were interested in miles per gallon. The factor that we thought might be influencing the mileage per gallon was the different brands. Now we are interested in two-factor analysis, two things affecting the mean that we are interested in.

12.3.1 Example 12.3

Suppose we are interested to know if years of work experience and geographical location influence annual salary. The data given below show salaries in thousands of dollars:

Region		Years of work experience		
		1	2	3
1	West	16, 16.5	19, 17	24, 25
2	Midwest	21, 20.5	20, 19	21, 22.5
3	Northeast	18, 19	21, 20.9	22, 21
4	South	13, 13.5	20, 20.8	25, 23

Here we are interested in the mean of the annual salary. We believe that two factors influence the mean: years of work experience and region of the country. It turns out that we should also look at the influence of the interaction of these two factors.

The variability of annual salary can thus be broken down in the following way:

$$\begin{aligned}\text{Total Variation} = {} & \text{Variation of Factor A} + \text{Variation of Factor B} \\ & + \text{Variation of Interaction of Factors AB} + \text{Random Error}\end{aligned}$$

In a two-way analysis of variance, there are three tests in which we are interested. In our salary example, they would be the following:

1. To test the hypothesis of no difference due to years of working experience:

H_0: mean year 1 = mean year 2 = mean year 3.
H_1: not all mean years are equal.

The test would follow an F distribution and would have the following ratio:

$$F = \frac{\text{variation of factor A(years of working experience)}}{\text{random error}}$$

2. To test the hypothesis of no difference due to geographic region:

H_0: mean region 1 = mean region 2 = mean region 3.
H_1: not all mean regions are equal.

The test would follow an F distribution and would have the following ratio:

$$F = \frac{\text{variation of factor B(geographic region)}}{\text{random error}}$$

3. To test the hypothesis of no difference due to interaction of work experience and region:

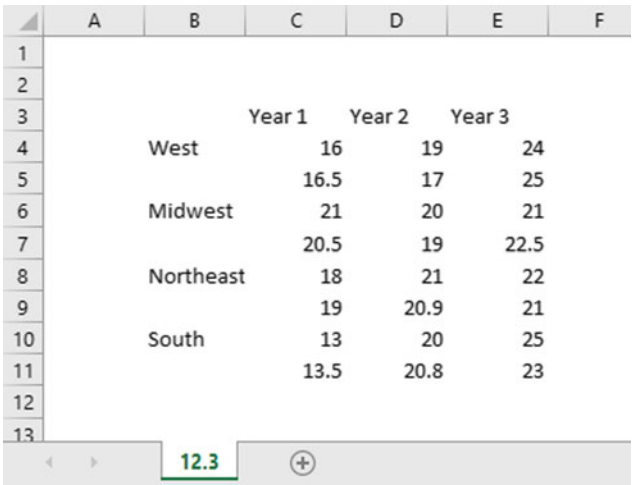
H_0: interaction of work experience and geographic region has no effect.
H_1: interaction of work experience and geographic region has an effect.

The test would follow an F distribution and would have the following ratio:

$$F = \frac{\text{variation of interaction of factor A and factor B(experience and region)}}{\text{random error}}$$

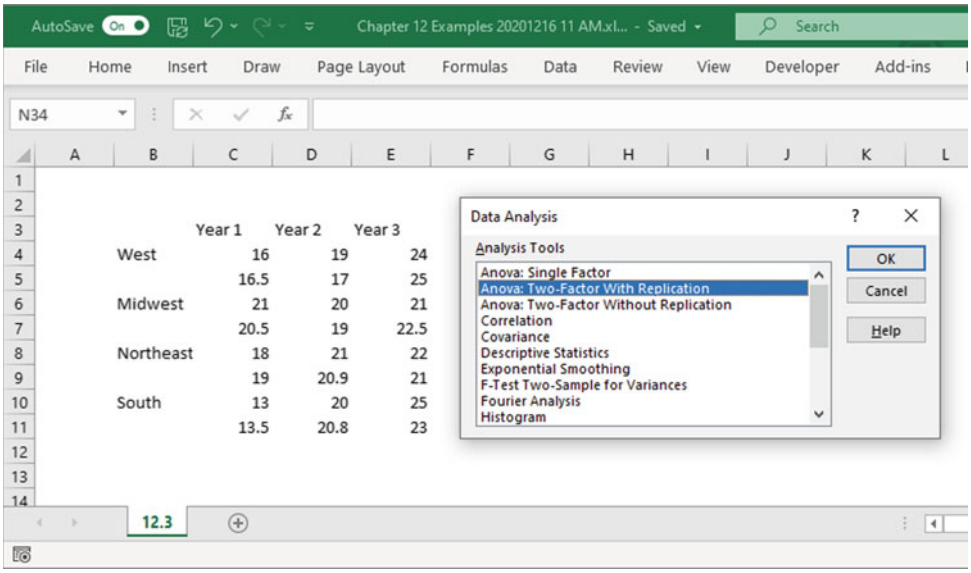
12.3.1.1 Two-Way Analysis of Variance

To do the two-way analysis of variance report in Excel, we will need to enter the data as shown below:

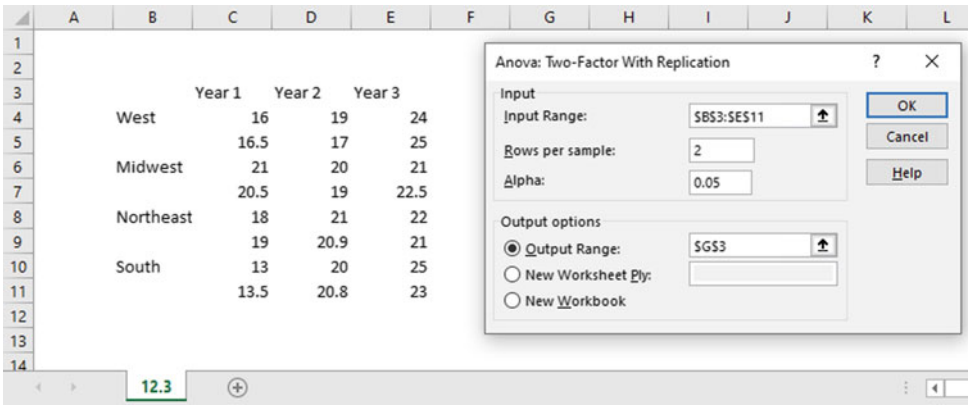


	A	B	C	D	E	F
1						
2						
3			Year 1	Year 2	Year 3	
4		West	16	19	24	
5			16.5	17	25	
6		Midwest	21	20	21	
7			20.5	19	22.5	
8		Northeast	18	21	22	
9			19	20.9	21	
10		South	13	20	25	
11			13.5	20.8	23	
12						
13						

There are two ANOVA options in Excel's data analysis box. We will choose the one with replication because there are two rows per region.

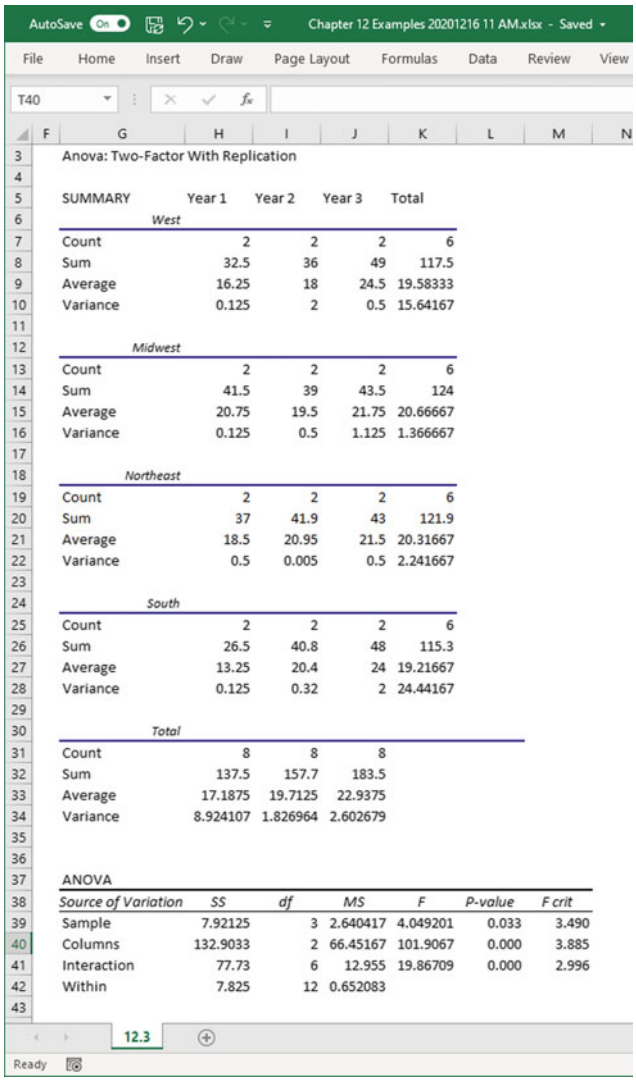


Fill out the *Anova: Two-Factor With Replication* dialog box as shown below:



The following is the *Anova*: *Two-Factor With Replication* report produced by Excel.
Looking at the Anova report, we have the following conclusions:

1. We conclude that the geographic region does affect annual salary because the calculated p-value 0.033 is less than the alpha value of interest, which is 0.05.
2. We conclude that work experience does affect annual salary because the calculated p-value 0.00 is less than the alpha value of interest, which is 0.05.
3. We conclude that the interaction of work experience and geographic regions does affect annual salary because the calculated p-value of 0.00 is less than the alpha value of interest, which is 0.05.



Anova: Two-Factor With Replication

SUMMARY	Year 1	Year 2	Year 3	Total
West				
Count	2	2	2	6
Sum	32.5	36	49	117.5
Average	16.25	18	24.5	19.58333
Variance	0.125	2	0.5	15.64167
Midwest				
Count	2	2	2	6
Sum	41.5	39	43.5	124
Average	20.75	19.5	21.75	20.66667
Variance	0.125	0.5	1.125	1.366667
Northeast				
Count	2	2	2	6
Sum	37	41.9	43	121.9
Average	18.5	20.95	21.5	20.31667
Variance	0.5	0.005	0.5	2.241667
South				
Count	2	2	2	6
Sum	26.5	40.8	48	115.3
Average	13.25	20.4	24	19.21667
Variance	0.125	0.32	2	24.44167
Total				
Count	8	8	8	
Sum	137.5	157.7	183.5	
Average	17.1875	19.7125	22.9375	
Variance	8.924107	1.826964	2.602679	

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Sample	7.92125	3	2.640417	4.049201	0.033	3.490
Columns	132.9033	2	66.45167	101.9067	0.000	3.885
Interaction	77.73	6	12.955	19.86709	0.000	2.996
Within	7.825	12	0.652083			

12.4 Chi-Square Test

This section will use the chi-square test to test the goodness of fit and independence of a data set.

12.5 Goodness of Fit

This section will show how the chi-square statistic can be used to the appropriateness of a distribution—its goodness of fit for a set of data. Goodness-of-fit tests are designed to study the frequency distribution to determine whether a set of data are generated from a certain probability distribution, such as the uniform, binomial, Poisson, or normal distribution.

The chi-square statistic for determining whether the data follow a specific probability distribution is

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{\left(\int_i^0 - \int_i^e\right)^2}{\int_i^e}$$

where

$\int_i^0$ = observed frequency,
$\int_i^e$ = expected frequency,
k = number of groups,
χ^2_{k-1} = chi-square statistic with $(k - 1)$ degrees of freedom.

12.5.1 Example 12.4

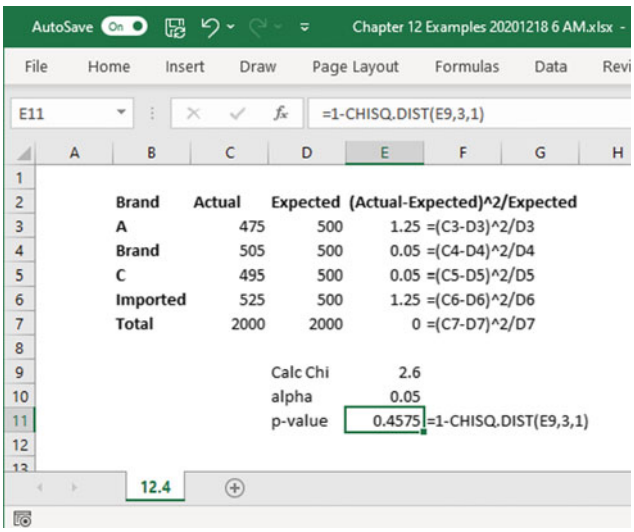
A marketing manager wants to test his belief that four different categories of cars share the auto market equally. These four categories of cars are brand A, brand B, brand C, and imported cars. He sends out 2000 questionnaires to car owners throughout the nations and receives the following responses:

Brand	Number of owners
A	475
B	505
C	495
Import	525
Total	2000

Test the following hypotheses:

H_0: Same market shares (uniform distribution).
H_1: Different market shares.

Below are the calculations that are done in Excel. We fail to reject the null hypothesis because the calculated p-value, 0.4575, is greater than the alpha value, 0.05.



12.6 Test of Independence

12.6.1 Example 12.5

We are now interested in checking to see whether two variables are independent of each other or not.

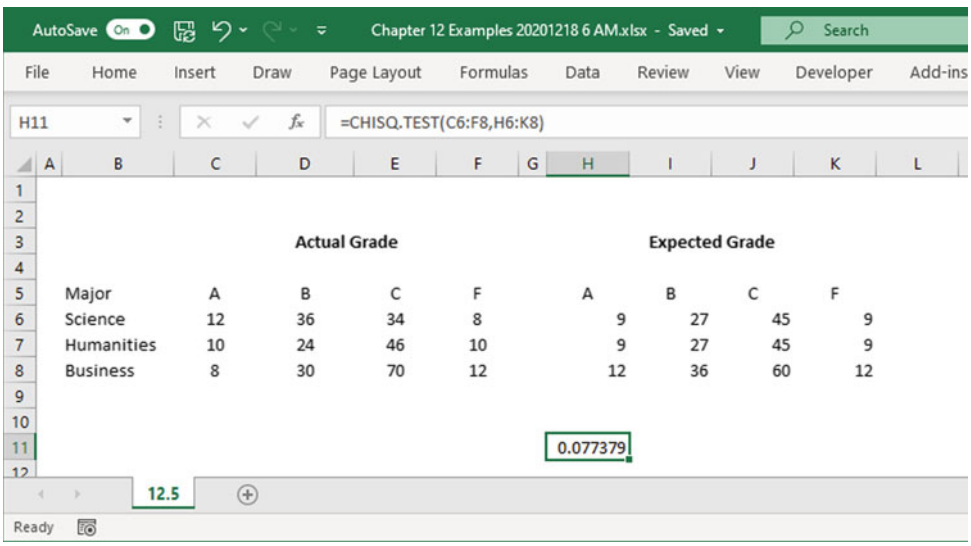
Suppose someone has asked us if majors and grades are related. The data for this question are shown below:

Major	Grades			
	A	B	C	F
Science	12	36	34	8
Humanities	10	24	46	10
Business	8	30	70	12

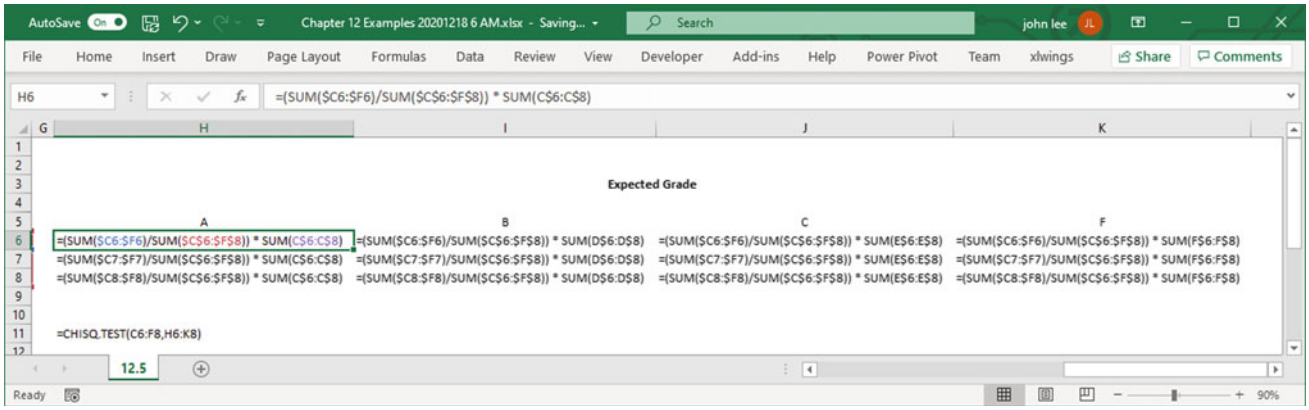
The hypotheses would be

H_0: major and grades are independent.
H_1: major and grades are not independent.

For this problem, we would follow a chi-square test.



Below shows how the above worksheet is set up in Excel.



In doing the calculations in Excel, we use the *chisq.test* function. This function returns a p-value. The calculated p-value in Excel is 0.07739. Since the p-value is greater than the alpha value of 0.05, we cannot reject the null hypothesis that major and grades are independent.

12.7 Using the Chi-Square Test and Python to Determine if the Rate of Return of Apple Inc. Is a Normal Distribution

```
!pip install requests_html
!pip install requests
!pip install bs4
!pip install yahoo_fin
```

```
[11] import pandas as pd
     import numpy as np
     from yahoo_fin.stock_info import *
     import matplotlib.pyplot as plt
     import matplotlib.ticker as mtick
     from tabulate import tabulate
     import scipy.stats as st
     import requests
     from bs4 import BeautifulSoup
     import datetime as dt#get current date
```

```
[12] def calculate_ret(ticker,start,end,period):
       data = get_data(ticker,start_date=start,end_date=end,index_as_date=True,interval=period)['adjclose'].dropna()
       ret = data.pct_change()[1:]
       return ret
```

```
def Plot(hist_ret,stock_name,Start,End):
  plt.figure(figsize=(15,6))
  plt.gca().xaxis.set_major_formatter(mtick.PercentFormatter())
  (count,bins,dum)=plt.hist(hist_ret*100)
  plt.suptitle('   hist for '+stock_name,fontsize = 24)
  plt.xlabel('Return',fontsize = 15)
  plt.ylabel('Frequency',fontsize = 15)

  plt.title(Start+' to '+End,fontsize = 16)
  bins = np.floor(bins)
  tuples = []
  for ele in range(len(bins)-1):
    tuple_range = (bins[ele],bins[ele+1])
    tuples.append(tuple_range)
  print('Estimated mean = '+str(round(hist_ret.mean(),4)))
  print('Estimated std = '+str(round(hist_ret.std(),4)))
  return tuples,count,bins
```

```
def html_beautifulsoup(url):
    response = requests.get(url)
    html = response.content

    #beautifulsoup html will not look like soure
    #review beautifulsoup html to use find and findall
    soup = BeautifulSoup(html,"lxml")
    return soup

#soup_html: soup html is not same as source html
def yahoo_description(soup_html):
    element = soup_html.find('h1',class_='D(ib) Fz(18px)')
    return element.text
```

```
def usr_input():
    previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
    Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
    years_ago = dt.date.today()-dt.timedelta(days=1852)
    Default_start = years_ago.strftime("%Y-%m-%d")
    Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
    Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
    End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
    T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'
    print("")

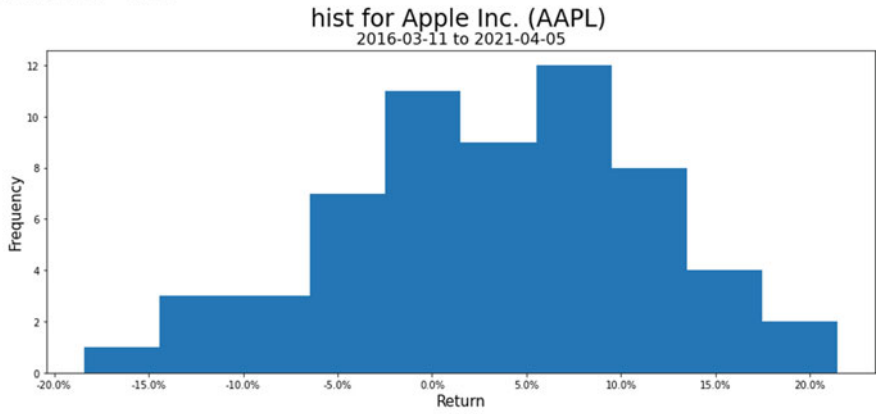
    soup_html = html_beautifulsoup("https://finance.yahoo.com/quote/"  + Ticker)
    stock_name = yahoo_description(soup_html)

    return Ticker,Start,End,T_interval,stock_name
```

```
[21] Ticker,Start,End,T_interval,stock_name = usr_input()
     ret = calculate_ret(Ticker,Start,End,T_interval)
     tuples,count,bins=Plot(ret,stock_name,Start,End)
```

```
Please enter ticker(Default = JNJ)-->AAPL
Enter start date(Default = 5 Years ago, 2016-03-11)-->2016-03-11
Enter end date(Default = Yesterday, 2021-04-05)-->2021-04-05
Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo

Estimated mean = 0.0326
Estimated std = 0.0831
```




```
dict_data = pd.Series(count,index=tuples)
frame_data = dict_data.to_frame()
frame_data.columns = ['frequency']
#normalized X
normd_X = []
for ele in range(len(bins)):
  normd_ele = round((bins[ele]/100.0-ret.mean())/ret.std(),2)
  normd_X.append(normd_ele)
#calculate cdf
cdf = np.round(st.norm.cdf(normd_X),4)
#shade under cdf
shade_cdf = np.diff(cdf)
#expected frequency
freq = np.multiply(shade_cdf,len(ret))
#difference of expected freq and actual freq
diff = count - freq
#output
output = np.divide(pow(diff,2),freq)
output[output==np.inf] = 0
frame_data.insert(1, "X", bins[:-1], True)
frame_data.insert(2, "Z", normd_X[:-1], True)
frame_data.insert(3, "CDF", cdf[:-1], True)
frame_data.insert(4, "Area P", shade_cdf, True)
frame_data.insert(5, "Expect freq", freq, True)
frame_data.insert(6, "(Fo - Fe)", diff, True)
frame_data.insert(7, "(Fo - Fe)\u00b2 /Fe", output, True)
print(tabulate(frame_data,frame_data.columns))
```

	frequency	X	Z	CDF	Area P	Expect freq	(Fo - Fe)	(Fo - Fe)² /Fe
(-19.0, -15.0)	1	-19	-2.68	0.0037	0.0102	0.612	0.388	0.245987
(-15.0, -11.0)	3	-15	-2.2	0.0139	0.0288	1.728	1.272	0.936333
(-11.0, -7.0)	3	-11	-1.72	0.0427	0.0666	3.996	-0.996	0.248252
(-7.0, -3.0)	7	-7	-1.23	0.1093	0.1173	7.038	-0.038	0.000205172
(-3.0, 1.0)	11	-3	-0.75	0.2266	0.167	10.02	0.98	0.0958483
(1.0, 5.0)	9	1	-0.27	0.3936	0.1896	11.376	-2.376	0.496253
(5.0, 9.0)	12	5	0.21	0.5832	0.1717	10.302	1.698	0.279868
(9.0, 13.0)	8	9	0.69	0.7549	0.1241	7.446	0.554	0.0412189
(13.0, 17.0)	4	13	1.17	0.879	0.0715	4.29	-0.29	0.0196037
(17.0, 21.0)	2	17	1.65	0.9505	0.0329	1.974	0.026	0.000342452

chi_square test

```
[23] chi_stat = np.round(st.chi2.ppf(0.95,df = len(count)-1),4)
     test_stat = np.round(sum(output),4)
     result = ''
     print("At 5% level of significance")

     if test_stat>chi_stat:
       print('We conclude that the return is not normally distributed with mean {:.4f},and variance of {:.4f}'.format(ret.mea
       print('The statistic '+str(test_stat)+' > then critical value of '+str(chi_stat))
     else:
       print('The statistic '+str(test_stat)+' <= then critical value of '+str(chi_stat))
       print('We conclude that the return is normally distributed with mean {:.4f},and variance of {:.4f}'.format(ret.mean(),

     At 5% level of significance
     The statistic 2.3639 <= then critical value of 16.919
     We conclude that the return is normally distributed with mean 0.0326,and variance of 0.0831
```

From the chi-square test, we conclude that the rate of return for Apple Inc. for the period March 3, 2016 to April 5, 2021 is normally distributed.

12.8 Summary

In this chapter, we use either one-way or two-way ANOVA to see if several sample means are statistically different from each other. In addition, we discussed how the chi-square statistic could be used to test whether two variables are independent of each other and where a set of data comes from a specific distribution (goodness of fit).

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.” Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.” BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.” O’Reilly Media, 2017.

13 Simple Linear Regression and the Correlation Coefficient

13.1 Introduction

In the previous chapters, we have been primarily interested in analyzing single variables. In this chapter, we are interested in analyzing two variables and the relationship between them. The two techniques that we will look at are regression analysis and correlation analysis.

This chapter is broken down into the following sections. In Sect. 13.2, we discuss regression analysis, and in Sect. 13.3 we look at retrieving data using power query. In Sect. 13.4, we combine power query data sets. In Sect. 13.5, we discuss scatter charts. Section 13.6 looks at deterministic relationship and stochastic relationship. Least square method is talked about in Sect. 13.7. The standard assumptions for linear regression are discussed in Sect. 13.8. Section 13.9 discusses the standard error of estimate, and the coefficient of determination is discussed in Sect. 13.10. The correlation coefficient is discussed in Sect. 13.11, and regression analysis in Excel is discussed in Sect. 13.12. INTERCEPT and SLOPE Excel functions are talked about in Sect. 13.13, and oil and gasoline regression analysis in Python is discussed in Sect. 13.14. Finally, Sect. 13.15 summarizes the chapter.

13.2 Regression Analysis

There are many types of relationships between two or more variables. Some of the types of relationships between two variables are linear, logarithmic, and polynomial. In the next four chapters, we will be studying the linear relationships between two or more variables. We will use regression analysis in studying linear relationships.

Mathematically, a linear equation looks like the following:

$$y = a + bx$$

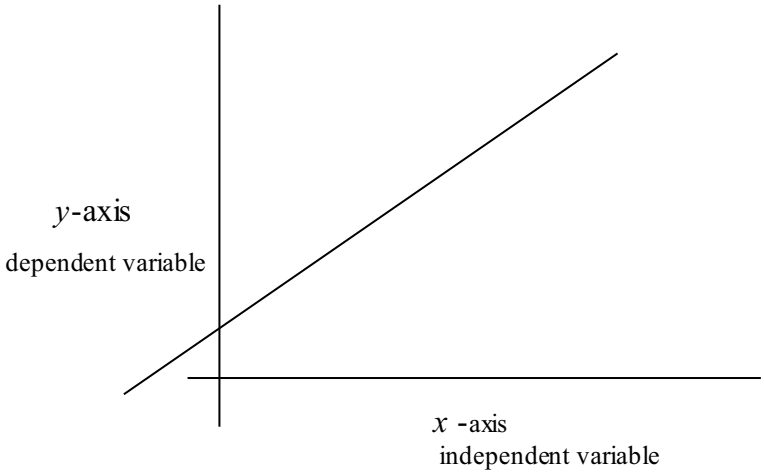
The two variables of interest in this equation are x and y. In this equation, x is called the independent variable and y is called the dependent variable.

In regression analysis, we are modeling the relationship between two variables and then using the model to predict the value of the dependent variable from the value of the independent variable.

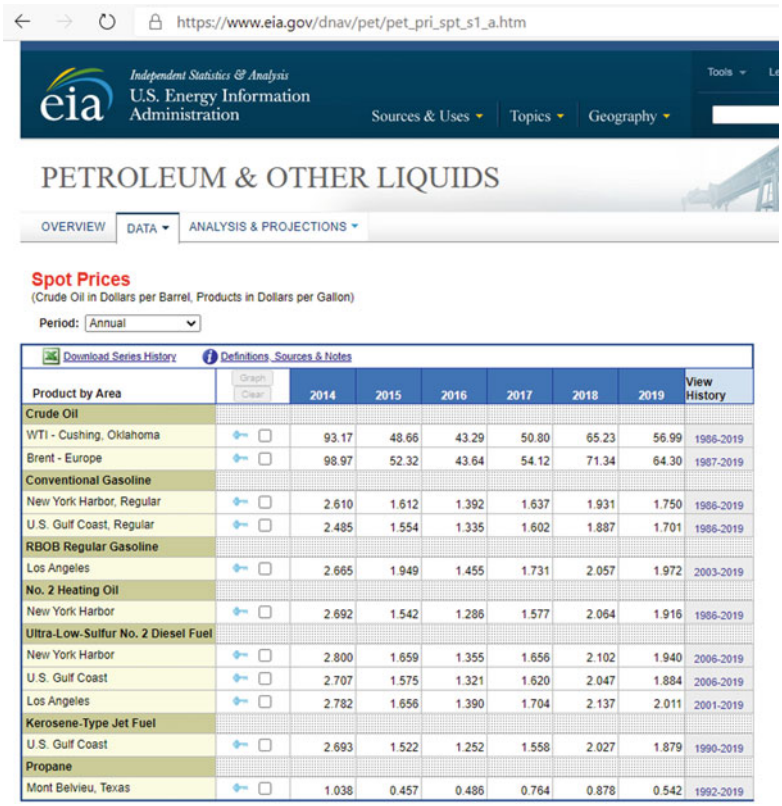
Graphically, the relationship between the two variables is believed to look like the following:

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_13



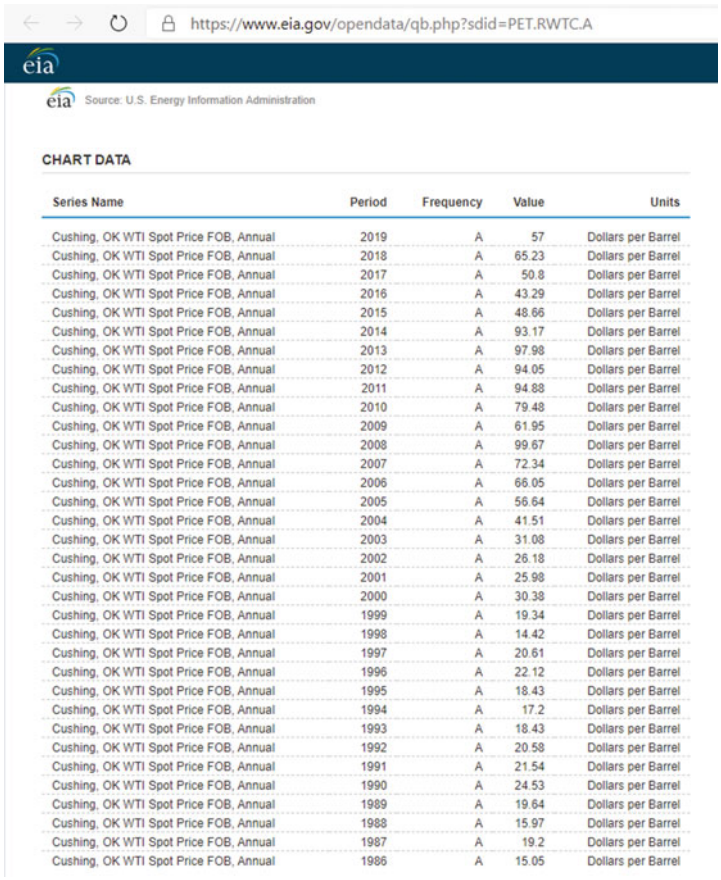
We will investigate the relationship between the price of gasoline and the price of crude oil. We will go to the *U.S. Energy Information Administration* website to get this information. The following URL contains https://www.eia.gov/dnav/pet/pet_pri_spt_s1_m.htm information about the annual price of many types of crude oil and gasoline.



Product by Area	2014	2015	2016	2017	2018	2019	View History
Crude Oil							
WTI - Cushing, Oklahoma	93.17	48.66	43.29	50.80	65.23	56.99	1986-2019
Brent - Europe	98.97	52.32	43.64	54.12	71.34	64.30	1987-2019
Conventional Gasoline							
New York Harbor, Regular	2.610	1.612	1.392	1.637	1.931	1.750	1986-2019
U.S. Gulf Coast, Regular	2.485	1.554	1.335	1.602	1.887	1.701	1986-2019
RBOB Regular Gasoline							
Los Angeles	2.665	1.949	1.455	1.731	2.057	1.972	2003-2019
No. 2 Heating Oil							
New York Harbor	2.692	1.542	1.286	1.577	2.064	1.916	1986-2019
Ultra-Low-Sulfur No. 2 Diesel Fuel							
New York Harbor	2.800	1.659	1.355	1.656	2.102	1.940	2006-2019
U.S. Gulf Coast	2.707	1.575	1.321	1.620	2.047	1.884	2006-2019
Los Angeles	2.782	1.656	1.390	1.704	2.137	2.011	2001-2019
Kerosene-Type Jet Fuel							
U.S. Gulf Coast	2.693	1.522	1.252	1.558	2.027	1.879	1990-2019
Propane							
Mont Belvieu, Texas	1.038	0.457	0.486	0.764	0.878	0.542	1992-2019

We will investigate the relationship between the price of *New York Harbor conventional gasoline* and the price of *WTI crude oil*.

The following is the URL for the annual price of WIT crude oil: https://www.eia.gov/opendata/qb.php?sdid=PET.RWTC.A.



https://www.eia.gov/opendata/qb.php?sdid=PET.RWTC.A

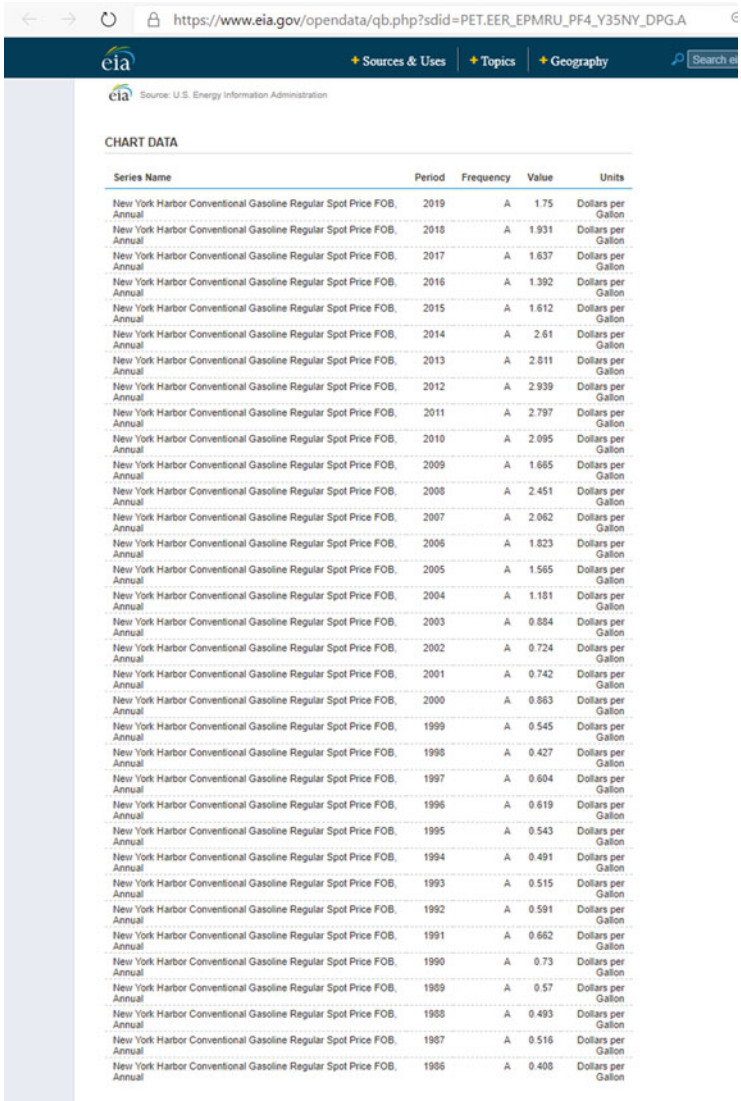
eia

eia Source: U.S. Energy Information Administration

CHART DATA

Series Name	Period	Frequency	Value	Units
Cushing, OK WTI Spot Price FOB, Annual	2019	A	57	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2018	A	65.23	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2017	A	50.8	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2016	A	43.29	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2015	A	48.66	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2014	A	93.17	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2013	A	97.98	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2012	A	94.05	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2011	A	94.88	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2010	A	79.48	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2009	A	61.95	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2008	A	99.67	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2007	A	72.34	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2006	A	66.05	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2005	A	56.64	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2004	A	41.51	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2003	A	31.08	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2002	A	26.18	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2001	A	25.98	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	2000	A	30.38	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1999	A	19.34	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1998	A	14.42	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1997	A	20.61	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1996	A	22.12	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1995	A	18.43	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1994	A	17.2	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1993	A	18.43	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1992	A	20.58	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1991	A	21.54	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1990	A	24.53	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1989	A	19.64	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1988	A	15.97	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1987	A	19.2	Dollars per Barrel
Cushing, OK WTI Spot Price FOB, Annual	1986	A	15.05	Dollars per Barrel

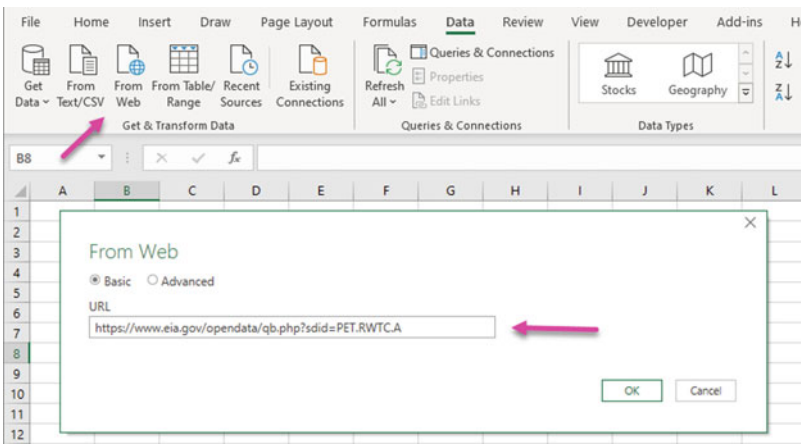
The following is the URL for the annual price of New York Harbor conventional gasoline: https://www.eia.gov/opendata/qb.php?sdid=PET.EER_EPMRU_PF4_Y35NY_DPG.A.



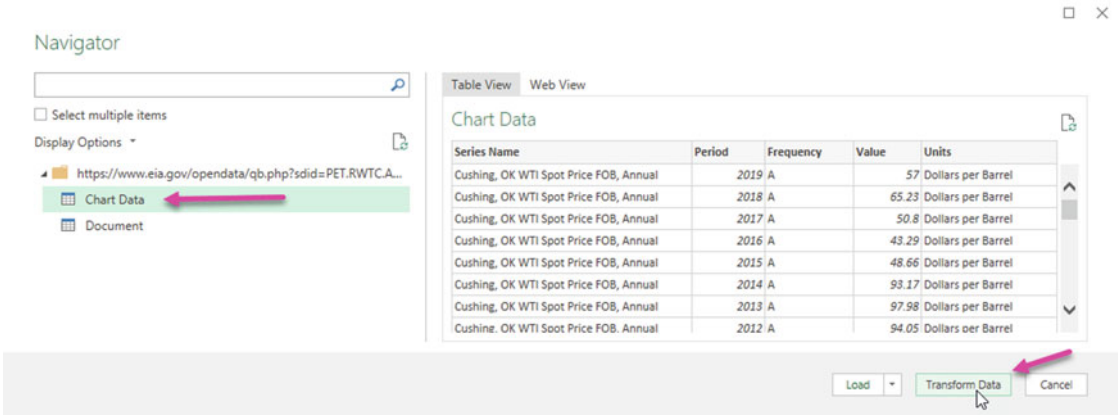
Series Name	Period	Frequency	Value	Units
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2019	A	1.75	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2018	A	1.931	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2017	A	1.637	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2016	A	1.392	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2015	A	1.612	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2014	A	2.61	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2013	A	2.811	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2012	A	2.939	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2011	A	2.797	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2010	A	2.095	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2009	A	1.665	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2008	A	2.451	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2007	A	2.062	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2006	A	1.823	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2005	A	1.565	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2004	A	1.181	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2003	A	0.884	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2002	A	0.724	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2001	A	0.742	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	2000	A	0.863	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1999	A	0.545	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1998	A	0.427	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1997	A	0.604	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1996	A	0.619	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1995	A	0.543	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1994	A	0.491	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1993	A	0.515	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1992	A	0.591	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1991	A	0.662	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1990	A	0.73	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1989	A	0.57	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1988	A	0.493	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1987	A	0.516	Dollars per Gallon
New York Harbor Conventional Gasoline Regular Spot Price FOB, Annual	1986	A	0.408	Dollars per Gallon

13.3 Retrieving Data Using Power Query

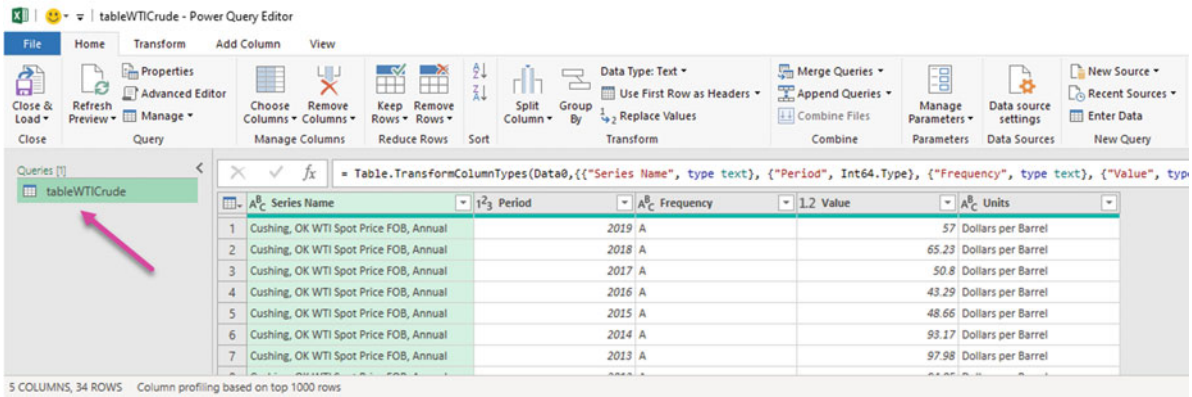
We will use *Power Query* to retrieve the crude oil and gasoline prices. We first enter the crude oil URL into *Power Query*.



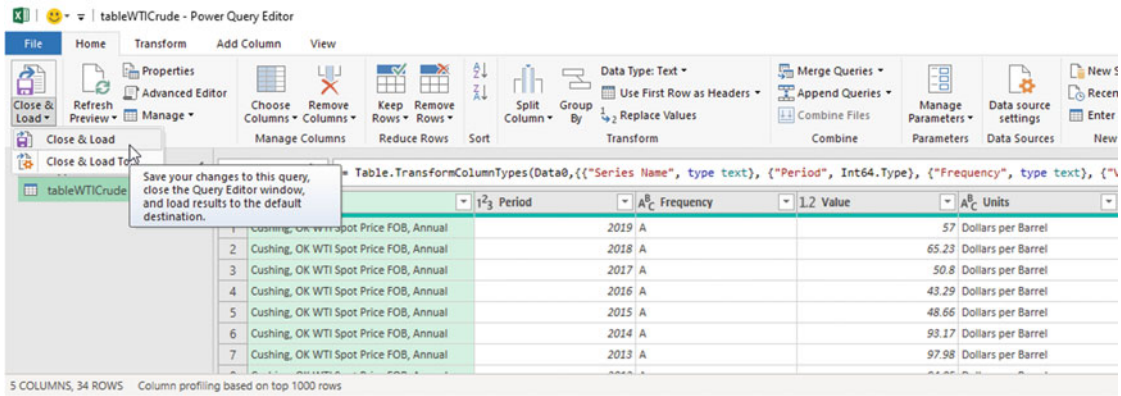
Next, we choose *Chart Data* and then click on *Transform Data.*



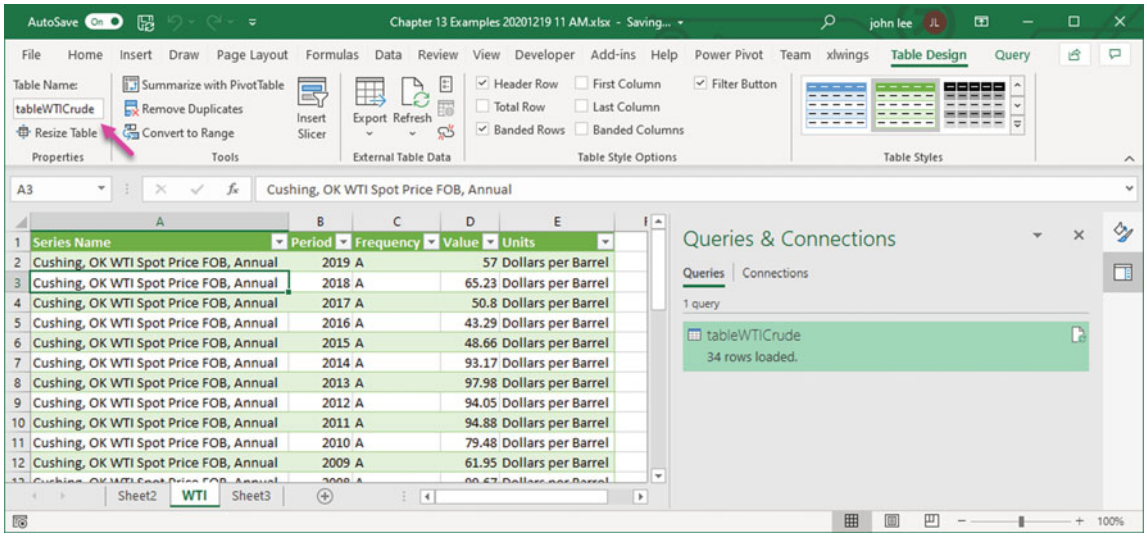
Next, rename the query to "tableWTICrude."



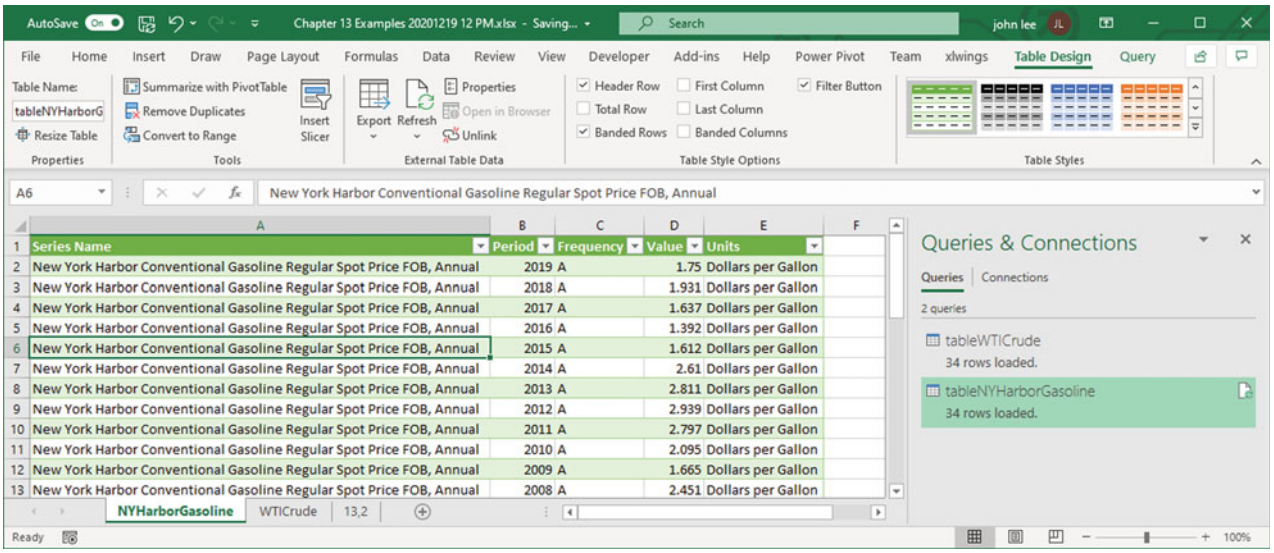
Next, click on *Close & Load.*



Power Query puts the WTI data into an Excel Table. The name of the Excel Table is the same as the *Power Query*. Let's rename the worksheet as "WTICrude."

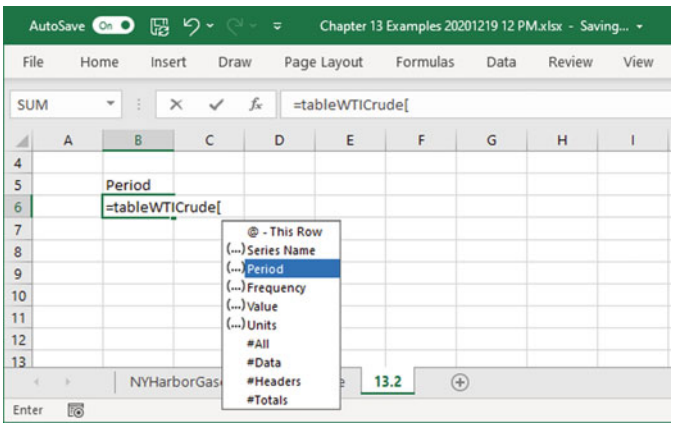


Next, do the same for the New York Harbor gasoline.

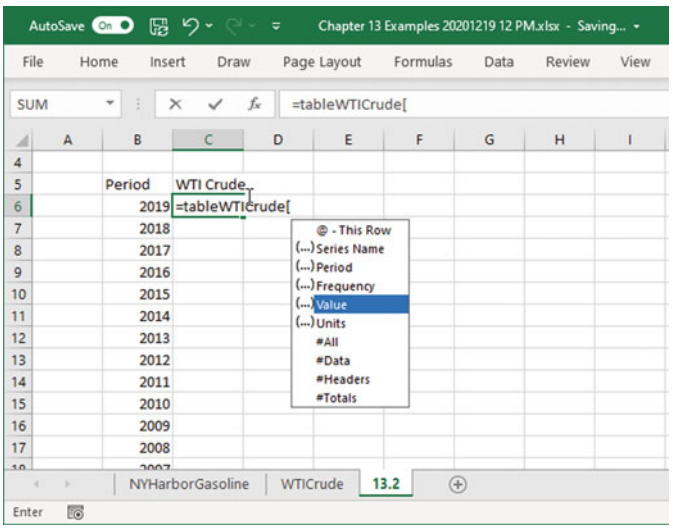


13.4 Combining Power Query Data Sets

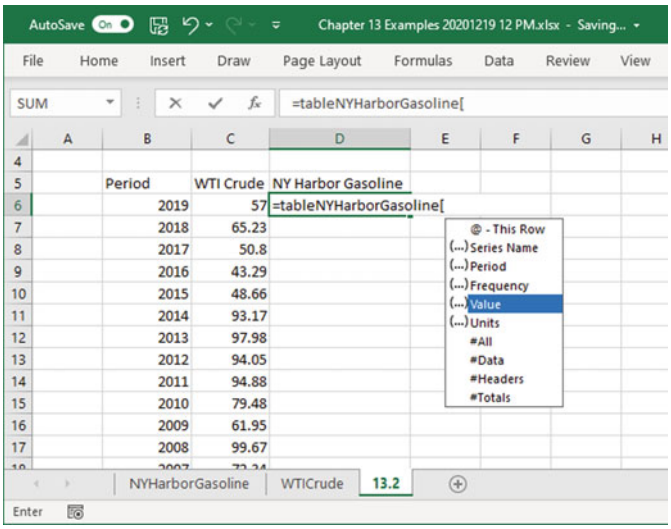
Let's now combine the two Excel Tables. We will first get the period information from the WTI crude Excel Table.



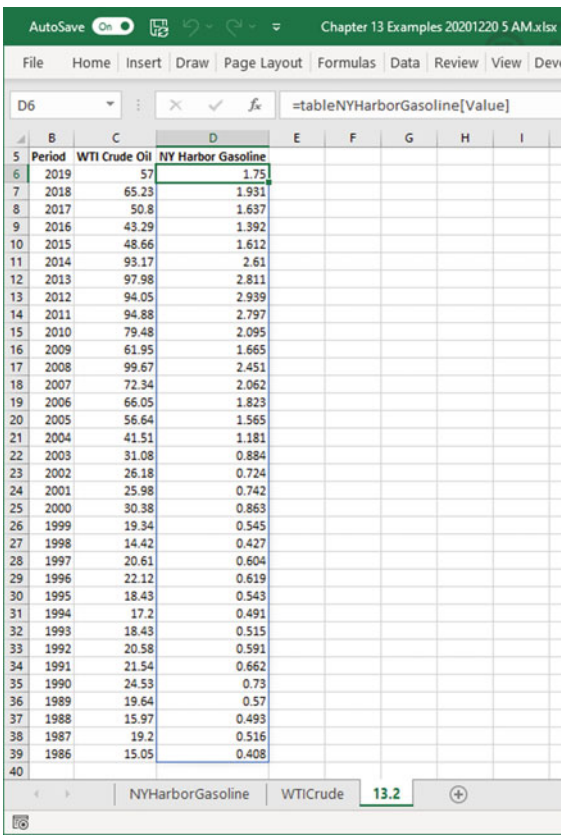
Next, we will get the WTI crude prices.



Next, we will get the NY Harbor gasoline prices.

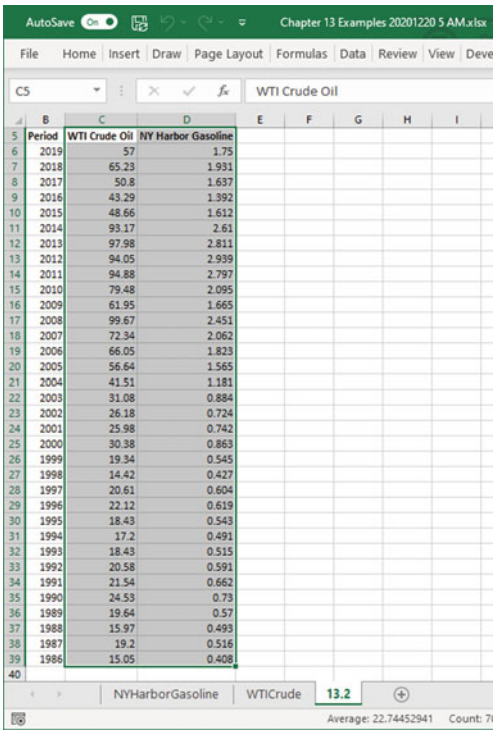


We used *dynamic arrays* to retrieve the data from the crude and gasoline Excel Tables. *Dynamic Arrays* is a feature only available in Excel 365.



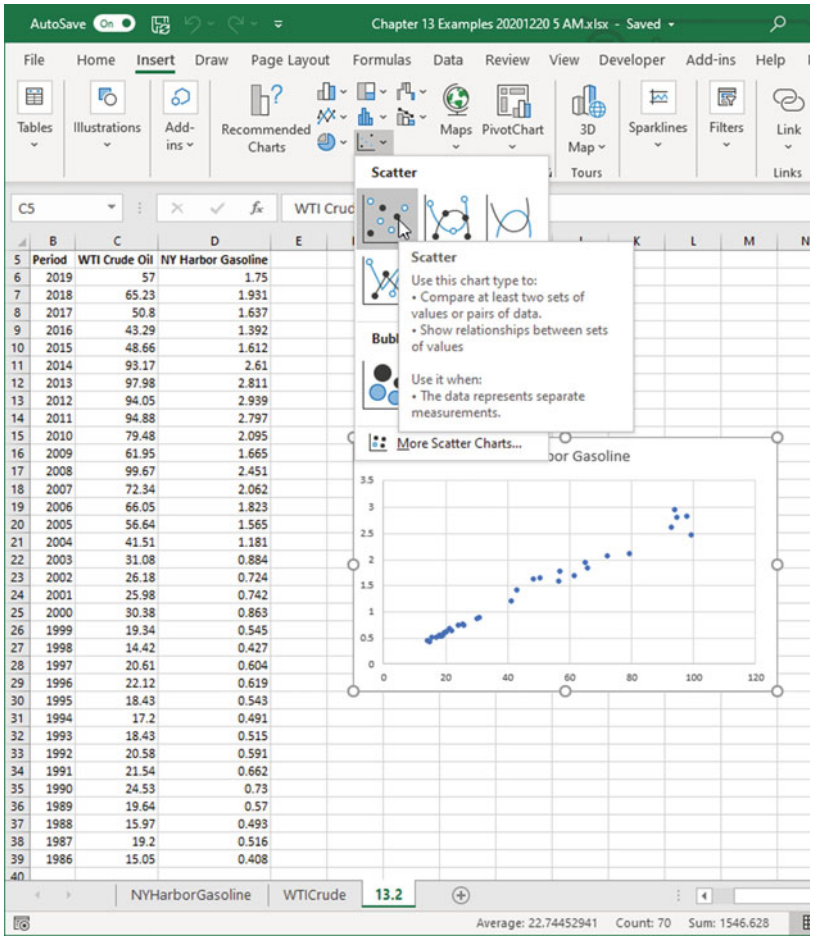
13.5 Scatter Chart

We will first create a *scatter chart* of the WTI Crude oil and New York Harbor gasoline to understand the relationship between WTI Crude oil and New York.

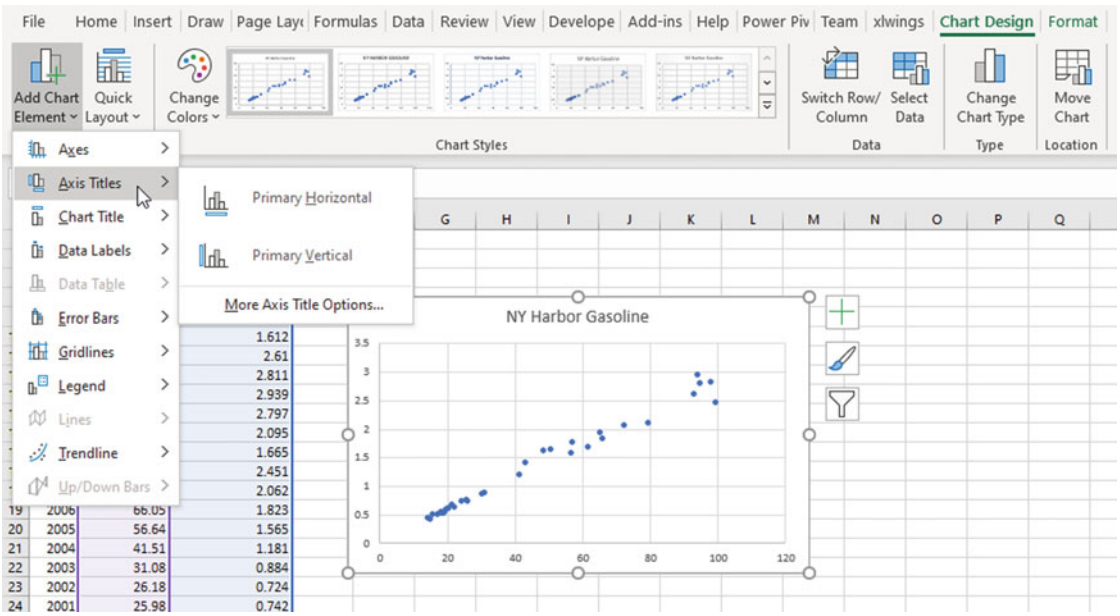


Period	WTI Crude Oil	NY Harbor Gasoline
2019	57	1.75
2018	65.23	1.931
2017	50.8	1.637
2016	43.29	1.392
2015	48.66	1.612
2014	93.17	2.61
2013	97.98	2.811
2012	94.05	2.939
2011	94.88	2.797
2010	79.48	2.095
2009	61.95	1.665
2008	99.67	2.451
2007	72.34	2.062
2006	66.05	1.823
2005	56.64	1.565
2004	41.51	1.181
2003	31.08	0.884
2002	26.18	0.724
2001	25.98	0.742
2000	30.38	0.863
1999	19.34	0.545
1998	14.42	0.427
1997	20.61	0.604
1996	22.12	0.619
1995	18.43	0.543
1994	17.2	0.491
1993	18.43	0.515
1992	20.58	0.591
1991	21.54	0.662
1990	24.53	0.73
1989	19.64	0.57
1988	15.97	0.493
1987	19.2	0.516
1986	15.05	0.408

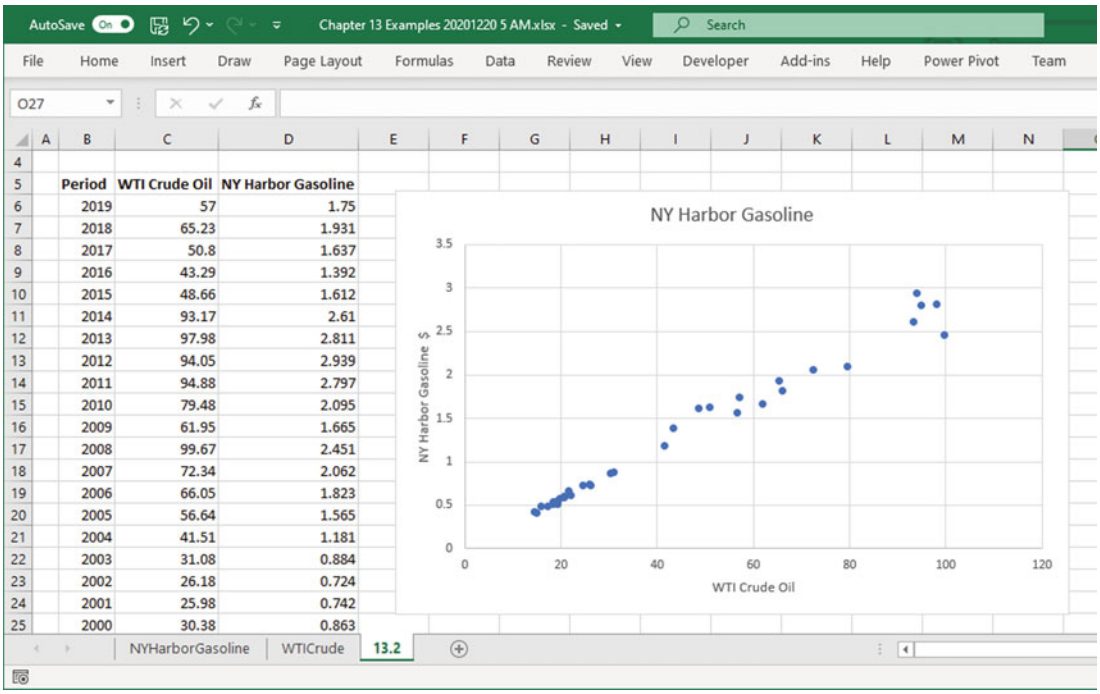
To graph the above data, first, select the WTI Crude Oil and New York Harbor gasoline columns and then choose **Insert → Scatter**.



Excel will make the first column selected as the x-axis and the second column as the y-axis.

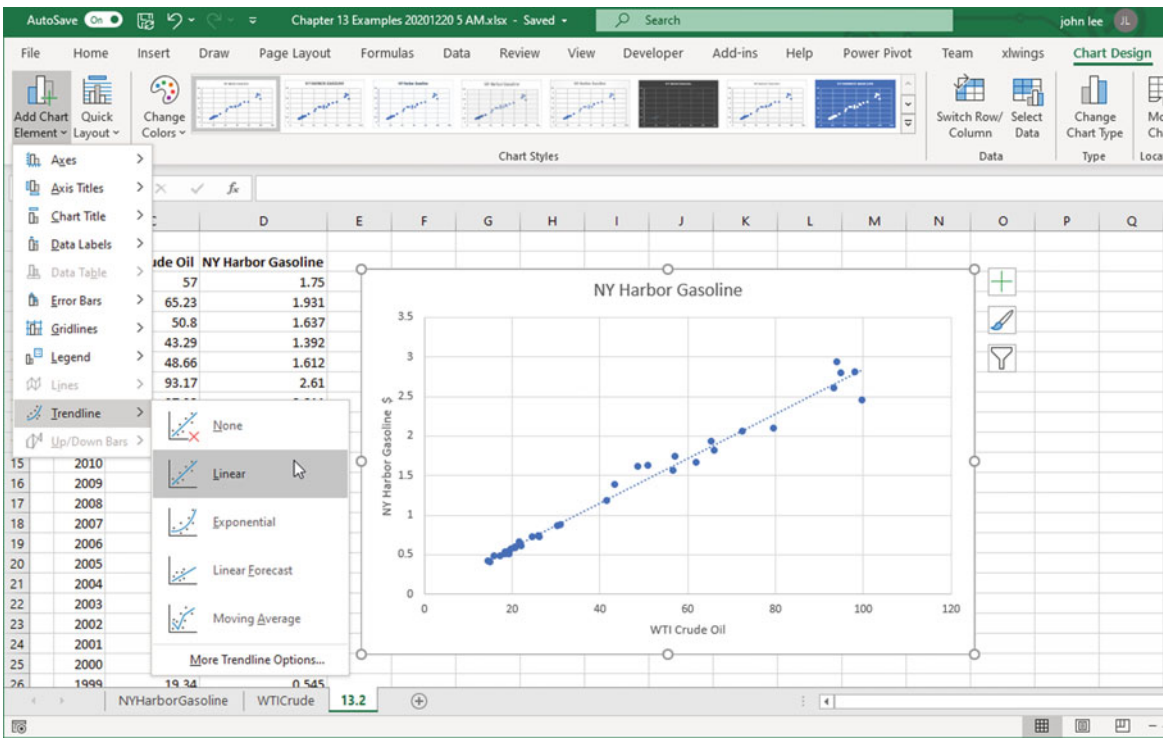


Choose Chart Design → Add Chart → Axis Titles to add the x- and y-axis titles.

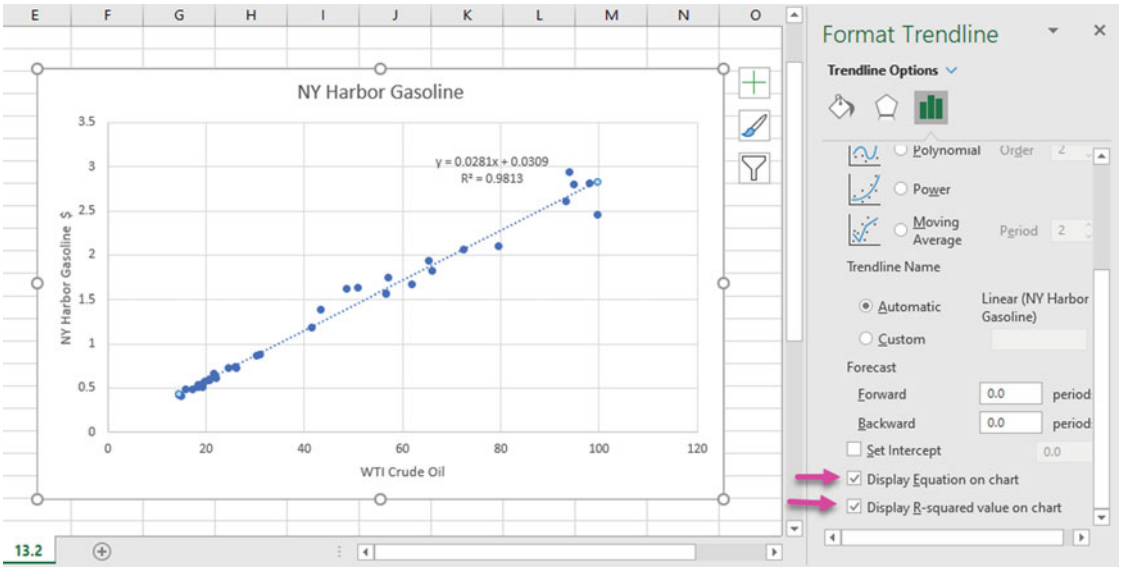


Period	WTI Crude Oil	NY Harbor Gasoline
2019	57	1.75
2018	65.23	1.931
2017	50.8	1.637
2016	43.29	1.392
2015	48.66	1.612
2014	93.17	2.61
2013	97.98	2.811
2012	94.05	2.939
2011	94.88	2.797
2010	79.48	2.095
2009	61.95	1.665
2008	99.67	2.451
2007	72.34	2.062
2006	66.05	1.823
2005	56.64	1.565
2004	41.51	1.181
2003	31.08	0.884
2002	26.18	0.724
2001	25.98	0.742
2000	30.38	0.863

From the above scatter chart, the WTI Crude oil and NY Harbor Gasoline data look linear. One question often asked is, "What is the best linear line that fits the data?" To answer this question, we can use a feature in Excel to draw the best linear line that fits the above data. To do this, choose Chart Design → Add Chart Element → Trendline → Linear.



Next, double-click on the line in the chart. The *Format Trendline* options will appear. Fill out the options, as shown below:



The scatter chart shows three things:

(1) It shows the best linear line that fitted the WTI Crude Oil and NY Harbor Gasoline data.
(2) It gave the equation of the best linear line that fitted WTI Crude Oil and NY Harbor Gasoline data.
(3) It calculated the R^2 for the best linear line that fitted the WTI Crude Oil and NY Harbor Gasoline data.

Excel gives the best linear line that fits the WTI Crude Oil and NY Harbor Gasoline data as

$$y = 0.0281x + 0.0309$$

It is important to note this linear equation produces an expected value or predicted value of y. For example, the way to interpret the above equation is when crude oil increases by one dollar, the expected increase of gasoline is \$0.059 (0.0281 * 1 + 0.0309).

R^2 is a number from 0 to 1. R^2 represents how well the linear line explains the relationship between gasoline and crude oil. Excel calculated the R^2 number as 0.9813. This means WTI crude oil explains 98.13% of the increase in NY Harbor gasoline price.

13.6 Deterministic Relationship and Stochastic Relationship

When we study linear equations in algebra, we expect that for each x-value, there is only one value of y. This kind of relationship is called a *deterministic relationship*. The equation for a line in algebra is

$$y = a + bx$$

When we study linear equations in statistics, this is not the case. In statistics, for each x-value, there is a probability of distribution of y. This kind of relationship is called a *stochastic relationship*. In statistics, the equation for a straight line or *regression line* is

$$y = a + bx + \varepsilon.$$

where ε is called the *residual* or the unexplained of the regression line. It is ε that has a distribution of values for each data point.

13.7 Least Square Method

Statisticians use the *Least Square Method* to find the best line that fits a data set. Using the *Least Square Method,* the following is the formula to find the slope of the best fit line,

$$b = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

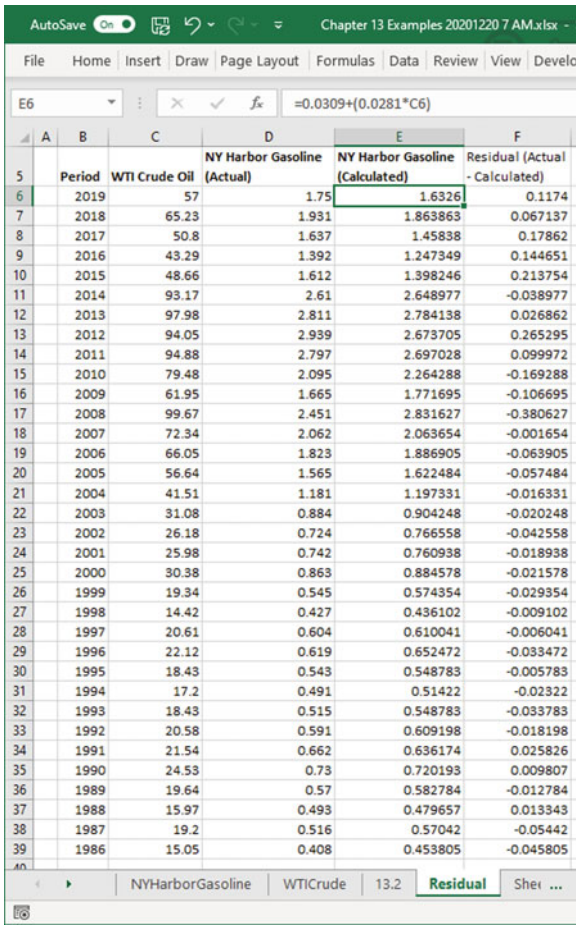
Using the *Least Square Method*, the following is the formula to find the best y-intercept of the best fit line,

$$a = \overline{y} - b\overline{x}$$

The Least Squares regression line would then be

$$\hat{y} = a + bx$$

13.8 Standard Assumptions for Linear Regression



	Period	WTI Crude Oil	NY Harbor Gasoline (Actual)	NY Harbor Gasoline (Calculated)	Residual (Actual - Calculated)
6	2019	57	1.75	1.6326	0.1174
7	2018	65.23	1.931	1.863863	0.067137
8	2017	50.8	1.637	1.45838	0.17862
9	2016	43.29	1.392	1.247349	0.144651
10	2015	48.66	1.612	1.398246	0.213754
11	2014	93.17	2.61	2.648977	-0.038977
12	2013	97.98	2.811	2.784138	0.026862
13	2012	94.05	2.939	2.673705	0.265295
14	2011	94.88	2.797	2.697028	0.099972
15	2010	79.48	2.095	2.264288	-0.169288
16	2009	61.95	1.665	1.771695	-0.106695
17	2008	99.67	2.451	2.831627	-0.380627
18	2007	72.34	2.062	2.063654	-0.001654
19	2006	66.05	1.823	1.886905	-0.063905
20	2005	56.64	1.565	1.622484	-0.057484
21	2004	41.51	1.181	1.197331	-0.016331
22	2003	31.08	0.884	0.904248	-0.020248
23	2002	26.18	0.724	0.766558	-0.042558
24	2001	25.98	0.742	0.760938	-0.018938
25	2000	30.38	0.863	0.884578	-0.021578
26	1999	19.34	0.545	0.574354	-0.029354
27	1998	14.42	0.427	0.436102	-0.009102
28	1997	20.61	0.604	0.610041	-0.006041
29	1996	22.12	0.619	0.652472	-0.033472
30	1995	18.43	0.543	0.548783	-0.005783
31	1994	17.2	0.491	0.51422	-0.02322
32	1993	18.43	0.515	0.548783	-0.033783
33	1992	20.58	0.591	0.609198	-0.018198
34	1991	21.54	0.662	0.636174	0.025826
35	1990	24.53	0.73	0.720193	0.009807
36	1989	19.64	0.57	0.582784	-0.012784
37	1988	15.97	0.493	0.479657	0.013343
38	1987	19.2	0.516	0.57042	-0.05442
39	1986	15.05	0.408	0.453805	-0.045805

The best line (regression line) that fitted the gasoline and crude oil data did not fit the data perfectly; there were differences or residuals. The residuals are calculated above for the regression line of the gasoline and crude oil data. It was also noted that the regression line was to be interpreted as an expected value for the dependent variable.

It is also important to note that there are five assumptions for the regression line. These assumptions must be true for us to properly interpret the regression line. The five assumptions are

1. The independent variables (x) are independent of the residuals.
2. The residuals are normally distributed.
3. The residuals have a mean of zero.
4. The residuals are independent of each other.
5. The residuals have constant variances.

The above five assumptions led us to interpret the resulting regressing line as the expected or average regression line. It is also important to note from the above five assumptions that for any given x-value of a regression line, the corresponding y-value has a normal distribution. In our regression analysis, we are assuming that the data population comes from a normal distribution. We can use the central limit theorem to indicate a normal distribution approximation because of a large sample size.

For example, in our gasoline/crude oil example, the regression equation $y = 0.0281x + 0.0309$ indicated that a crude oil price of \$57 should result in an expected price of \$1.6326. The actual observed gasoline price was \$1.75.

13.9 Standard Error of Estimate

There are two ways to measure the goodness of fit of the resulting regression line. The first measure of goodness of fit in a regression analysis is called the *standard error of estimate,* which is usually denoted as s_e. The mathematical formula for the standard error of estimate is

$$s_e = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-2}}$$

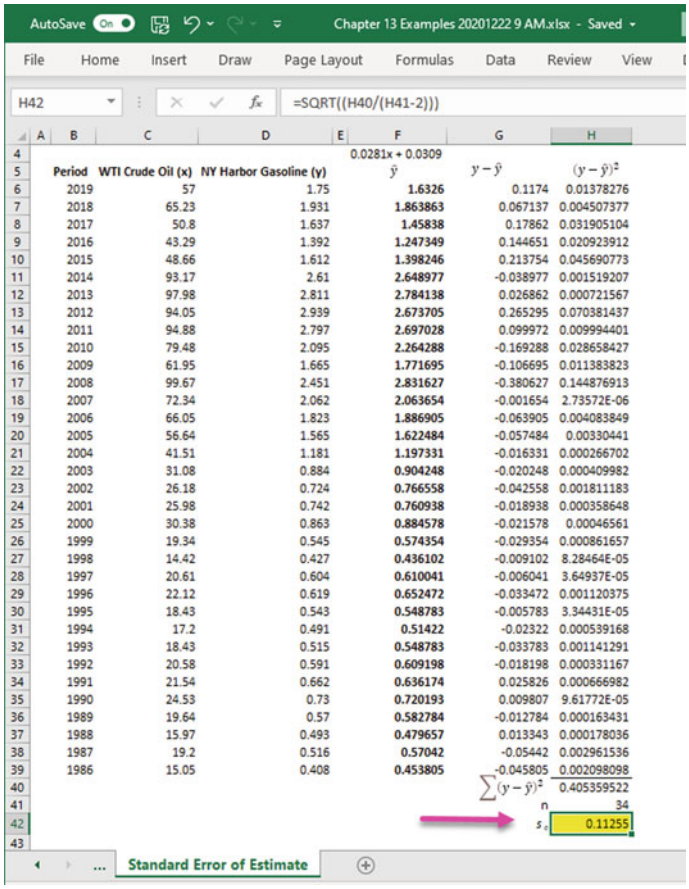
where

y_i is the actual value of the ith element and
$\hat{y}_i$ is the predicted value of the ith element.

The standard error of estimate is much like the standard deviation because they both measure dispersion. The standard deviation measures dispersion around the mean, while the standard error of estimate measures dispersion around the regression line.

If a data set fits perfectly on the regression line, the value of the standard error of estimate would be zero.

The standard error of estimate calculation for our crude oil and gasoline data set is shown below. The standard error of estimate as 0.11255 is calculated below:



We can also interpret the standard error of estimate like the standard deviation because both assume that the data sets are normally distributed. In regression analysis, we are assuming that the data points are normally distributed around the regression line. Therefore, we can expect to find 68% of the points within 1 standard error of the regression line. We can expect that 95% of the points are within 2 standard errors of the regression line. We can expect 98% of the data points are within 3 standard errors of the regression line.

13.10 The Coefficient of Determination

The second measure of goodness of fit in a regression analysis is called the *coefficient of determination,* which is usually denoted as R^2.

In regression analysis, statisticians like to talk about two types of variation, an unexplained variation and an explained variation. The sum of the of the unexplained variation plus the explained variation is equal to the total variation.

$$\text{SST} = \text{SSE} + \text{SSR}$$
$$\text{Total} \rightarrow \text{Unexplained} \rightarrow \text{Explained}$$

Let's now manipulate the above equation to get R^2. From the above equation, we can get

$$\text{SST}/\text{SST} = 1 = \text{SSE}/\text{SST} + \text{SSR}/\text{SST}$$
$$\text{SSR}/\text{SST} = 1 - \text{SSE}/\text{SST}.$$

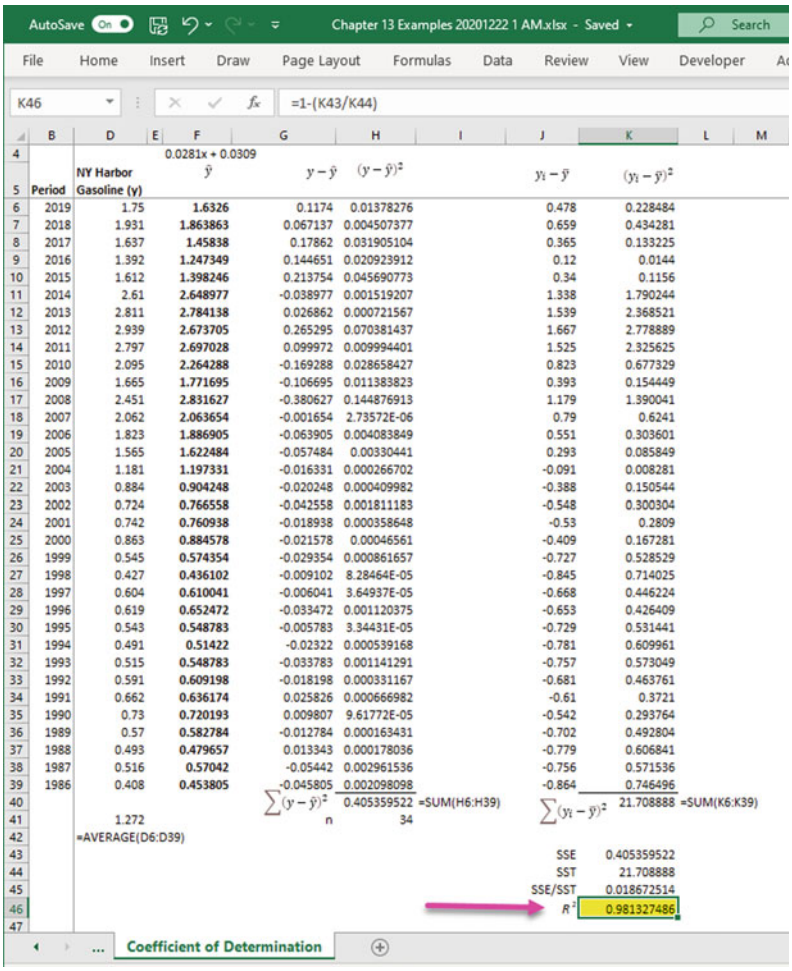
Total variation, sum of squares total (SST), is defined as $\sum_{i=1}^{n} (y_i - \bar{y})^2$.

Unexplained variation, sum of squares error (SSE), is defined as $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$.

Explained variation, sum of squares due to regression (SSR), is defined as $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

The *coefficient of determination* calculation for our crude oil and gasoline example is shown below:

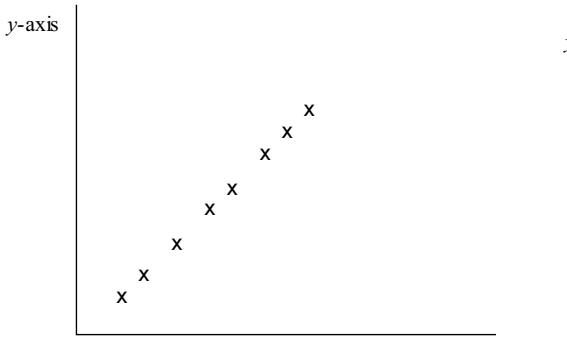


Period	NY Harbor Gasoline (y)	$\hat{y}$	$y - \hat{y}$	$(y - \hat{y})^2$	$y_i - \bar{y}$	$(y_i - \bar{y})^2$
2019	1.75	1.6326	0.1174	0.01378276	0.478	0.228484
2018	1.931	1.863863	0.067137	0.004507377	0.659	0.434281
2017	1.637	1.45838	0.17862	0.031905104	0.365	0.133225
2016	1.392	1.247349	0.144651	0.020923912	0.12	0.0144
2015	1.612	1.398246	0.213754	0.045690773	0.34	0.1156
2014	2.61	2.648977	-0.038977	0.001519207	1.338	1.790244
2013	2.811	2.784138	0.026862	0.000721567	1.539	2.368521
2012	2.939	2.673705	0.265295	0.070381437	1.667	2.778889
2011	2.797	2.697028	0.099972	0.009994401	1.525	2.325625
2010	2.095	2.264288	-0.169288	0.028658427	0.823	0.677329
2009	1.665	1.771695	-0.106695	0.011383823	0.393	0.154449
2008	2.451	2.831627	-0.380627	0.144876913	1.179	1.390041
2007	2.062	2.063654	-0.001654	2.73572E-06	0.79	0.6241
2006	1.823	1.886905	-0.063905	0.004083849	0.551	0.303601
2005	1.565	1.622484	-0.057484	0.00330441	0.293	0.085849
2004	1.181	1.197331	-0.016331	0.000266702	-0.091	0.008281
2003	0.884	0.904248	-0.020248	0.000409982	-0.388	0.150544
2002	0.724	0.766558	-0.042558	0.001811183	-0.548	0.300304
2001	0.742	0.760938	-0.018938	0.000358648	-0.53	0.2809
2000	0.863	0.884578	-0.021578	0.00046561	-0.409	0.167281
1999	0.545	0.574354	-0.029354	0.000861657	-0.727	0.528529
1998	0.427	0.436102	-0.009102	8.28464E-05	-0.845	0.714025
1997	0.604	0.610041	-0.006041	3.64937E-05	-0.668	0.446224
1996	0.619	0.652472	-0.033472	0.001120375	-0.653	0.426409
1995	0.543	0.548783	-0.005783	3.34431E-05	-0.729	0.531441
1994	0.491	0.51422	-0.02322	0.000539168	-0.781	0.609961
1993	0.515	0.548783	-0.033783	0.001141291	-0.757	0.573049
1992	0.591	0.609198	-0.018198	0.000331167	-0.681	0.463761
1991	0.662	0.636174	0.025826	0.000666982	-0.61	0.3721
1990	0.73	0.720193	0.009807	9.61772E-05	-0.542	0.293764
1989	0.57	0.582784	-0.012784	0.000163431	-0.702	0.492804
1988	0.493	0.479657	0.013343	0.000178036	-0.779	0.606841
1987	0.516	0.57042	-0.05442	0.002961536	-0.756	0.571536
1986	0.408	0.453805	-0.045805	0.002098098	-0.864	0.746496
			$\sum(y - \hat{y})^2$	0.405359522	$\sum(y_i - \bar{y})^2$	21.708888
	1.272		n	34		

The calculated R^2 number is 0.9813. This number tells us that variation in crude oil prices explains 98.13% of the variation in gasoline price.

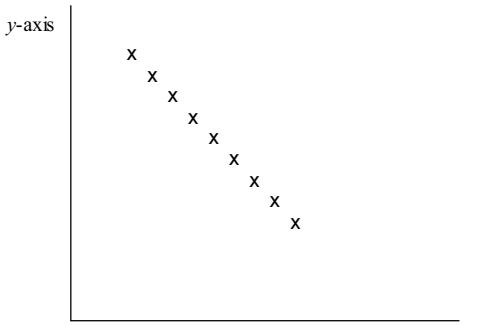
13.11 Correlation Coefficient

Correlation analysis is another way to study the relationship between two variables. In particular, it *measures the strength* of association between two variables. Correlation is expressed in values between −1 and +1. When $r = +1$, then x and y exist in a perfectly positive linear relationship; $r = -1$ means that the two variables x and y exist in a perfect negative linear relationship; and $r = 0$ means that x and y are not linearly related. These concepts are graphically illustrated below.



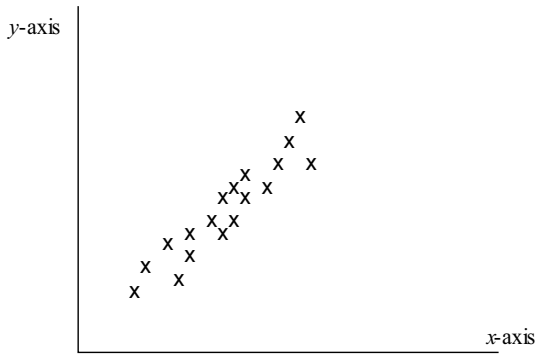
Perfect Correlation of +1

$r = 1$



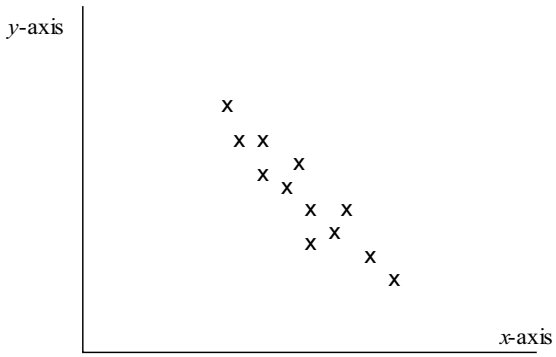
Perfect Correlation of −1

$r = -1$



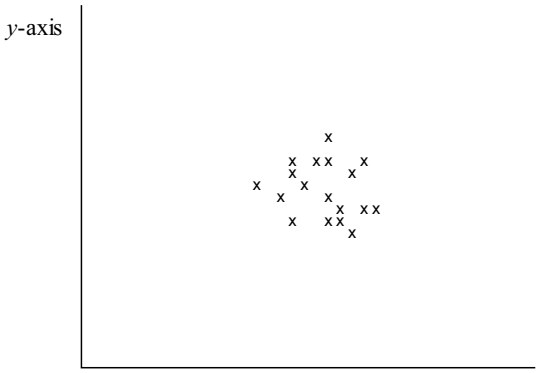
Line Deteriorates, correlation less than +1

$0 < r < 1$



Line Deteriorates, correlation greater than −1

$-1 < r < 0$



No linear relationship

$r = 0$

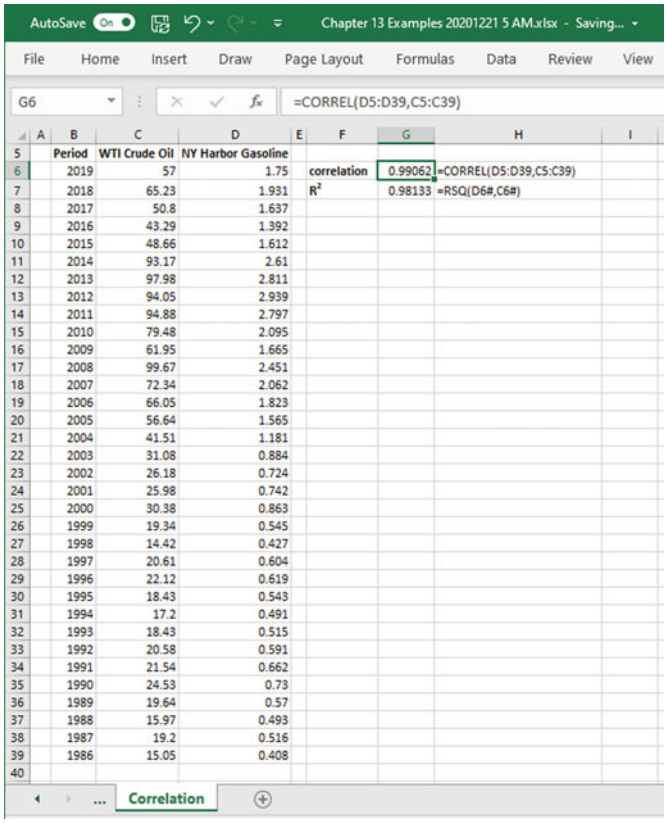
The formula for the correlation coefficient is

$$r = \frac{\text{sample covariance}}{(\text{sample standard deviation of } x)(\text{sample standard deviation of } y)}$$

$$= \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2\right]^{1/2}\left[\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2\right]^{1/2}} = \frac{s_{xy}}{s_x s_y}$$

From the above formula, we can see that the calculation of the correlation coefficient is labor-intensive.

In Excel, there is the *CORREL* worksheet function to calculate the correlation coefficient. The use of this function is shown below. The Excel function *RSQ* to calculate the coefficient of determination is shown below:



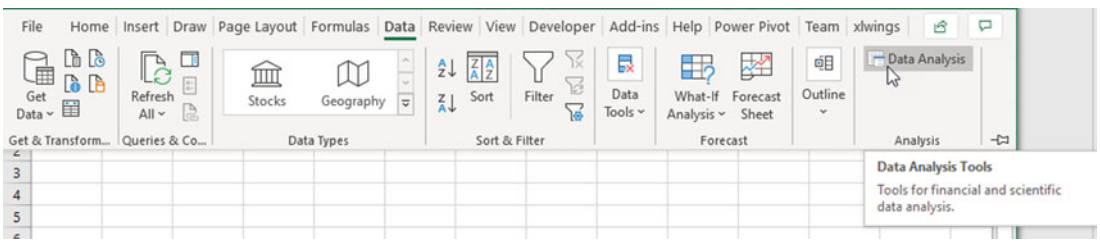
Period	WTI Crude Oil	NY Harbor Gasoline
2019	57	1.75
2018	65.23	1.931
2017	50.8	1.637
2016	43.29	1.392
2015	48.66	1.612
2014	93.17	2.61
2013	97.98	2.811
2012	94.05	2.939
2011	94.88	2.797
2010	79.48	2.095
2009	61.95	1.665
2008	99.67	2.451
2007	72.34	2.062
2006	66.05	1.823
2005	56.64	1.565
2004	41.51	1.181
2003	31.08	0.884
2002	26.18	0.724
2001	25.98	0.742
2000	30.38	0.863
1999	19.34	0.545
1998	14.42	0.427
1997	20.61	0.604
1996	22.12	0.619
1995	18.43	0.543
1994	17.2	0.491
1993	18.43	0.515
1992	20.58	0.591
1991	21.54	0.662
1990	24.53	0.73
1989	19.64	0.57
1988	15.97	0.493
1987	19.2	0.516
1986	15.05	0.408

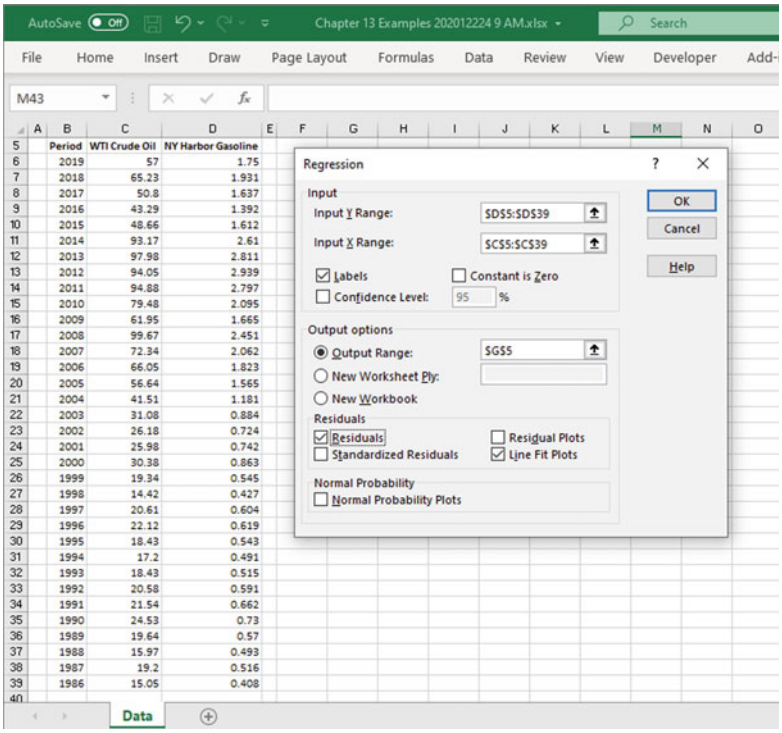
There is a relationship between the correlation coefficient and coefficient of determination. You would square the correlation coefficient to get the coefficient of determination.

One thing to keep in mind is that a strong correlation of 0.7 between two variables is equal to a coefficient of determination of 0.49.

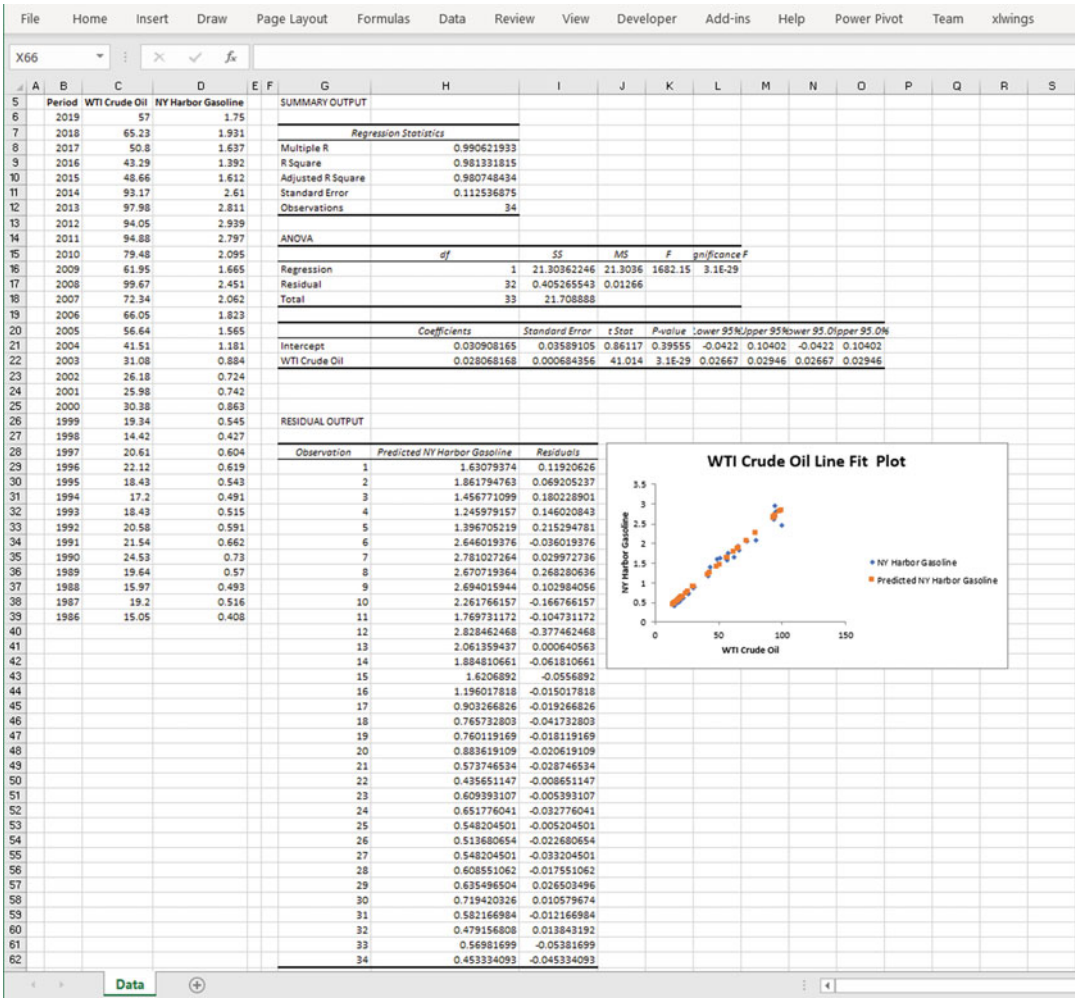
13.12 Regression Analysis in Excel

In Excel, we can do a regression analysis to produce a report with all the statistical information that we produced above. We will use Excel's *Data Analysis* feature to produce the regression report. To use the *Data Analysis* feature, choose **Data** → **Data Analysis**, and then choose the regression option.





After filling out the *Regression* dialog box, as shown above, press the *OK* button. After pressing the *OK* button, you will get the following regression analysis and chart in Excel:



Period	WTI Crude Oil	NY Harbor Gasoline
2019	57	1.75
2018	65.23	1.931
2017	50.8	1.637
2016	43.29	1.392
2015	48.66	1.612
2014	93.17	2.61
2013	97.98	2.811
2012	94.05	2.939
2011	94.88	2.797
2010	79.48	2.095
2009	61.95	1.665
2008	99.67	2.451
2007	72.34	2.062
2006	66.05	1.823
2005	56.64	1.565
2004	41.51	1.181
2003	31.08	0.884
2002	26.18	0.724
2001	25.98	0.742
2000	30.38	0.863
1999	19.34	0.545
1998	14.42	0.427
1997	20.61	0.604
1996	22.12	0.619
1995	18.43	0.543
1994	17.2	0.491
1993	18.43	0.515
1992	20.58	0.591
1991	21.54	0.662
1990	24.53	0.73
1989	19.64	0.57
1988	15.97	0.493
1987	19.2	0.516
1986	15.05	0.408

SUMMARY OUTPUT

Regression Statistics

Multiple R	0.990621933
R Square	0.981331815
Adjusted R Square	0.980748434
Standard Error	0.112536875
Observations	34

ANOVA

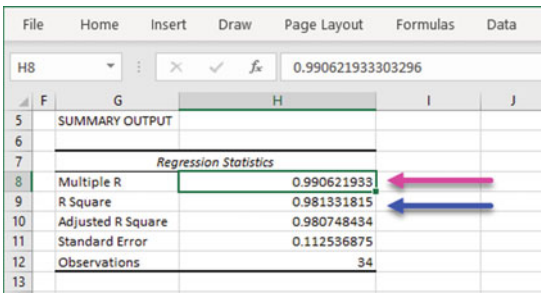
	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	21.30362246	21.3036	1682.15	3.1E-29
Residual	32	0.405265543	0.01266		
Total	33	21.70888			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.030908165	0.03589105	0.86117	0.39555	-0.0422	0.10402	-0.0422	0.10402
WTI Crude Oil	0.028068168	0.000684356	41.014	3.1E-29	0.02667	0.02946	0.02667	0.02946

RESIDUAL OUTPUT

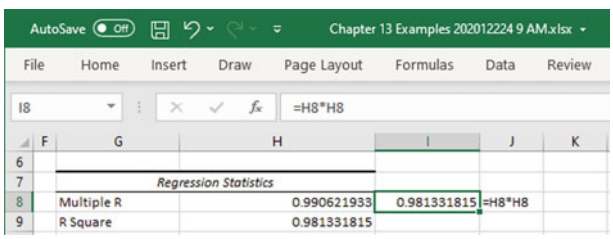
Observation	*Predicted NY Harbor Gasoline*	*Residuals*
1	1.63079374	0.11920626
2	1.861794763	0.069205237
3	1.456771099	0.180228901
4	1.245979157	0.146020843
5	1.396705219	0.215294781
6	2.646019376	-0.036019376
7	2.781027264	0.029972736
8	2.670719364	0.268280636
9	2.694015944	0.102984056
10	2.261766157	-0.166766157
11	1.769731172	-0.104731172
12	2.828462468	-0.377462468
13	2.061359437	0.000640563
14	1.884810661	-0.061810661
15	1.6206892	-0.0556892
16	1.196017818	-0.015017818
17	0.903266826	-0.019266826
18	0.765732803	-0.041732803
19	0.760119169	-0.018119169
20	0.883619109	-0.020619109
21	0.573746534	-0.028746534
22	0.435651147	-0.008651147
23	0.609393107	-0.005393107
24	0.651776041	-0.032776041
25	0.548204501	-0.005204501
26	0.513680654	-0.022680654
27	0.548204501	-0.033204501
28	0.608551062	-0.017551062
29	0.635496504	0.026503496
30	0.719420326	0.010579674
31	0.582166984	-0.012166984
32	0.479156808	0.013843192
33	0.56981699	-0.05381699
34	0.453334093	-0.045334093

13.12.1 Correlation and Coefficient of Determination



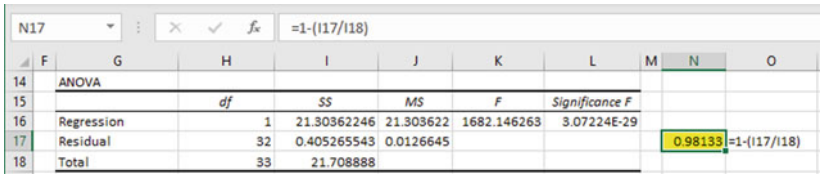
SUMMARY OUTPUT		
	Regression Statistics	
Multiple R	0.990621933	
R Square	0.981331815	
Adjusted R Square	0.980748434	
Standard Error	0.112536875	
Observations	34	

In the regression report, cell h8 is the *correlation* and cell h9 is the *coefficient of determination* or R^2. We can confirm that the square of the correlation is equal to the coefficient of determination.



	Regression Statistics		
Multiple R	0.990621933	0.981331815	=H8*H8
R Square	0.981331815		

We can calculate the R^2 number from the following section of the regression report:



	F	G	H	I	J	K	L	M	N	O
14		ANOVA								
15			*df*	*SS*	*MS*	*F*	*Significance F*			
16		Regression	1	21.30362246	21.303622	1682.146263	3.07224E-29			
17		Residual	32	0.405265543	0.0126645				0.98133	=1-(I17/I18)
18		Total	33	21.708888						

Cell i16 is SSR. Cell i17 is SSE. Cell i18 is SST. Therefore,

$$\begin{aligned} R^2 &= 1 - \text{Residual}/\text{Total} \\ &= 1 - (0.405265543/21.708888) \\ &= 1 - 0.018668185 \\ &= 0.981331815 \end{aligned}$$

13.12.2 Regression Line

We can construct the regression line from the following section of the regression report:

	F	G	H	I	J
19					
20			*Coefficients*	*Standard Error*	*t Stat*
21		Intercept	0.030908165	0.03589105	0.8611664
22		WTI Crude Oil	0.028068168	0.000684356	41.013976

The y-intercept of the regression line is reported in cell h21. The slope of the regression line is reported in cell h22. The regression line is

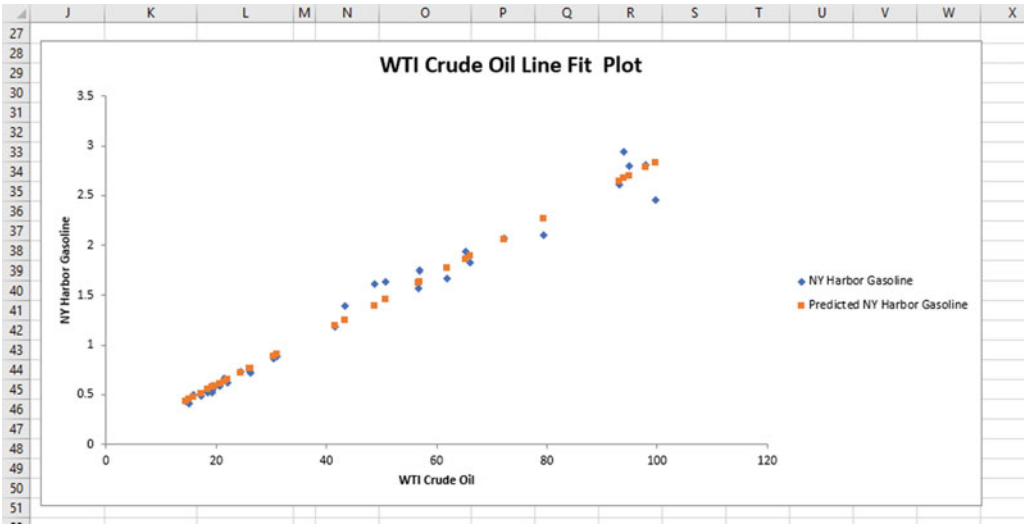
$$y = 0.0281x + 0.0309$$

13.12.3 Residuals of the Regression Line

	F	G	H	I
26		RESIDUAL OUTPUT		
27				
28		*Observation*	*Predicted NY Harbor Gasoline*	*Residuals*
29		1	1.63079374	0.11920626
30		2	1.861794763	0.069205237
31		3	1.456771099	0.180228901
32		4	1.245979157	0.146020843
33		5	1.396705219	0.215294781
34		6	2.646019376	-0.036019376
35		7	2.781027264	0.029972736
36		8	2.670719364	0.268280636
37		9	2.694015944	0.102984056
38		10	2.261766157	-0.166766157
39		11	1.769731172	-0.104731172
40		12	2.828462468	-0.377462468
41		13	2.061359437	0.000640563
42		14	1.884810661	-0.061810661
43		15	1.6206892	-0.0556892
44		16	1.196017818	-0.015017818
45		17	0.903266826	-0.019266826
46		18	0.765732803	-0.041732803
47		19	0.760119169	-0.018119169
48		20	0.883619109	-0.020619109
49		21	0.573746534	-0.028746534
50		22	0.435651147	-0.008651147
51		23	0.609393107	-0.005393107
52		24	0.651776041	-0.032776041
53		25	0.548204501	-0.005204501
54		26	0.513680654	-0.022680654
55		27	0.548204501	-0.033204501
56		28	0.608551062	-0.017551062
57		29	0.635496504	0.026503496
58		30	0.719420326	0.010579674
59		31	0.582166984	-0.012166984
60		32	0.479156808	0.013843192
61		33	0.56981699	-0.05381699
62		34	0.453334093	-0.045334093

Cells h29 to h62 show the residuals of every data point in the data set. Column h shows the predicted gasoline price. Column i shows the difference or residuals from the actual gasoline price.

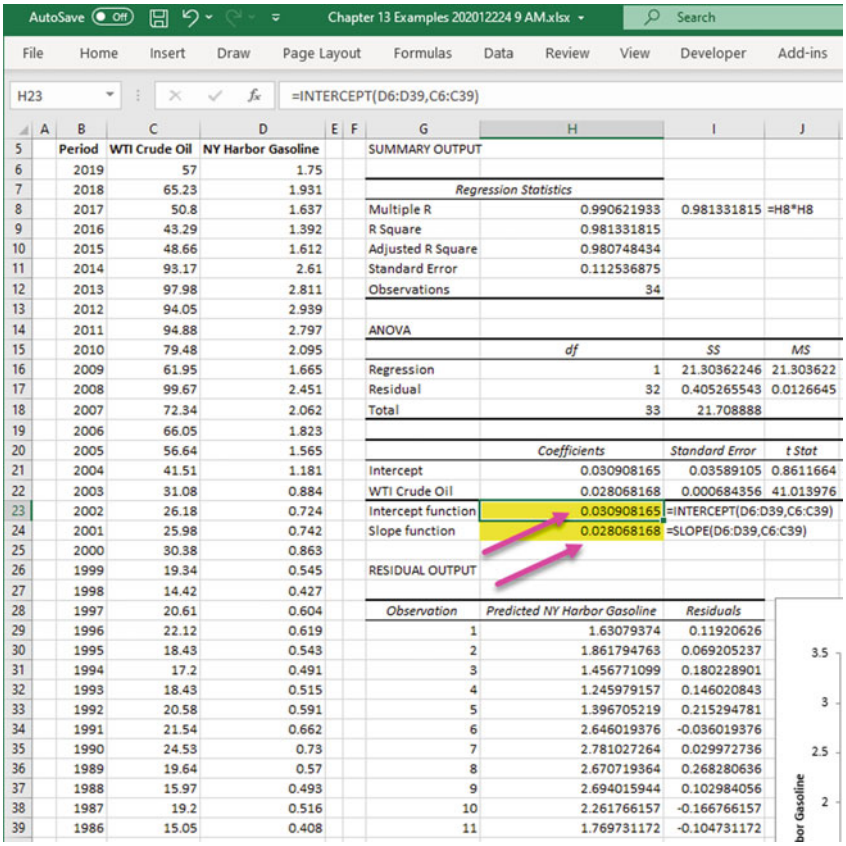
13.12.4 Fit Plot of the Data Set



The regression report also created a scattered chart fit plot of the gasoline and crude oil data. This scatter chart is similar to the scatter chart that we produced at the beginning of this chapter. The orange data points are the predicted value of the gasoline price. The blue data points are the actual gasoline price.

13.13 *INTERCEPT* and *SLOPE* Excel Functions

It is also possible to create the gasoline and crude oil data set regression line by using the *INTERCEPT* and *SLOPE* Excel functions. The *INTERCEPT* Excel function is used to find the y-intercept of the regression line and the *SLOPE* Excel function is used to find the slip of the regression line.



AutoSave Off — Chapter 13 Examples 202012224 9 AM.xlsx — Search

File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins

H23 =INTERCEPT(D6:D39,C6:C39)

	Period	WTI Crude Oil	NY Harbor Gasoline
6	2019	57	1.75
7	2018	65.23	1.931
8	2017	50.8	1.637
9	2016	43.29	1.392
10	2015	48.66	1.612
11	2014	93.17	2.61
12	2013	97.98	2.811
13	2012	94.05	2.939
14	2011	94.88	2.797
15	2010	79.48	2.095
16	2009	61.95	1.665
17	2008	99.67	2.451
18	2007	72.34	2.062
19	2006	66.05	1.823
20	2005	56.64	1.565
21	2004	41.51	1.181
22	2003	31.08	0.884
23	2002	26.18	0.724
24	2001	25.98	0.742
25	2000	30.38	0.863
26	1999	19.34	0.545
27	1998	14.42	0.427
28	1997	20.61	0.604
29	1996	22.12	0.619
30	1995	18.43	0.543
31	1994	17.2	0.491
32	1993	18.43	0.515
33	1992	20.58	0.591
34	1991	21.54	0.662
35	1990	24.53	0.73
36	1989	19.64	0.57
37	1988	15.97	0.493
38	1987	19.2	0.516
39	1986	15.05	0.408

SUMMARY OUTPUT

Regression Statistics

Multiple R	0.990621933	0.981331815	=H8*H8
R Square	0.981331815		
Adjusted R Square	0.980748434		
Standard Error	0.112536875		
Observations	34		

ANOVA

	df	*SS*	*MS*
Regression	1	21.30362246	21.303622
Residual	32	0.405265543	0.0126645
Total	33	21.708888	

	Coefficients	*Standard Error*	*t Stat*
Intercept	0.030908165	0.03589105	0.8611664
WTI Crude Oil	0.028068168	0.000684356	41.013976
Intercept function	0.030908165	=INTERCEPT(D6:D39,C6:C39)	
Slope function	0.028068168	=SLOPE(D6:D39,C6:C39)	

RESIDUAL OUTPUT

Observation	*Predicted NY Harbor Gasoline*	*Residuals*
1	1.63079374	0.11920626
2	1.861794763	0.069205237
3	1.456771099	0.180228901
4	1.245979157	0.146020843
5	1.396705219	0.215294781
6	2.646019376	-0.036019376
7	2.781027264	0.029972736
8	2.670719364	0.268280636
9	2.694015944	0.102984056
10	2.261766157	-0.166766157
11	1.769731172	-0.104731172

3.5 3 2.5 2 bor Gasoline

13.14 Oil and Gasoline Regression Analysis in Python

The oil and gasoline regression analysis in Python is given below:

```
[1] #Date: Apr-04-2021
    # get html
    import requests as _requests
    from bs4 import BeautifulSoup

    #Data Source
    oil_url = "https://www.eia.gov/opendata/qb.php?sdid=PET.RWTC.A"
    gas_url = "https://www.eia.gov/opendata/qb.php?sdid=PET.EER_EPMRU_PF4_Y35NY_DPG.A"

    oil_html = _requests.get(oil_url)
    gas_html = _requests.get(gas_url)
```

```
[2] # create html parser object
    oil_soup = BeautifulSoup(oil_html.content, "html.parser", from_encoding=oil_html.apparent_encoding)
    gas_soup = BeautifulSoup(gas_html.content, "html.parser", from_encoding=gas_html.apparent_encoding)
```

```
[3] # find table for two object
    oil_table = oil_soup.find('tbody')
    gas_table = gas_soup.find('tbody')
```

```
[4] # find all price and store in to a list
    oil_price = []
    gas_price = []
    dates = []
    for tr in oil_table.find_all('tr'):
      node = tr.find_all('td')[3]
      datenode = tr.find_all('td')[1]
      oil_price.append(float(node.text))
      dates.append(datenode.text)
    for tr in gas_table.find_all('tr'):
      node = tr.find_all('td')[3]
      gas_price.append(float(node.text))
```

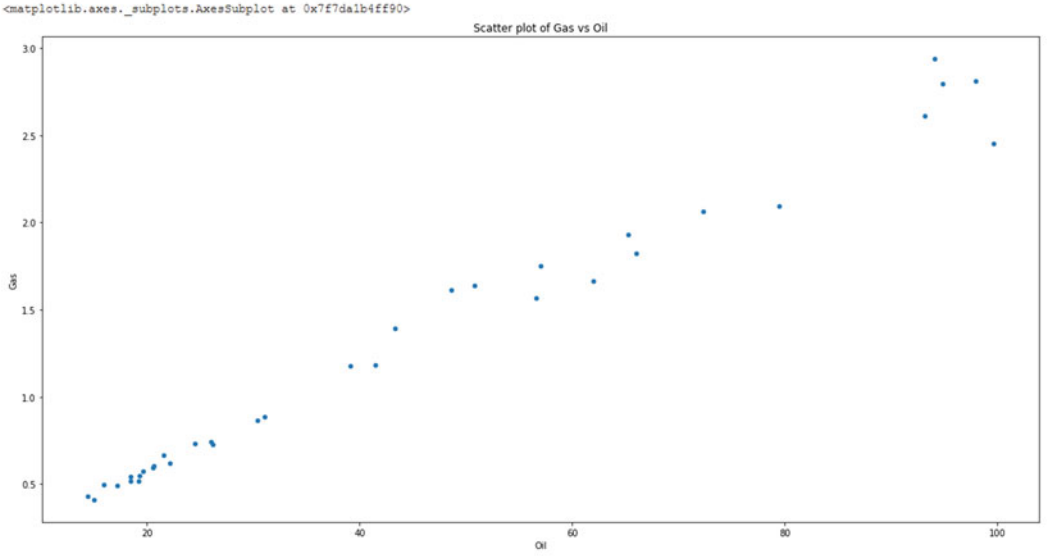
```
[5] # combine it to dataframe
    import pandas as pd
    price = pd.DataFrame(list(zip(dates, gas_price, oil_price)), columns = ['Date','Gas', 'Oil'])
```

```
print(price)
```

```
    Date    Gas    Oil
0   2020  1.178  39.16
1   2019  1.750  57.00
2   2018  1.931  65.23
3   2017  1.637  50.80
4   2016  1.392  43.29
5   2015  1.612  48.66
6   2014  2.610  93.17
7   2013  2.811  97.98
8   2012  2.939  94.05
9   2011  2.797  94.88
10  2010  2.095  79.48
11  2009  1.665  61.95
12  2008  2.451  99.67
13  2007  2.062  72.34
14  2006  1.823  66.05
15  2005  1.565  56.64
16  2004  1.181  41.51
17  2003  0.884  31.08
18  2002  0.724  26.18
19  2001  0.742  25.98
20  2000  0.863  30.38
21  1999  0.545  19.34
22  1998  0.427  14.42
23  1997  0.604  20.61
24  1996  0.619  22.12
25  1995  0.543  18.43
26  1994  0.491  17.20
27  1993  0.515  18.43
28  1992  0.591  20.58
29  1991  0.662  21.54
30  1990  0.730  24.53
31  1989  0.570  19.64
32  1988  0.493  15.97
33  1987  0.516  19.20
34  1986  0.408  15.05
```

```
[7] price.plot(x = 'Oil', y = 'Gas', kind = 'scatter', title = "Scatter plot of Gas vs Oil",figsize=(20,10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7da1b4ff90>
```



```
[8] # changed a library here
    import statsmodels.api as sm
    import statsmodels.formula.api as smf

    fit = smf.ols('Gas ~ Oil', data = price).fit()
    print(fit.summary())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecated
  import pandas.util.testing as tm
                            OLS Regression Results
==============================================================================
Dep. Variable:                    Gas   R-squared:                       0.981
Model:                            OLS   Adj. R-squared:                  0.981
Method:                 Least Squares   F-statistic:                     1726.
Date:                Sun, 04 Apr 2021   Prob (F-statistic):           4.50e-30
Time:                        17:26:55   Log-Likelihood:                 28.266
No. Observations:                  35   AIC:                            -52.53
Df Residuals:                      33   BIC:                            -49.42
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0327      0.035      0.928      0.360      -0.039       0.104
Oil            0.0281      0.001     41.542      0.000       0.027       0.029
==============================================================================
Omnibus:                       10.098   Durbin-Watson:                   1.198
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               18.293
Skew:                          -0.461   Prob(JB):                     0.000107
Kurtosis:                       6.420   Cond. No.                         97.7
==============================================================================
```

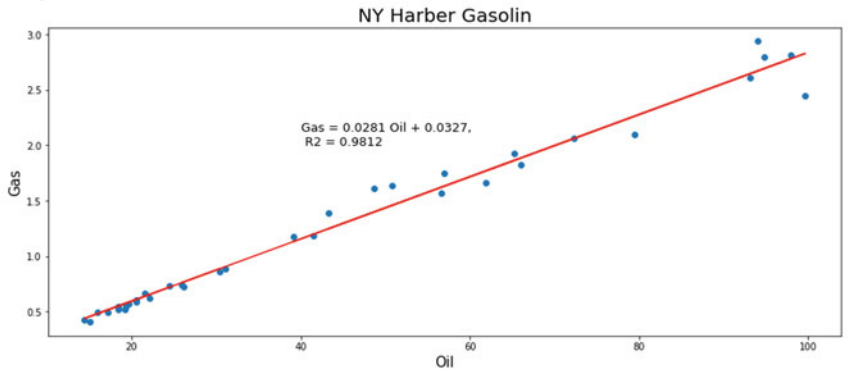
```
[9] # anova report
    table = sm.stats.anova_lm(fit, typ=2)
    print(table)
```

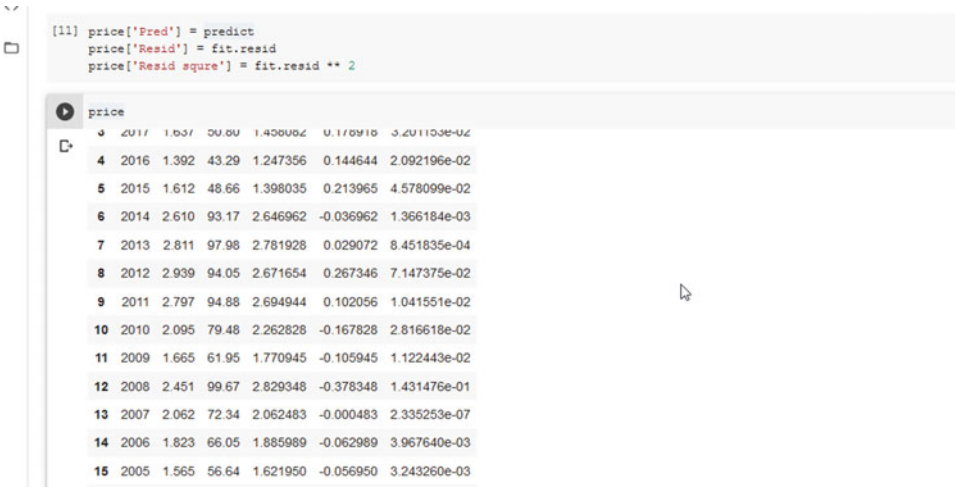
```
              sum_sq    df           F        PR(>F)
Oil       21.309975   1.0  1725.73148  4.497356e-30
Residual   0.407496  33.0         NaN           NaN
```

```
# plot the original graph and prediction
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score

plt.figure(figsize=(15,6))
predict = fit.predict(price['Oil'])
r2 = r2_score(gas_price, predict)
plt.xlabel("Oil", fontsize = 15)
plt.ylabel("Gas", fontsize = 15)
plt.title("NY Harber Gasolin", fontsize = 20)
txt = f"Gas = {fit.params[1]:.4f} Oil + {fit.params[0]:.4f},\n R2 = {r2:.4f}"
plt.text(40, 2, txt, fontsize = 13)
plt.scatter(oil_price, gas_price)
plt.plot(oil_price, predict, color ='red')
```

```
[<matplotlib.lines.Line2D at 0x7f7d81ed9950>]
```





```
[11] price['Pred'] = predict
     price['Resid'] = fit.resid
     price['Resid squre'] = fit.resid ** 2
```

```
price
```

4	2016	1.392	43.29	1.247356	0.144644	2.092196e-02
5	2015	1.612	48.66	1.398035	0.213965	4.578099e-02
6	2014	2.610	93.17	2.646962	-0.036962	1.366184e-03
7	2013	2.811	97.98	2.781928	0.029072	8.451835e-04
8	2012	2.939	94.05	2.671654	0.267346	7.147375e-02
9	2011	2.797	94.88	2.694944	0.102056	1.041551e-02
10	2010	2.095	79.48	2.262828	-0.167828	2.816618e-02
11	2009	1.665	61.95	1.770945	-0.105945	1.122443e-02
12	2008	2.451	99.67	2.829348	-0.378348	1.431476e-01
13	2007	2.062	72.34	2.062483	-0.000483	2.335253e-07
14	2006	1.823	66.05	1.885989	-0.062989	3.967640e-03
15	2005	1.565	56.64	1.621950	-0.056950	3.243260e-03

13.15 Summary

In this chapter, we look at two methods of analyzing the relationship between two variables. The two methods are the simple linear regression method and the correlation method. The simple linear regression analysis can be used to measure how much one variable is explained by another variable.

It is important to note that in algebra, linear equations have a deterministic relationship, while in statistics, linear equations have a *stochastic relationship.*

The correlation analysis can be used to measure the strength of the linear relationship between two variables.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O’Reilly Media, 2017.

14 Simple Linear Regression and Correlation: Analyses and Applications

14.1 Introduction

When we do regression analysis, there are five underlying assumptions:

1. The dependent and independent variables have a linear relationship.
2. The expected value of the error term is zero.
3. The variance of the error term is constant.
4. The error terms are independent.
5. The errors are normally distributed.

In general, we use sample instead of population data to estimate regression coefficients. If the regression model follows the above five assumptions, we can make inferences about the population regression coefficients from the sample regression coefficients. In the last chapter, we used regression and correlation analysis to relate two variables based on sample information. But data from a sample represent only part of the total population. Because of this, we may think of our estimated sample regression line as an estimate of a true but unknown population regression line of the form:

$$y = \alpha + \beta x$$

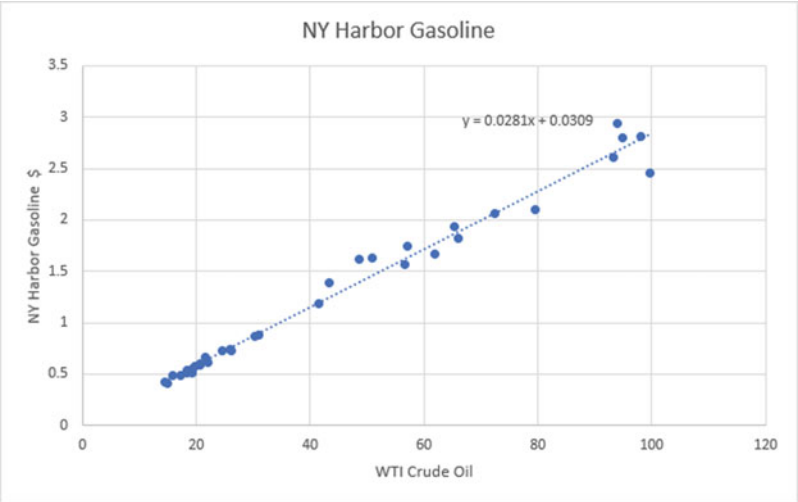
Therefore, what we are doing is using the sample regression line, $\hat{y} = a + bx$, to estimate the true population regression line. We will use the sample regression line to make inferences about the population regression line.

This chapter is broken down into the following sections. Section 14.2 discusses the standard error of estimate. Section 14.3 discusses for μ when σ2 is unknown, Sect. 14.4 discusses two-tailed t-Test for *, and Sect. 14.5 discusses confidence interval of β. In Sect. 14.6, we look at the F test, and in Sect. 14.7, we talk about the relationship between the F-test and the t-test. Market model is discussed in Sect. 14.8. The Yahoo! Finance beta screener is discussed in Sect. 14.9, while historical monthly data from Yahoo! Finance is discussed in Sect. 14.10. The market model of Apple Inc. in Excel is discussed in Sect. 14.11, and the market model of the Clorox Company in Excel is discussed in Sect. 14.12. Market model in Python is talked about in Sect. 14.13. Finally, Sect. 14.14 summarizes the chapter.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_14

14.2 Standard Error of Estimate



There are two ways to measure the goodness of fit of the resulting regression line. The first measure of goodness of fit in a regression analysis is called the *standard error of estimate,* which is usually denoted as s_e. The mathematical formula for the standard error of estimate is,

$$s_e = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n-2}}$$

where

y_i is the actual value of the *i*th element.

$\hat{y}_i$ is the predicted value of the *i*th element.

The standard error of estimate is much like the standard deviation because they both measure dispersion. The standard deviation measures dispersion around the mean, while the standard error of estimate measures dispersion around the regression line.

Excel's regression report reports the *standard error of estimate.*

	F	G	H	I
5		SUMMARY OUTPUT		
6				
7		*Regression Statistics*		
8		Multiple R	0.990621933	
9		R Square	0.981331815	
10		Adjusted R Square	0.980748434	
11		Standard Error	0.112536875	
12		Observations	34	
13				

14.3 Two-Tailed *t*-Test for β

One statistical inference that we are interested in is to test the hypothesis about the population value of β. We are interested in testing to see if x and y are linearly related. If they are related, then β does not equal to 0. The null hypothesis (H_0) and the alternative hypothesis (H_1) would be the following.

$$H_0: \beta = 0$$
$$H_1: \beta \neq 0$$

The following is the standard error of b is,

$$s_b = \frac{s_e}{\sqrt{\sum (x_i - x)^2}}$$

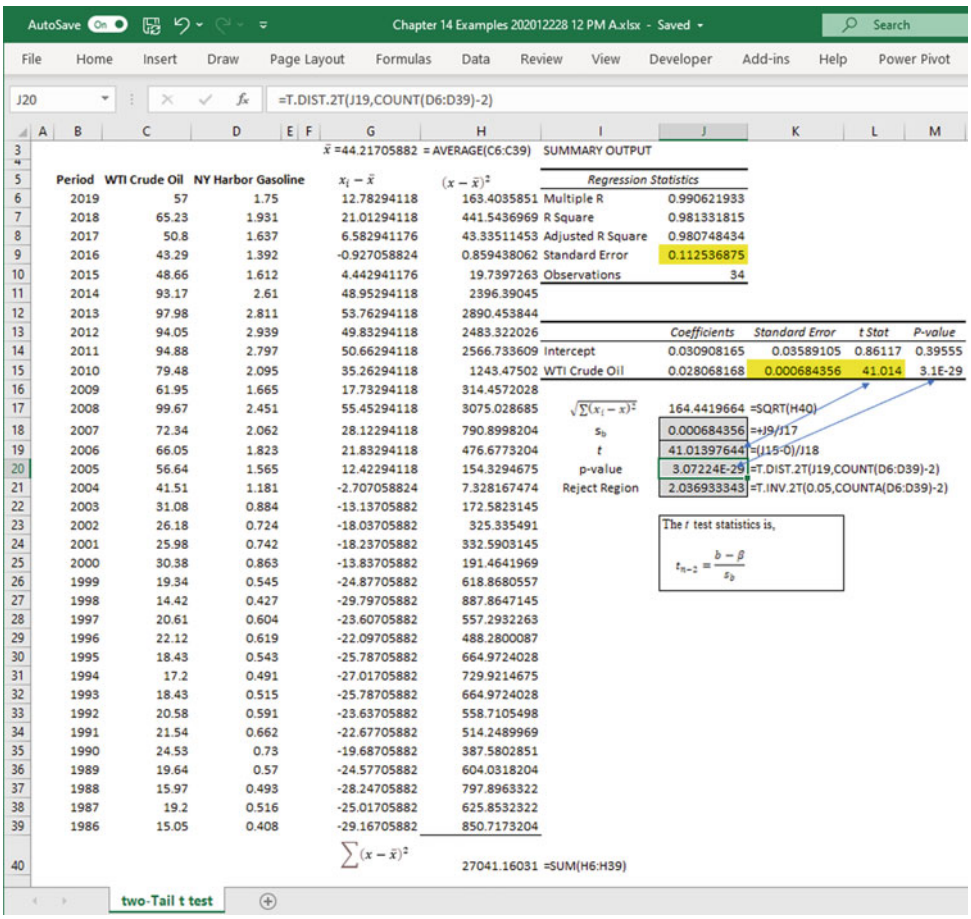
where s_e is the standard error of estimates and s_b is the standard error of b.

We will use the t-distribution to test the above hypothesis. The t-test statistics is,

$$t_{n-2} = \frac{b - \beta}{s_b}$$

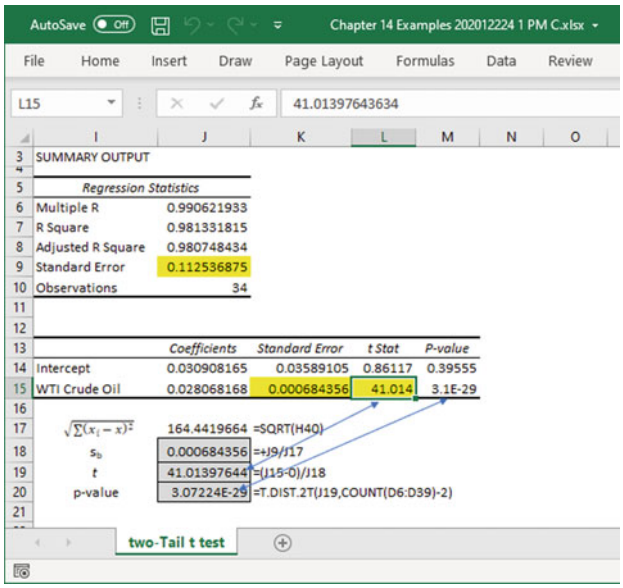
In the t-test statistic, there are $n - 2$ degrees of freedom.

In Chap. 13, we created the following regression line, y = 0.0281x + 0.0309, for the gasoline and crude oil data set. Below shows the calculation to test to see if the slope is different from zero.



Cell J19 manually calculated the t statistics as 41.01. Cell J21 calculated the p-value of 3.07224408348771E−29, which is basically zero. Since the t statistic is greater than the reject region of 2.0369 for an alpha value of 0.05, we reject the null hypothesis that the slope equals zero. We can also reject the null hypothesis because the p-value is less than the two-tailed alpha value of 0.025.

The range I3 to M20 is reproduced from the statistical report shown in Chap. 13.



Note that the manual t statistics calculation in cell J19 agrees with the statistical report value in cell L15. The calculated p-value in cell J20 agrees with the statistical report value in cell M15.

14.4 Two-Tailed t-Test for α

We can also test whether the population intercept α is significantly different from zero or not. We will use the t statistic to test whether α is significantly different from zero or not. The null and alternative hypotheses for α would be:

$$H_0 : \alpha = 0$$
$$H_1 : \alpha \neq 0$$

The following is the standard error of a is,

$$s_a = \sqrt{\frac{s_e^2 \sum_{i=1}^{n} x_i^2}{n \sum_{i=i}^{n} (x_i - \overline{x})^2}}$$

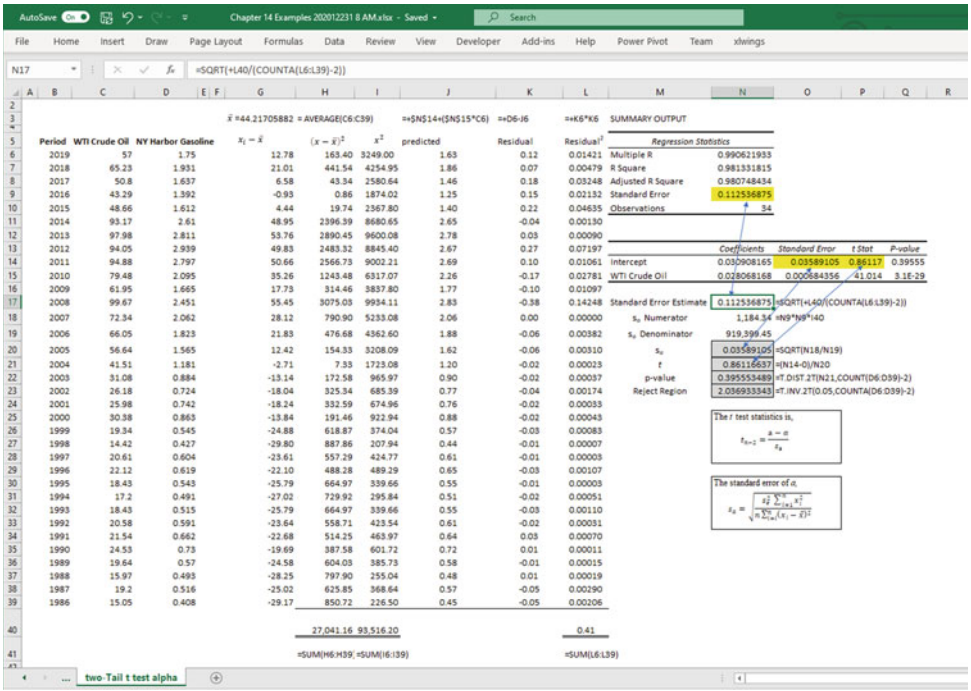
where s_e is the standard error of estimates and s_a is the standard error of a.

We will use the t-distribution to test the above hypothesis. The t-test statistics is,

$$t_{n-2} = \frac{a - \alpha}{s_a}$$

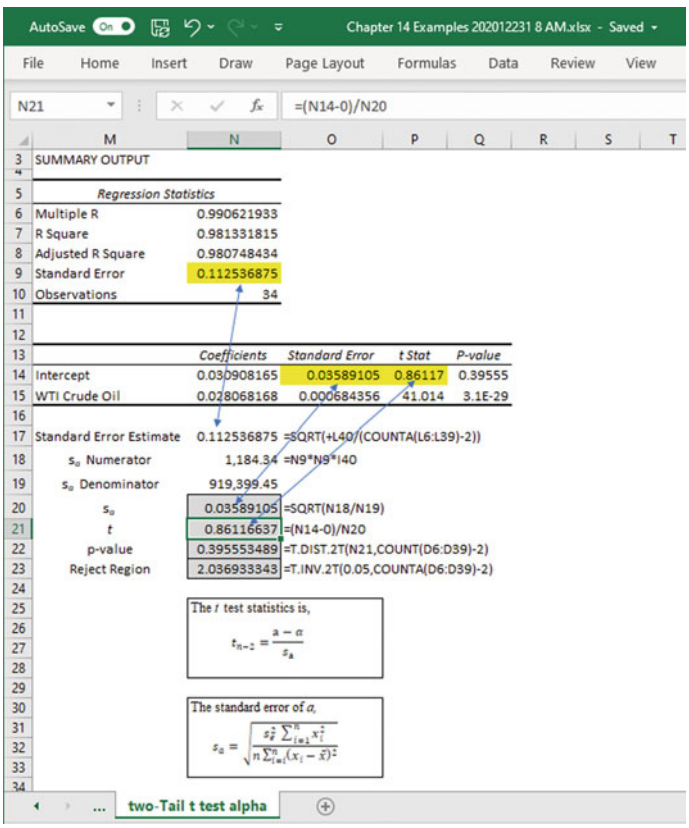
In the t-test statistic, there are $n - 2$ degrees of freedom.

In Chap. 13, we created the following regression line, y = 0.0281x + 0.0309 for the gasoline and crude oil data set. Below shows the calculation to test to see if the y-intercept is different from zero.



Cell N21 manually calculated the *t* statistics as 0.86117. Cell J22 calculated the *p*-value of 0.3955. Since the *t* statistic is less than the reject region of 2.0369 for an alpha value of 0.05, we cannot reject the null hypothesis that the y-intercept equals zero. We also can not reject the null hypothesis because the *p*-value is less than the two-tailed alpha value of 0.025.

The range M3 to Q5 is reproduced from the statistical report shown in Chap. 13.



Note that the manual t statistics calculation in cell N21 agrees with the statistical report value in cell P14. The calculated p-value in cell N22 agrees with the statistical report value in cell O14. The calculated standard error of estimate in cell N17 agrees with the statistical report value in cell N9.

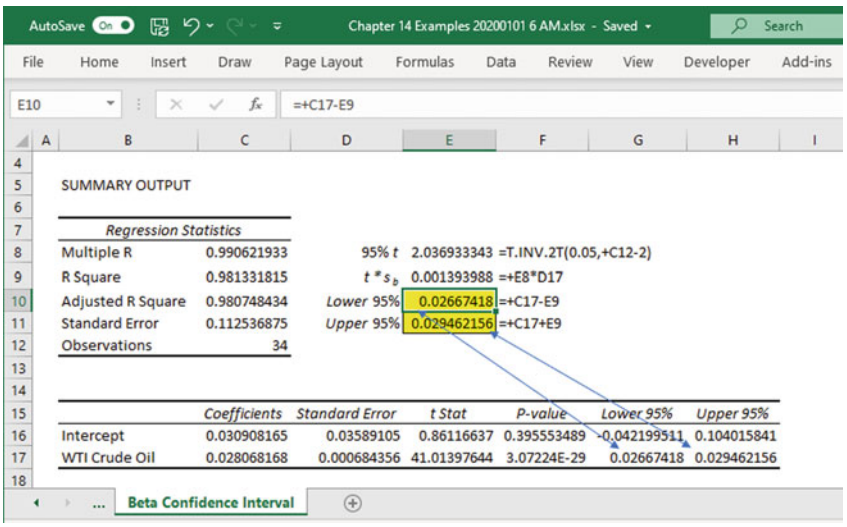
14.5 Confidence Interval of β

Another way to make inferences about the population parameter is to create a confidence interval. If we are interested in a 95% confidence interval, this would mean that 95% of the confidence intervals created will contain the true population parameter.

The formula for the β confidence interval is

$$b - t_{\alpha/2}s_b \leq \beta \leq b + t_{\alpha/2}s_b$$

where β is the population slope, and b is the sample slope. Cell e10 is the manual calculation of the 95% lower confidence interval. Cell e11 is the manual calculation of the 95% upper confidence interval. Range B15 to H17 is the regression report produced by Excel. Note that cell E10 agrees with cell G17 and that cell E11 agrees with cell H17.

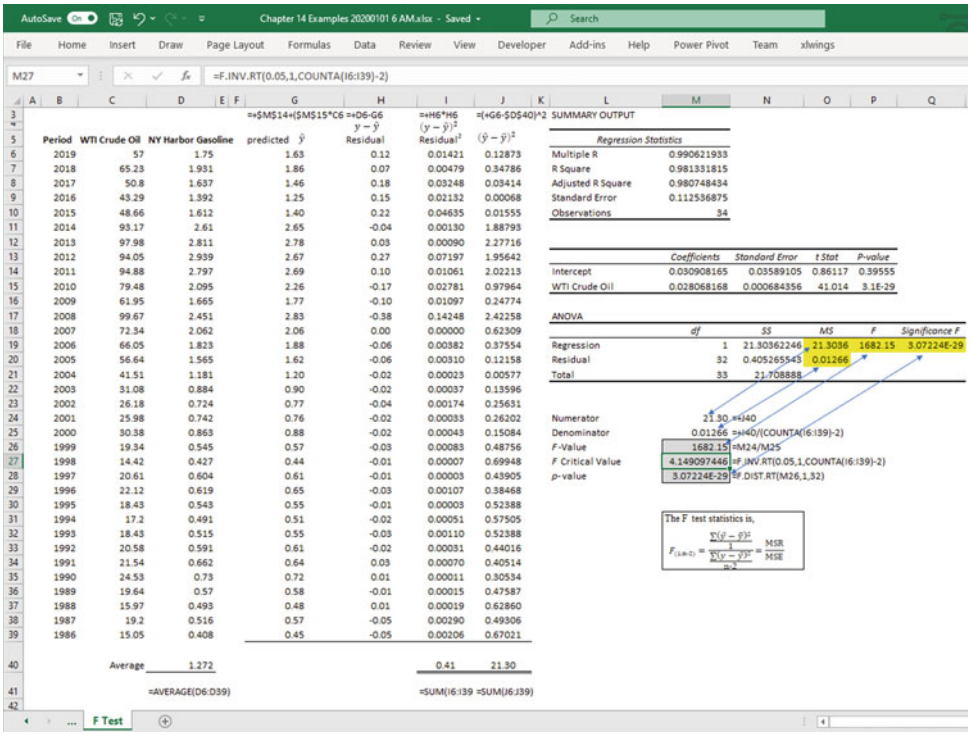


From the calculation above, an increase between \$0.02667 and \$0.0294 for every dollar increase in crude oil is expected.

14.6 *F* Test

Besides using the t statistics to test whether the regression slope is significantly different from zero, we can also use the F statistics to test whether the regression slope is significantly different from zero. The formula for the F test is,

$$F_{(1,n-2)} = \frac{\frac{\sum(\hat{y}-\bar{y})^2}{1}}{\frac{\sum(y-\hat{y})^2}{n-2}} = \frac{MSR}{MSE}$$



Cell M26 manually calculated the F value as 1682.15. Cell M27 shows the F critical value of 4.149 at the significance level when alpha is 0.05. Because the F value, 1682.15, is larger than the critical value of 4.149, we reject the null hypothesis and conclude that crude oil does indeed help explain the price of gasoline.

Note that the manual calculated F value in cell m26 agrees with Excel's regression report value in cell p19. The calculated p-value in cell m28 agrees with Excel's report value in cell q19.

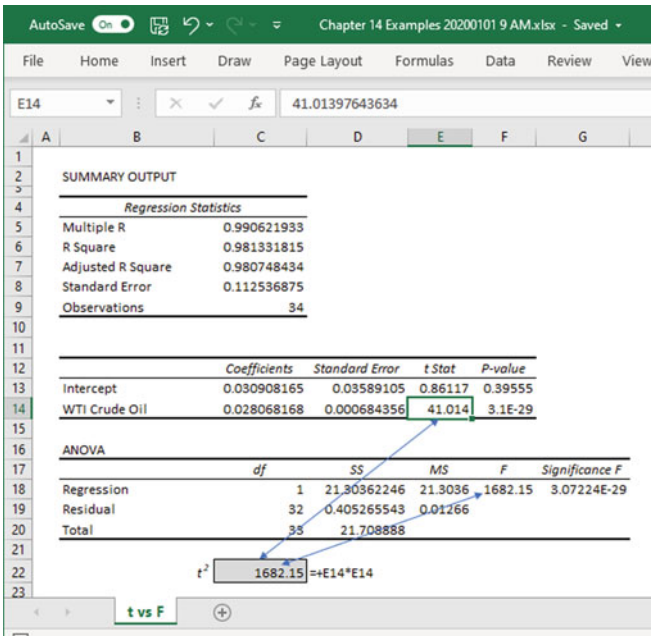
14.7 The Relationship Between the *F*-Test and the *t*-Test

There is a relationship between the F-test and the t-test. The mathematical relationship is

$$F = t^2$$

Our t value is 41.01397643631 and our F value is 1682.15. If we square our t value, we get

$$t^2 = 41.01397643631^2 = 1682.15$$



14.8 Market Model

In finance, there is a model called the *market model*. This model indicates that the return of a stock is linearly related to the return of the market. The equation of the market model is,

$$\boldsymbol{R} = \boldsymbol{a} + \beta \boldsymbol{R_m} + \boldsymbol{\varepsilon}$$

where R is the return of a particular stock and R_m is the return of the market. The coefficient $\boldsymbol{\beta}$ is called the stock's *beta* coefficient. The coefficient indicates how sensitive a stock's return is to the movement of the market. It is common to use the S&P500 index as a proxy for the market. A beta of 1 indicates the stock's return moves in tangent with the market. A beta of 0 indicates that the stock's return is not affected by the market. A beta of 2 indicates that a stock's return is twice as sensitive as the market. A negative beta indicates that a stock's return moves in the opposite direction of the market.

14.9 Yahoo! Finance Beta Screener

As of January 2, 2021, Yahoo! Finance has calculated 15,782 beta calculations for stocks in the United States.

You can go to the URL https://finance.yahoo.com/screener/new to filter for stocks with specific betas. The Yahoo! Finance screener indicated that there were only two mega-cap companies with a beta greater than 2. Those two mega-cap stocks are Tesla, Inc. and Advanced Micro Devices. Mega-cap stocks have a market capitalization of greater than $200 billion.

The Yahoo! Finance screener indicated that there were only four large-cap companies with a beta less than −0.2. Those four companies are shown below.

Large-cap stocks have a market capitalization of greater than $10 billion. Mid-cap stocks have a market capitalization between $2 billion and $10 billion. Small-cap stocks have a market capitalization between $300 million and $2 billion.

14.10 Historical Monthly Data from Yahoo! Finance

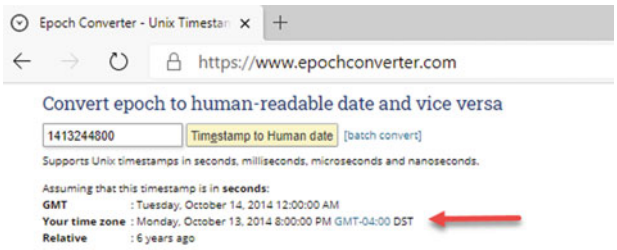
The following is the URL to retrieve monthly Apple Inc. pricing data from October 2014 to January 2021 Yahoo! Finance, https://finance.yahoo.com/quote/AAPL/history?period1=1413244800&period2=1609804800&interval=1mo&filter=history&frequency=1mo&includeAdjustedClose=true.

The following is the URL to retrieve monthly Walmart pricing data from October 2014 to January 2021, https://finance.yahoo.com/quote/WMT/history?period1=1413244800&period2=1609804800&interval=1mo&filter=history&frequency=1mo&includeAdjustedClose=true.

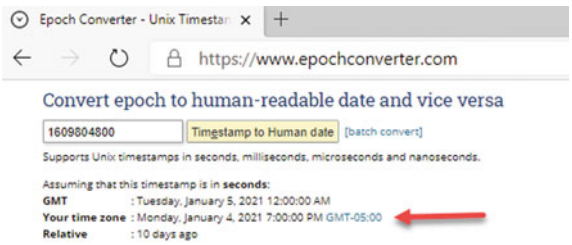
The only difference between the Apple Inc. and Walmart URL is the ticker symbol in the URL.

The numbers after "period1" and "period2" in the URLs are Epoch time. The URL https://www.maketecheasier.com/what-is-epoch-time/ has a more detailed discussion on epoch time.

You can use the URL https://www.epochconverter.com/ to convert epoch time to regular time. The epoch time 1413244800 is Monday, October 13 at 8 pm.



The epoch time 1609804800 is Monday, January 4, 2021.



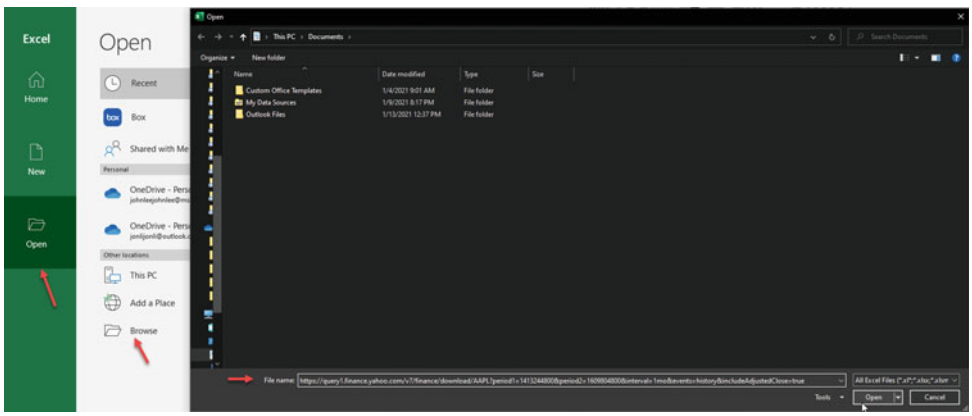
The following shows the Yahoo! Finance web page for the URL, https://finance.yahoo.com/quote/AAPL/history?period1=1413244800&period2=1609804800&interval=1mo&filter=history&frequency=1mo&includeAdjustedClose=true.

The web page has a download link. If you right-mouse click on the link, you can get the URL to retrieve Apple Inc.'s data as a "CSV" file.

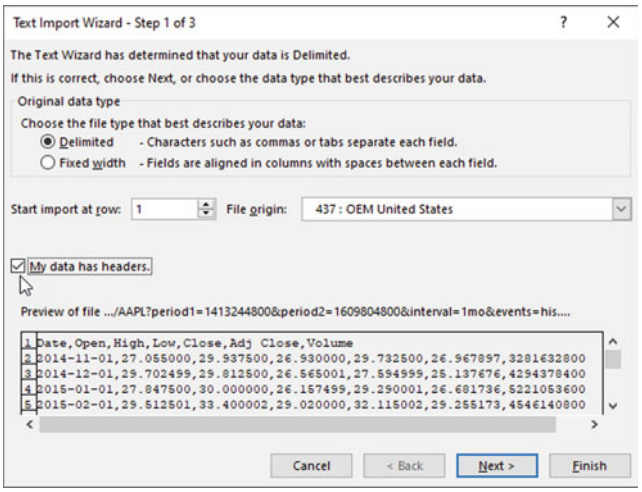
Apple's URL to download monthly data from October 2014 to January 2021 is, https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1413244800&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

14.10.1 Excel's Import Text Wizard

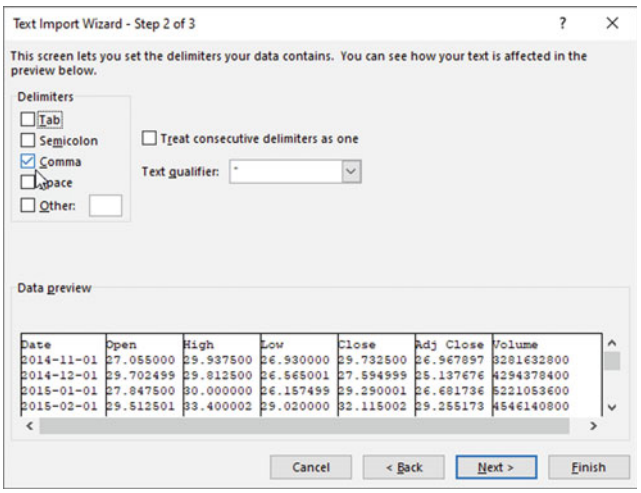
You can retrieve Apple Inc.'s monthly pricing data in Excel by "opening" the URL.



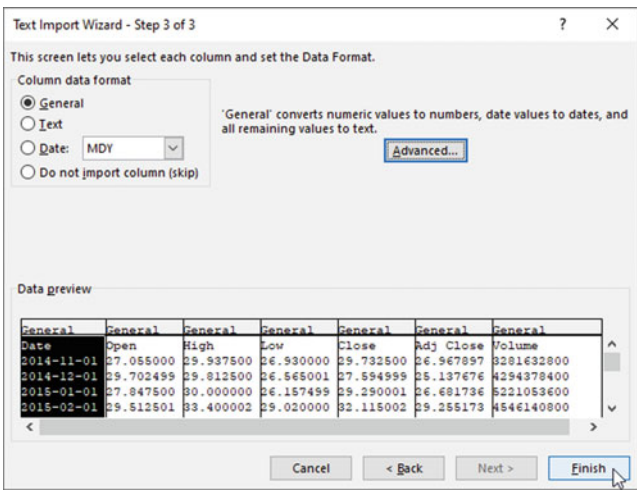
After clicking on the *Open* button, step 1 of the *Text Import Wizard* will be displayed.



Click on the *My Data has headers* check box and then press the *Next* button.

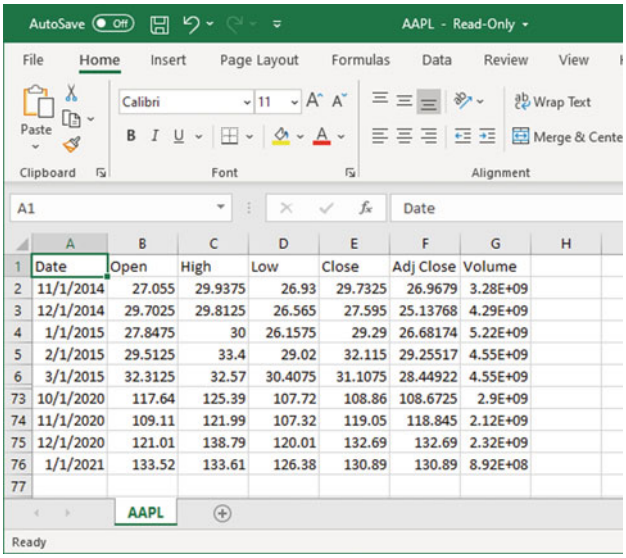


In step 2, make *Comma* as the delimiter and then press the *Next* button.



In step 3, click on the *Finish* button.

Excel will put Apple Inc.'s monthly pricing data on the worksheet.



	A	B	C	D	E	F	G	H
1	Date	Open	High	Low	Close	Adj Close	Volume	
2	11/1/2014	27.055	29.9375	26.93	29.7325	26.9679	3.28E+09	
3	12/1/2014	29.7025	29.8125	26.565	27.595	25.13768	4.29E+09	
4	1/1/2015	27.8475	30	26.1575	29.29	26.68174	5.22E+09	
5	2/1/2015	29.5125	33.4	29.02	32.115	29.25517	4.55E+09	
6	3/1/2015	32.3125	32.57	30.4075	31.1075	28.44922	4.55E+09	
73	10/1/2020	117.64	125.39	107.72	108.86	108.6725	2.9E+09	
74	11/1/2020	109.11	121.99	107.32	119.05	118.845	2.12E+09	
75	12/1/2020	121.01	138.79	120.01	132.69	132.69	2.32E+09	
76	1/1/2021	133.52	133.61	126.38	130.89	130.89	8.92E+08	
77								

14.11 Market Model of Apple Inc. in Excel

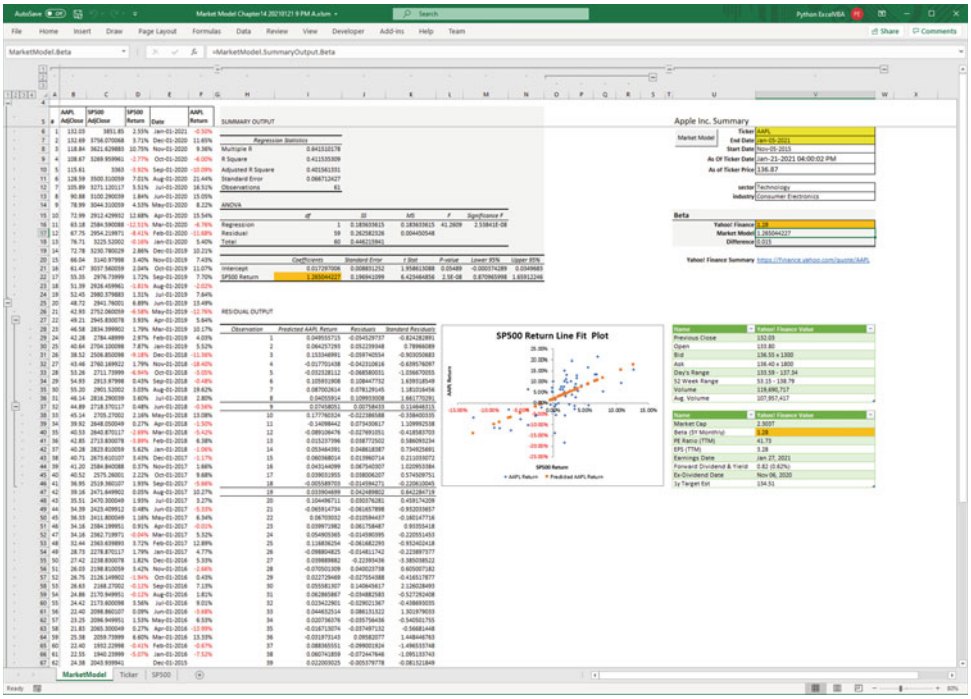
Yahoo! Finance reports the 5-year beta for most stocks. The web page https://finance.yahoo.com/quote/aapl reported that the 5-year beta of Apple Inc. on January 21, 2021, was 1.28.

To create the market model of Apple Inc., we would need Apple Inc.'s pricing data and S&P 500 pricing data. We will use a 60-month time period like Yahoo! Finance. We will get 61 months of pricing data from Yahoo! Finance to calculate 61 returns. Regressing 61 returns has 60 degrees of freedom.

The following is the URL for the 62-month S&P 500 pricing data from Yahoo! Finance. Notice that the ticker for the S&P 500 is *^GSPC*, https://query1.finance.yahoo.com/v7/finance/download/^GSPC?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

The following is the URL for the 62-month Apple Inc. data from Yahoo! Finance, https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

The workbook below shows the market model's regression report for Apple Inc.



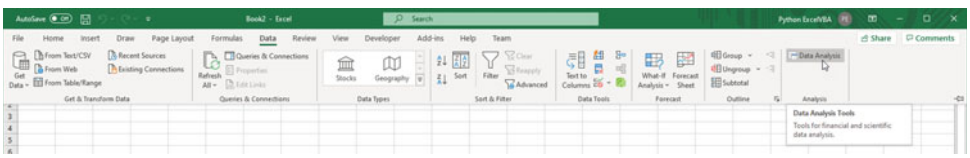
Sixty-two months of prices for Apple Inc. and S&P 500 were retrieved from Yahoo! Finance. The URL https://finance.yahoo.com/quote/AAPL/history is the source for the historical data for Apple Inc. The URL https://finance.yahoo.com/quote/^GSPC/history is the source for the historical data for the S&P 500. The 62 monthly data were used to calculate 61 monthly returns. The monthly returns for Apple Inc. are shown in cells F6 to F66. The monthly returns for the S&P 500 are shown in cell D6 to D66.

The regression model calculated Apple Inc.'s beta as 1.265. The market model's beta number is shown in cell I22. Yahoo! Finance reported that the beta for Apple Inc. on January 21, 2021, was 1.28. Yahoo! Finance's beta number is shown in cell v40. We interpret this to mean that for each 1% increase in the return of the S&P 500, the average increase in Apple Inc.'s return is 1.28%.

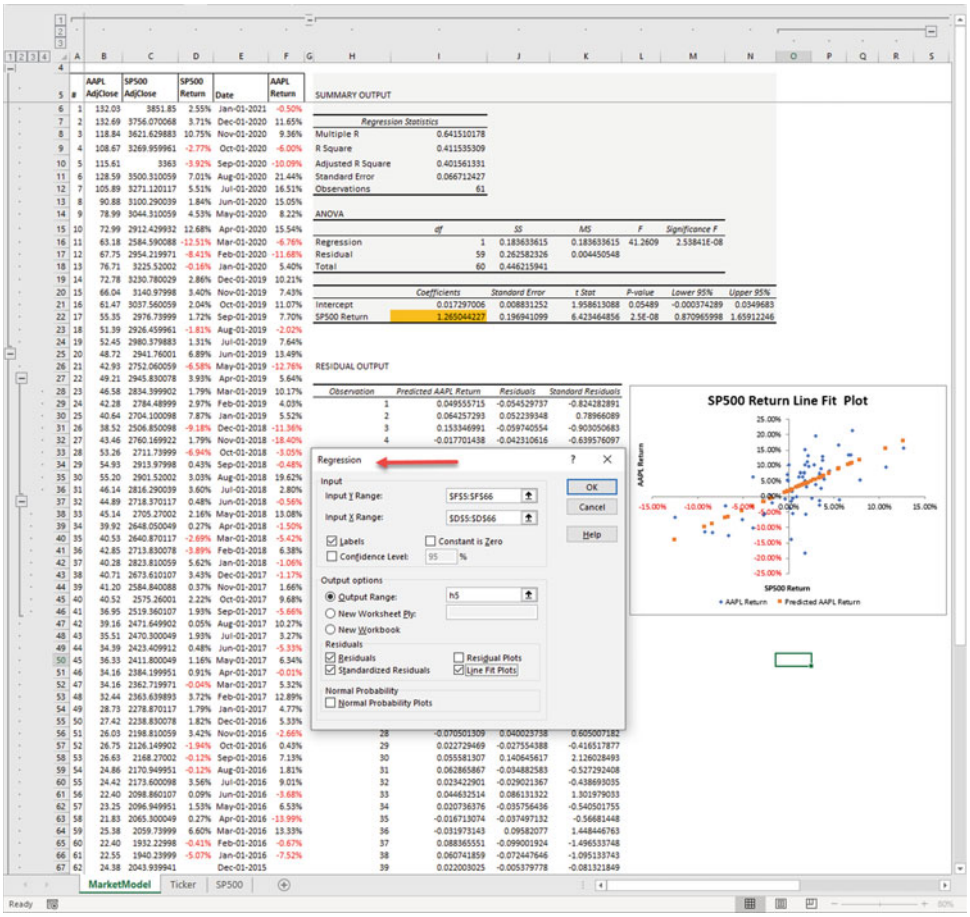
Cell I9 of the regression report indicated that the coefficient of determination was 0.4194; this means that 41.94% of the variation of Apple Inc.'s return is explained by the variation in the S&P 500 returns.

14.11.1 Data Analysis and Regression Report

Excel's *Data Analysis* feature was used to create the market model regression report. The *Data Analysis* tool is part of the *Data* tab.



Below shows the *Regression* screen from the *Data Analysis* tool.



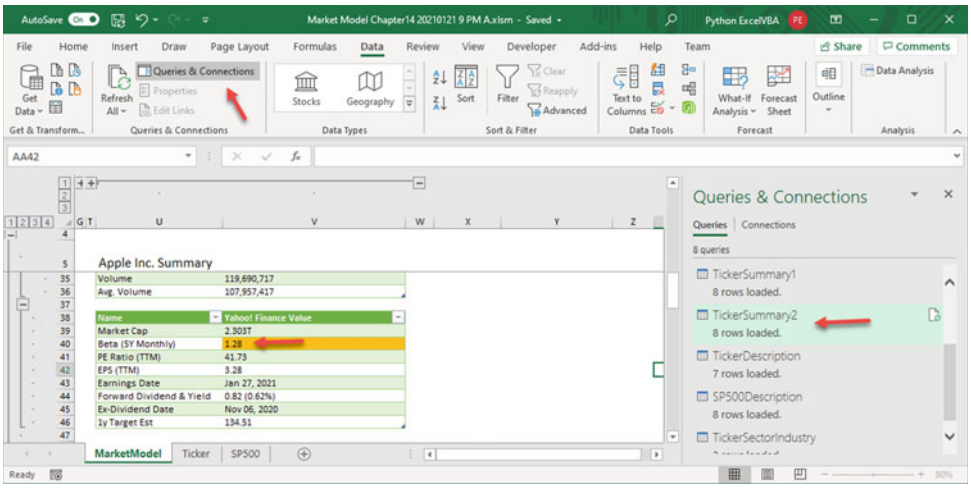
The summary regression output is shown in cells H5:N22. Apple Inc.'s returns are the "*y*" input range. S&P 500 returns are the x input range.

The *residual output* report starts on cell H26. The residual out report shows the predicted value of each return and the residual value. The residual value is the difference between the model's predicted value and the actual value of the model. The residual output also shows the standard residual value. The standard residual value indicates how many standard deviations a return is away from the return's mean.

The regression report also produces a graph of Apple Inc.'s actual return and the predicted value of Apple Inc.'s return. The graph is at cell M23.

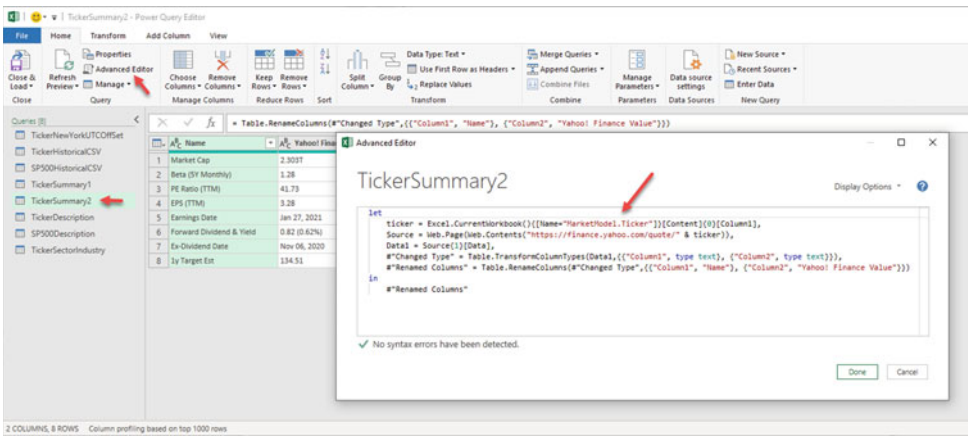
14.11.2 Yahoo! Finance Beta and Power Query

Cell v40 shows the 1.28 beta value that Yahoo! Finance reported for Apple Inc.

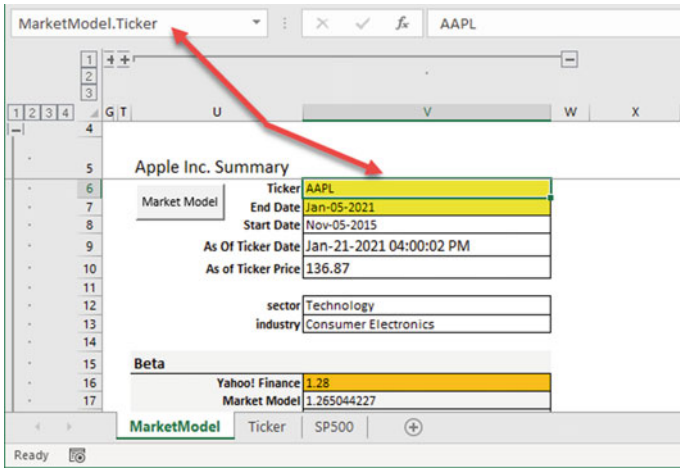


Cells U38:V36 is the red section shown below from the URL https://finance.yahoo.com/quote/AAPL.

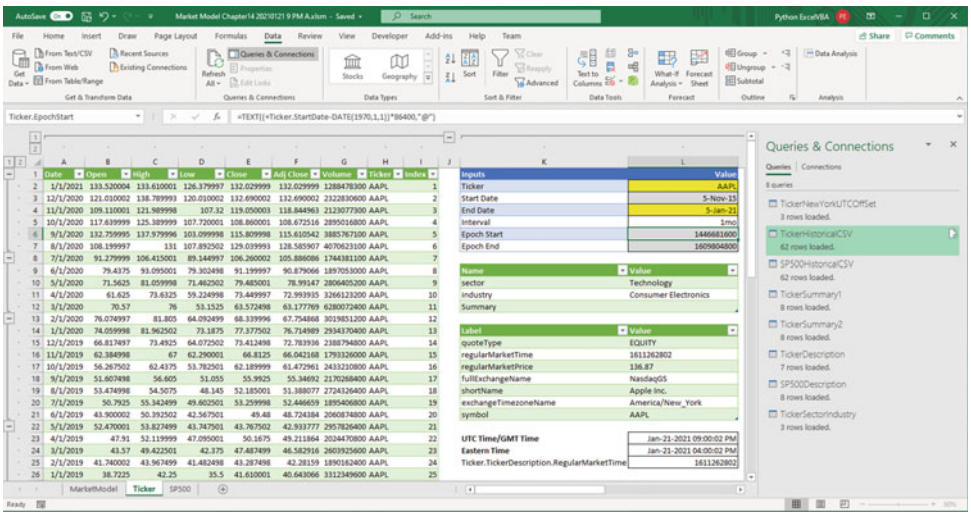
Excel used power query *TickerSummary2* to retrieve this Yahoo! Finance summary section. The following is the power query code to return Yahoo! Finance summary section. The power query code is stored in the *Advanced Editor* menu in the *Home* tab.



The power query code uses the range name "MarketModel.Ticker" to know which ticker summary to retrieve. Cell v6 has the range name "MarketModel.Ticker."



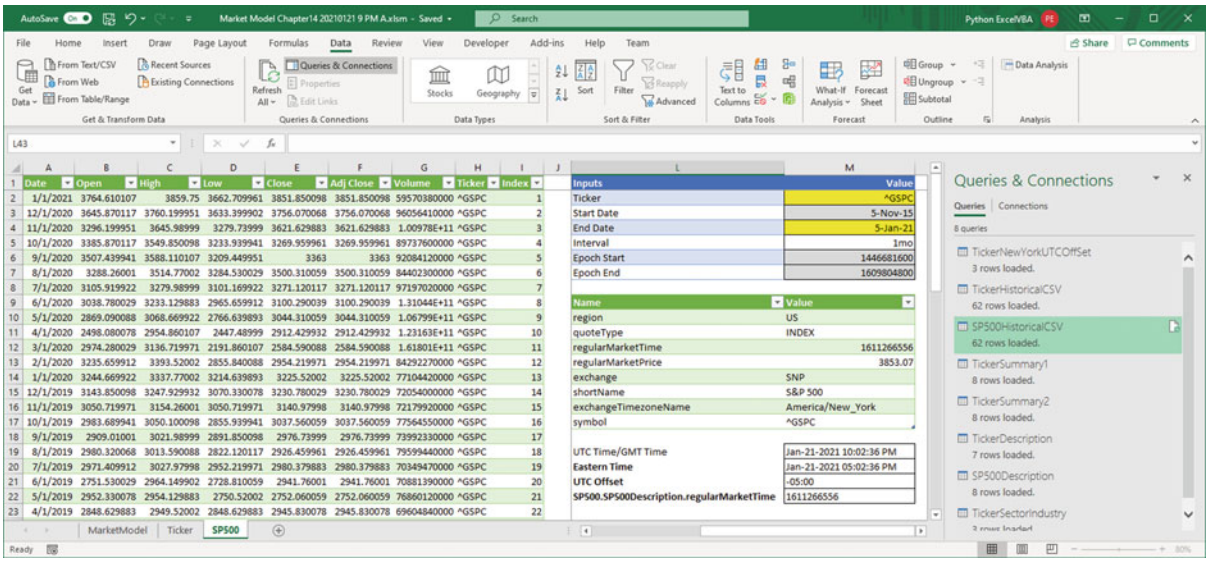
14.11.3 Yahoo! Finance Ticker Historical Data



The worksheet "Ticker" contains monthly historical pricing ticker data. The "Ticker" worksheet uses power query *TickerHistoricalCSV* to retrieve this historical ticker data from Yahoo! Finance. The following is the power query code to retrieve historical ticker data from Yahoo! Finance.

The power query code used the following URL to retrieve the ticker's monthly prices, https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

14.11.4 Yahoo! Finance S&P500 Historical Data



The worksheet "SP500" contains the monthly historical S&P500 prices. The "SP500" worksheet uses power query *SP500HistoricalCSV* to retrieve this historical S&P500 data from Yahoo! Finance. The following is the power query code to retrieve historical ticker data from Yahoo! Finance.

The power query code used the following URL to retrieve the S&P500's monthly prices, https://query1.finance.yahoo.com/v7/finance/download/^GSPC?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

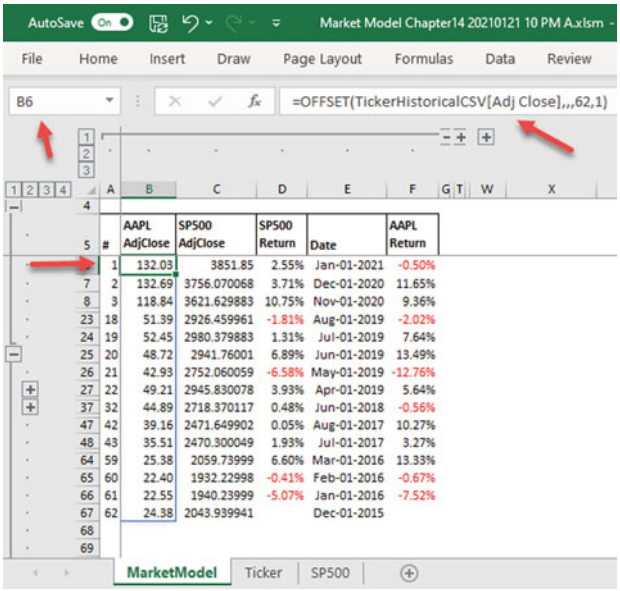
14.11.5 Calculating Rate of Return

The rate of return calculations for Apple Inc. and the S&P 500 are shown in columns D and F.

14.11.5.1 Dynamic Arrays

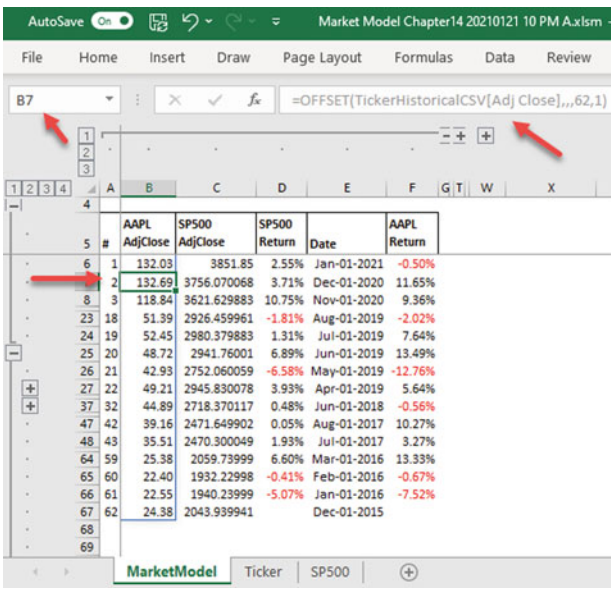
To calculate the rate of return for Apple Inc., the closing data are first retrieved from the *Ticker* worksheet. The formula = *OFFSET(Ticker_Historical[Adj Close],,,62,1)* is used and put in cell b6. This formula pulls in the *Adj Close* column from the *Ticker* worksheet and places the data to cells b6 to b67. This range is highlighted, which indicates that the return

data is a dynamic array. The formula in cell b6 also puts a copy of the formula in cells b7 to b67. The font for the formula in the first cell of the dynamic array formula is bold.



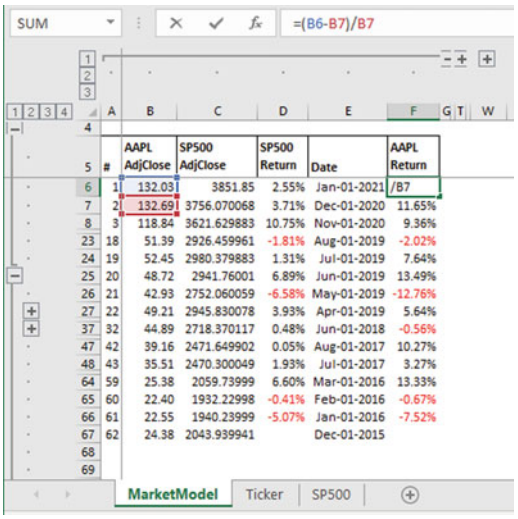
#	AAPL AdjClose	SP500 AdjClose	SP500 Return	Date	AAPL Return
1	132.03	3851.85	2.55%	Jan-01-2021	-0.50%
2	132.69	3756.070068	3.71%	Dec-01-2020	11.65%
3	118.84	3621.629883	10.75%	Nov-01-2020	9.36%
18	51.39	2926.459961	-1.81%	Aug-01-2019	-2.02%
19	52.45	2980.379883	1.31%	Jul-01-2019	7.64%
20	48.72	2941.76001	6.89%	Jun-01-2019	13.49%
21	42.93	2752.060059	-6.58%	May-01-2019	-12.76%
22	49.21	2945.830078	3.93%	Apr-01-2019	5.64%
32	44.89	2718.370117	0.48%	Jun-01-2018	-0.56%
42	39.16	2471.649902	0.05%	Aug-01-2017	10.27%
43	35.51	2470.300049	1.93%	Jul-01-2017	3.27%
59	25.38	2059.73999	6.60%	Mar-01-2016	13.33%
60	22.40	1932.22998	-0.41%	Feb-01-2016	-0.67%
61	22.55	1940.23999	-5.07%	Jan-01-2016	-7.52%
62	24.38	2043.939941		Dec-01-2015	

All the other font for the other cell formulas in the array formula is light grey.



#	AAPL AdjClose	SP500 AdjClose	SP500 Return	Date	AAPL Return
1	132.03	3851.85	2.55%	Jan-01-2021	-0.50%
2	132.69	3756.070068	3.71%	Dec-01-2020	11.65%
3	118.84	3621.629883	10.75%	Nov-01-2020	9.36%
18	51.39	2926.459961	-1.81%	Aug-01-2019	-2.02%
19	52.45	2980.379883	1.31%	Jul-01-2019	7.64%
20	48.72	2941.76001	6.89%	Jun-01-2019	13.49%
21	42.93	2752.060059	-6.58%	May-01-2019	-12.76%
22	49.21	2945.830078	3.93%	Apr-01-2019	5.64%
32	44.89	2718.370117	0.48%	Jun-01-2018	-0.56%
42	39.16	2471.649902	0.05%	Aug-01-2017	10.27%
43	35.51	2470.300049	1.93%	Jul-01-2017	3.27%
59	25.38	2059.73999	6.60%	Mar-01-2016	13.33%
60	22.40	1932.22998	-0.41%	Feb-01-2016	-0.67%
61	22.55	1940.23999	-5.07%	Jan-01-2016	-7.52%
62	24.38	2043.939941		Dec-01-2015	

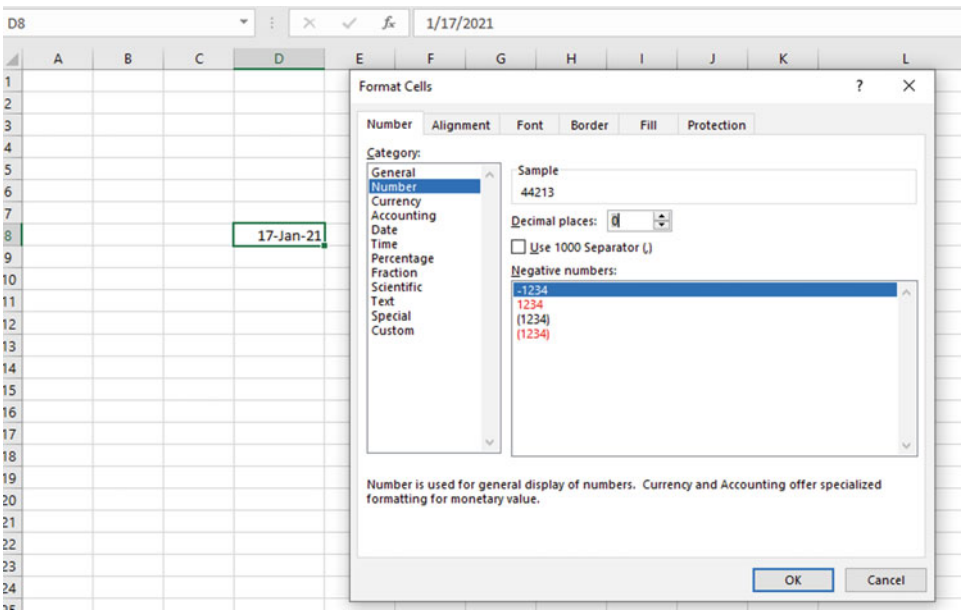
The following shows the formula for calculating the monthly rate of return for January 2021.



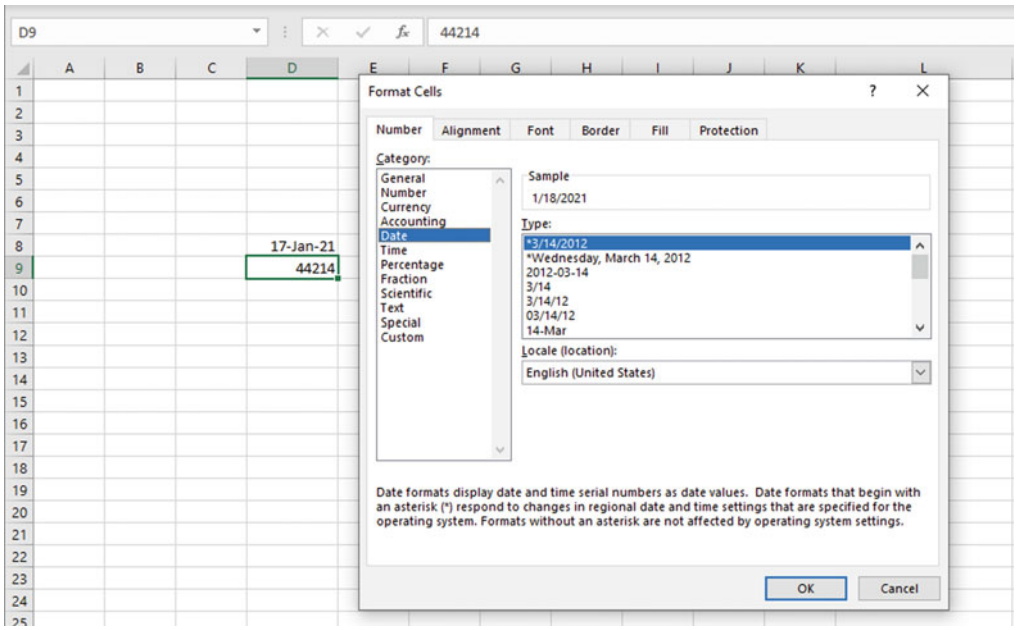
The procedure for calculating the rate of return for the S&P 500 is the same as calculating the rate of return for Apple Inc.

14.11.6 Date, Time, and Epoch Time

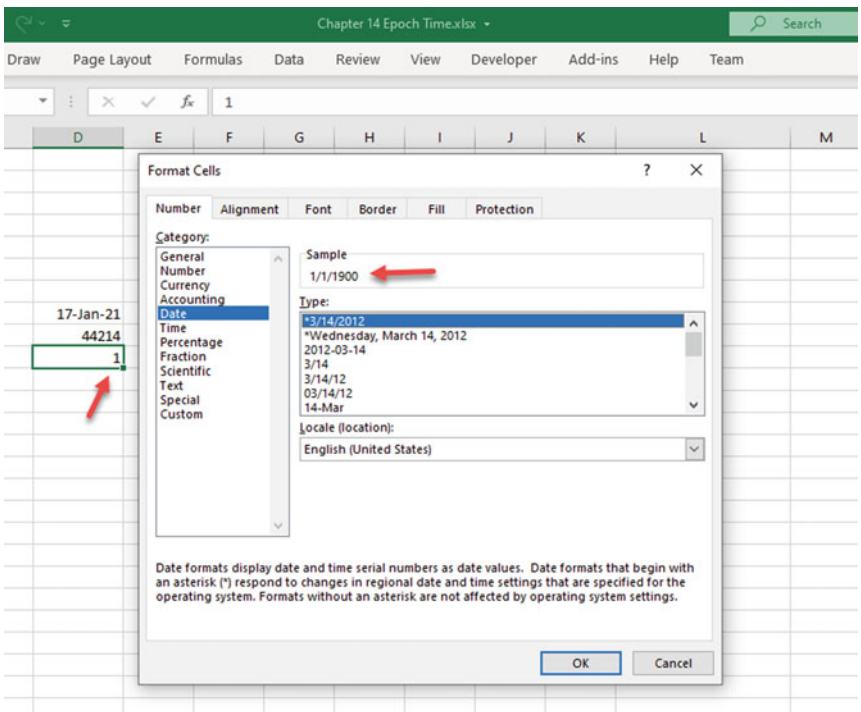
Dates in Excel are actually saved as numbers. For example, formatting the date January 17, 2021, to a number in Excel would show the number 44213.



The number 44214 would represent January 18, 2021.



The value of 1 is considered as January 1, 1900, in Excel. January 1, 1900 is considered as the *Epoch* time in Excel because it's the starting point of time in Excel.



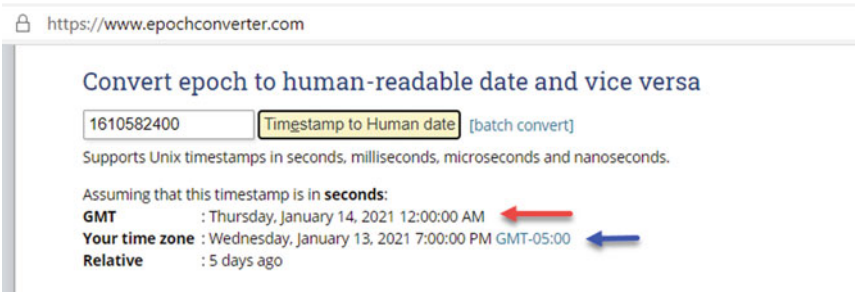
Different systems have different *Epoch* times. For the Mac, Epoch time is January 1, 1904. For Microsoft Windows Epoch time January 1, 1601. For Unix, the Epoch time is January 1, 1970. Most data providers on the internet use Unix Epoch time. Yahoo! Finance uses Epoch time.

There are many time zones in the world. Many people around the world are interested in the same data set. When reporting a time for a data set, there needs to be an anchor time zone used. The anchor time zone is Greenwich Mean Time (GMT). London is located in the GMT time zone. Unix Epoch time anchored on the GMT time zone is also known as *Unix Time*.

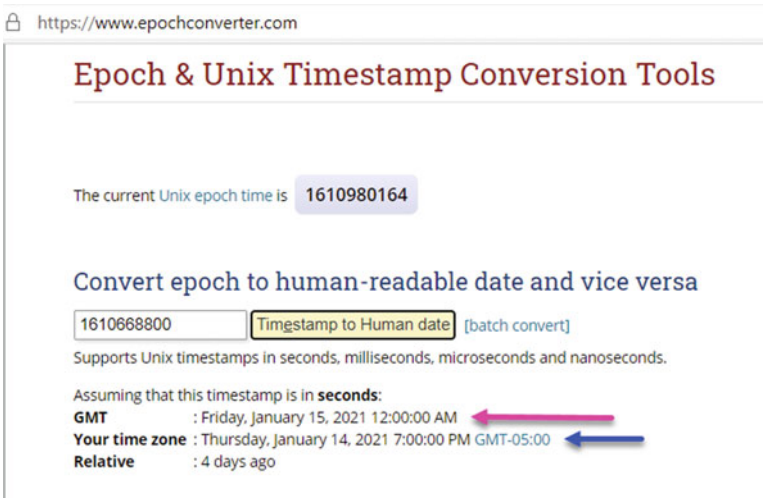
14.11.6.1 Epoch Unix Time

Yahoo! Finance uses *Epoch Unix Time* in its URL to indicate time. For example, the URL, https://finance.yahoo.com/quote/AAPL/history?period1=1610582400&period2=1610668800&interval=1d&filter=history&frequency=1d&includeAdjustedClose=true will show the historical price for January 14, 2021.

The number 1610582400 is January 14, 2021, 12:00:00 a.m. Greenwich Mean Time, but January 13, 2021, 7 p.m. Eastern Standard Time. Use the URL https://www.epochconverter.com/ to convert epoch time to regular time.



The number 1610668800 is January 15, 2021, 12:00:00 a.m. Greenwich Mean Time, but January 14, 2021, 7 p.m. Eastern Standard Time.

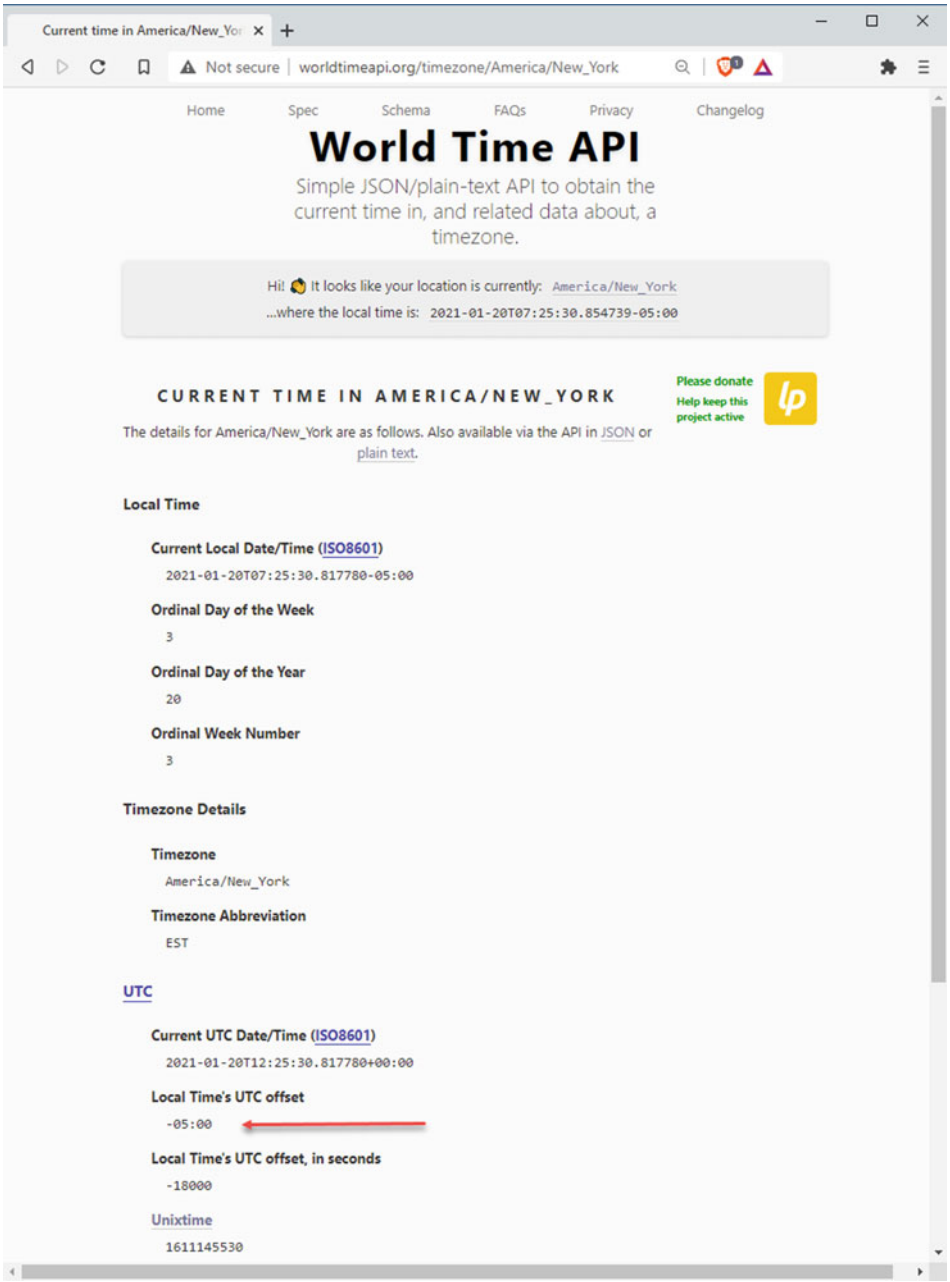


14.11.6.2 New York Time and UTC Offset

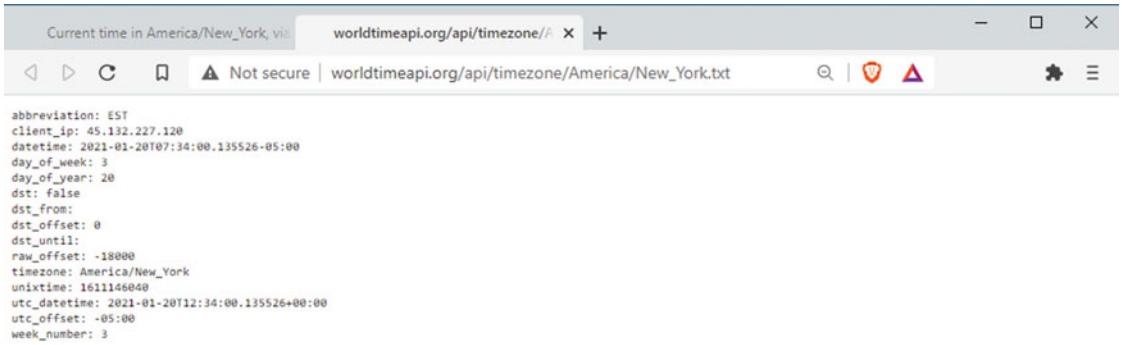
UTC, which stands for *Coordinated Universal Time*, is another time acronym that is used a lot. While there are technical differences between UTC time and GMT time, for all practical purposes, they are used as the same time. A lot of applications report in UTC time. Go to the URL https://www.timeanddate.com/time/gmt-utc-time.html to learn about the differences between UTC time and GMT time.

During standard time, we need to subtract 5 h from UTC time to get New York time. During daylight savings time, we need to subtract 4 h from UTC time to get New York time. Most people do not remember when daylight saving time is or when standard time is, and for most people. It is also confusing how to create a formula to determine if its daylight saving time or standard time.

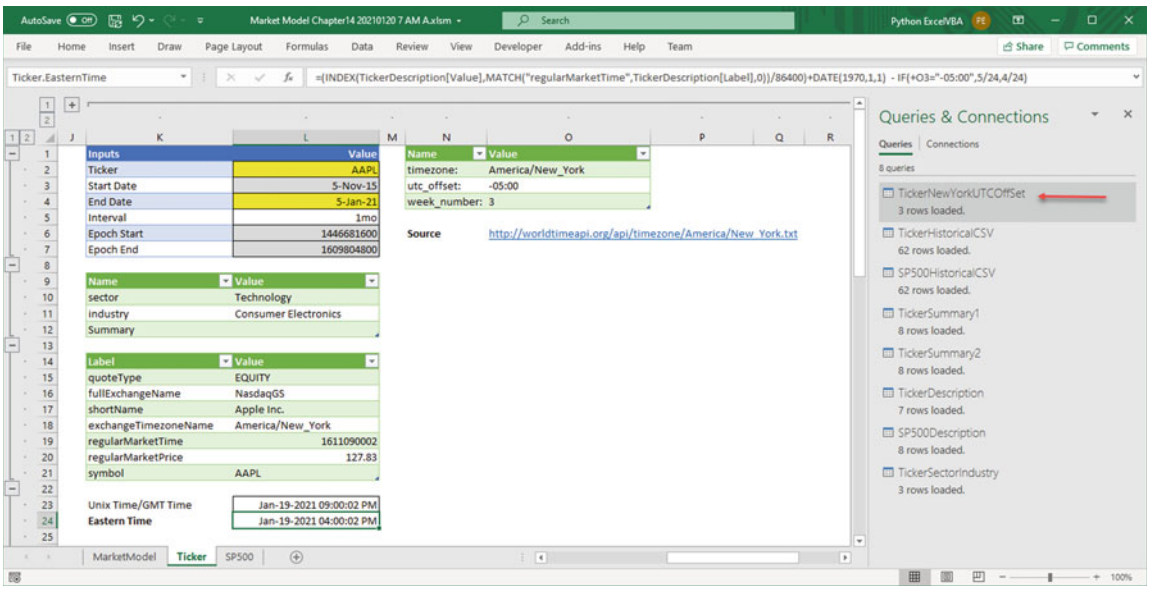
For any time, the website http://worldtimeapi.org/timezone/America/New_York gives specific time information for New York City.



One thing that the website provides is the UTC offset to get from UTC time to New York City time. This website also provides the API http://worldtimeapi.org/api/timezone/America/New_York.txt that also provides the same information.



The worksheet "Ticker" contains the New York City UTC offset information. The "Ticker" worksheet uses power query *TickerNeweYorkUTCOffset* to retrieve New York City time information from World Time API.

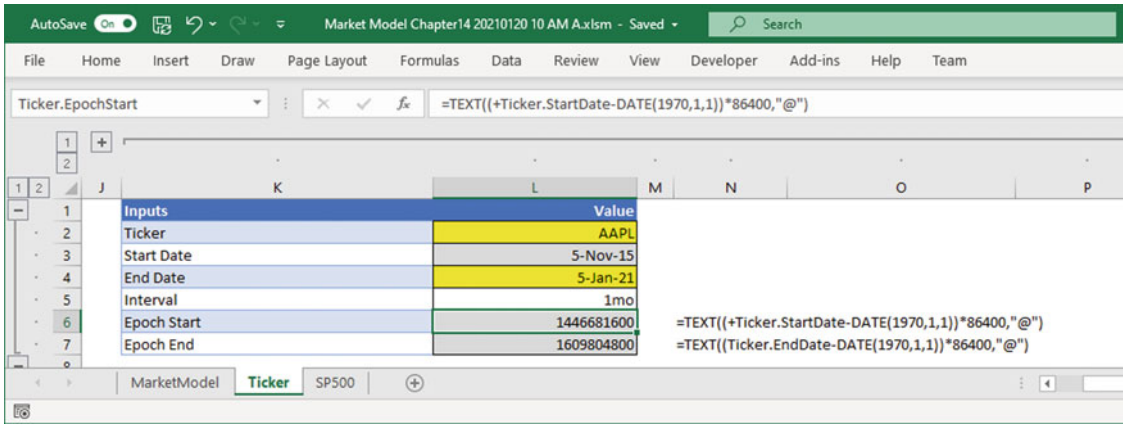


The following is the power query code to retrieve New York City time information from World Time API.

14.11.7 Converting to and from Epoch Time

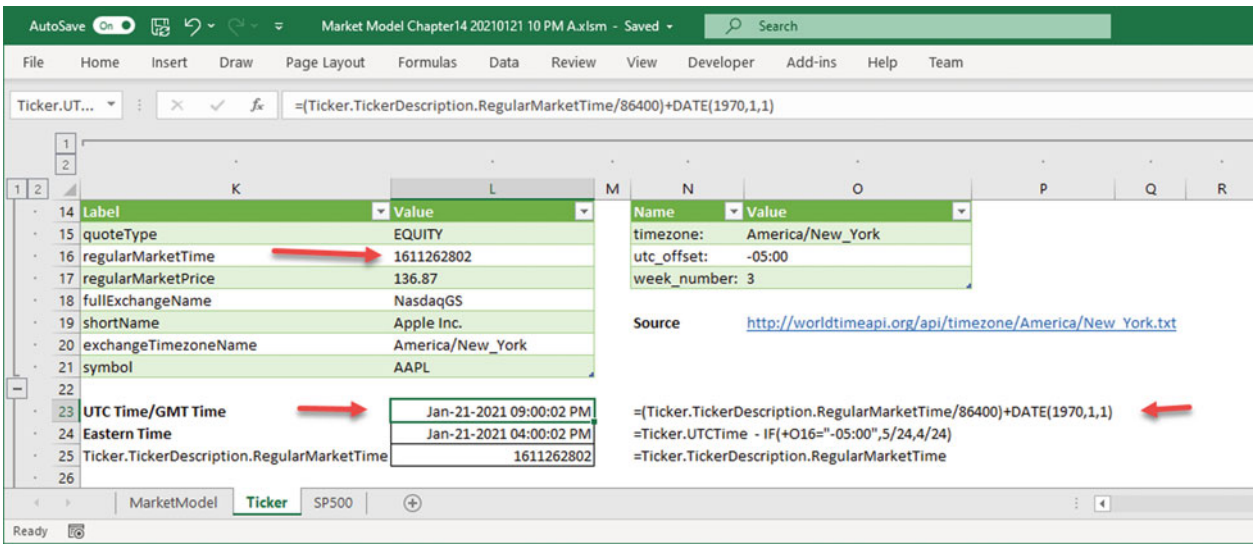
14.11.7.1 Converting to Epoch Time

Below shows the formulas to convert "Start Date" and "End Date" to *Epoch Time*.



To convert January 5, 2021 to epoch time, one would first subtract the date by January 1, 1970, and then times the result by 86,400. Doing this would give us the epoch time of 1609804800. Cell n7 shows the Excel formula to convert the date January 5, 2021 to epoch time.

14.11.7.2 Converting from Epoch Time

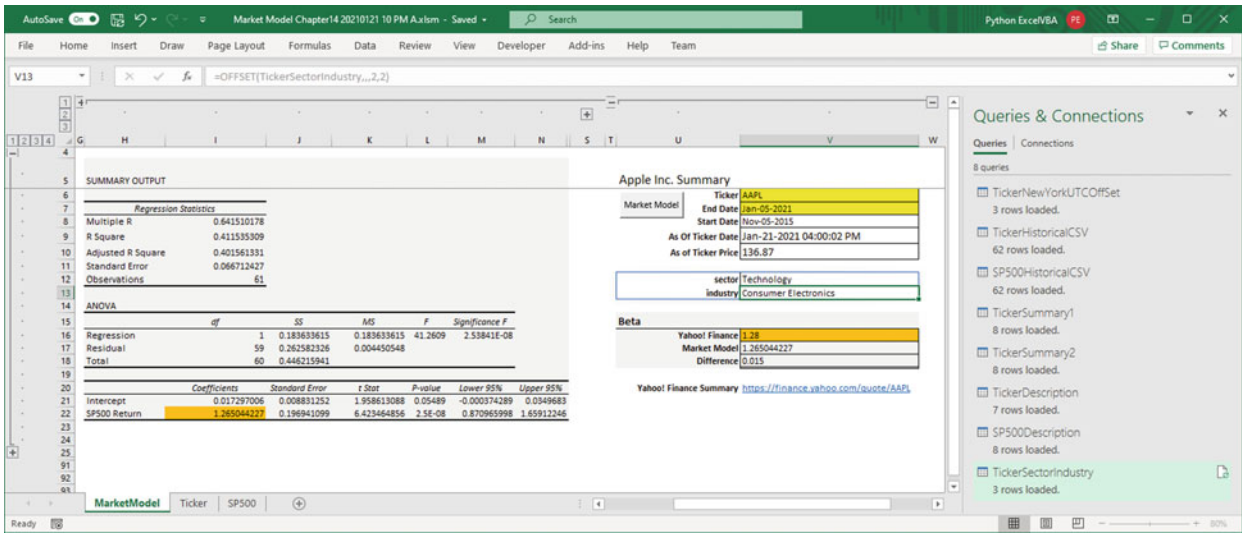


Above shows that at the epoch time 1611262802, the price of Apple Inc. computers was 136.87. To convert this epoch time to regular time, we would first divide by 86,400 from the date January 1, 1970, to get the result. The resulting date and time is January 21, 2021 at 9:00:02 p.m. This date and time is GMT time. To get New York time, we need to subtract 5 h if it's standard time and 4 h if it's daylight-saving time. Cell O16 indicates what needs to be adjusted to get New York time. The source for cell O16 is http://worldtimeapi.org/api/timezone/America/New_York.txt.

Cell N23 shows the Excel formula to convert epoch time to regular time, which is always GMT time.

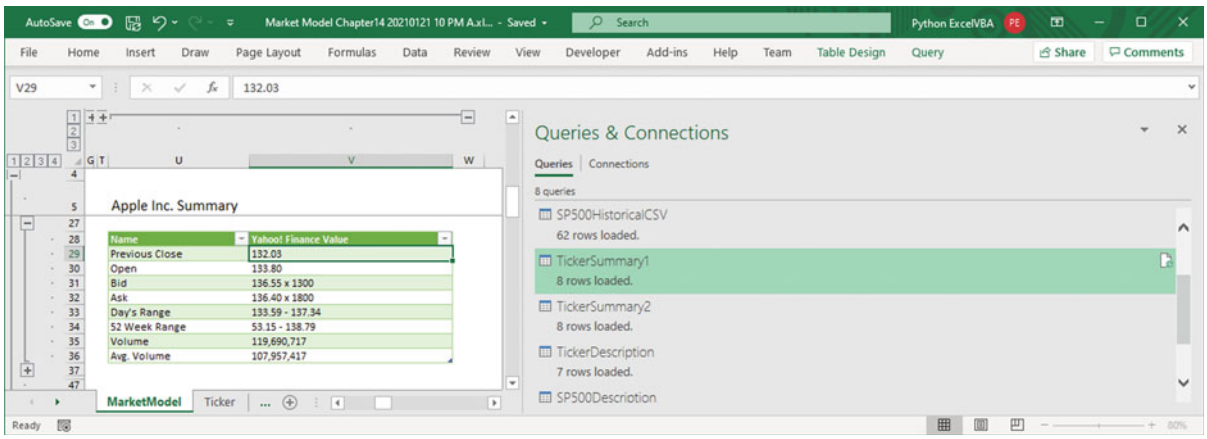
Cell N24 shows the Excel formula on how to convert GMT time to New York time.

14.11.8 Other Power Queries



The market model Excel workbook contains eight power queries. Besides retrieving historical ticker data, there are power queries that retrieve other information about a ticker. We have already seen four of the eight power queries. In this section, we will examine the other four queries.

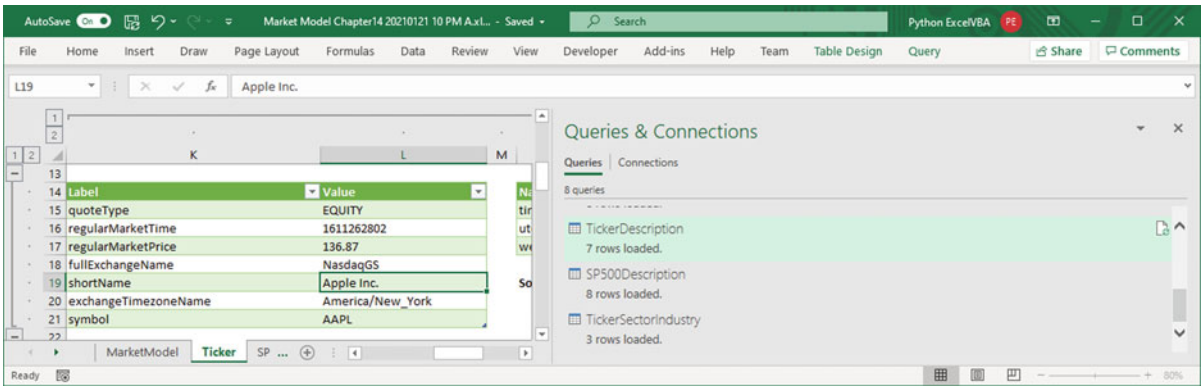
14.11.8.1 TickerSummary1



Cells U28:V36 is the green section shown below from the URL https://finance.yahoo.com/quote/AAPL.

The following is the *TickerSummary1* power query m code.

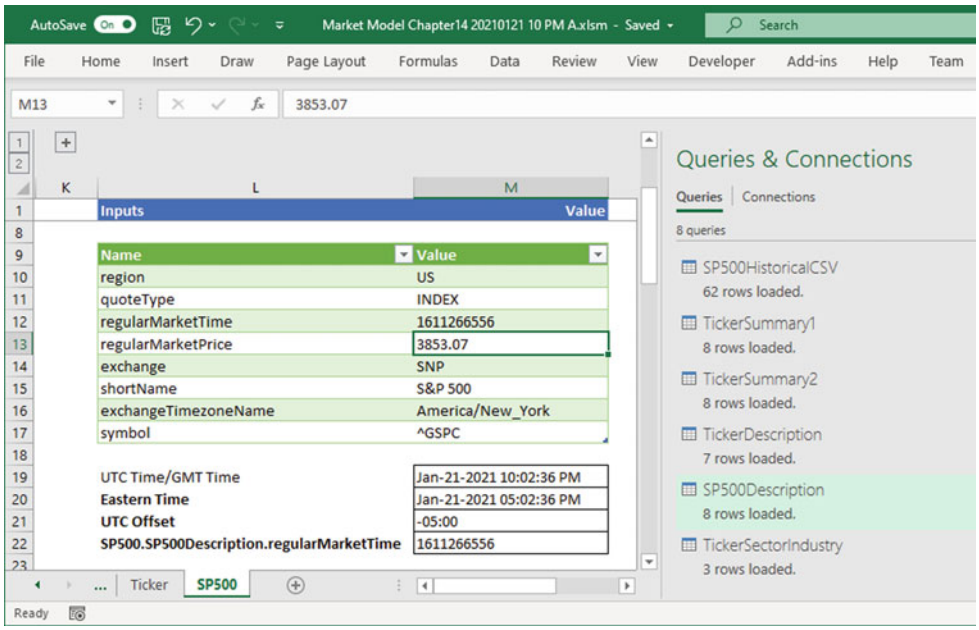
14.11.8.2 TickerDescription



The *TickerDescription* power query returns the security type for the ticker, the time and date of the last price available for the ticker, the exchange the ticker was traded on, the time zone where the ticker was traded, and the name of the ticker.

The following is the *TickerDescription* power query m code.

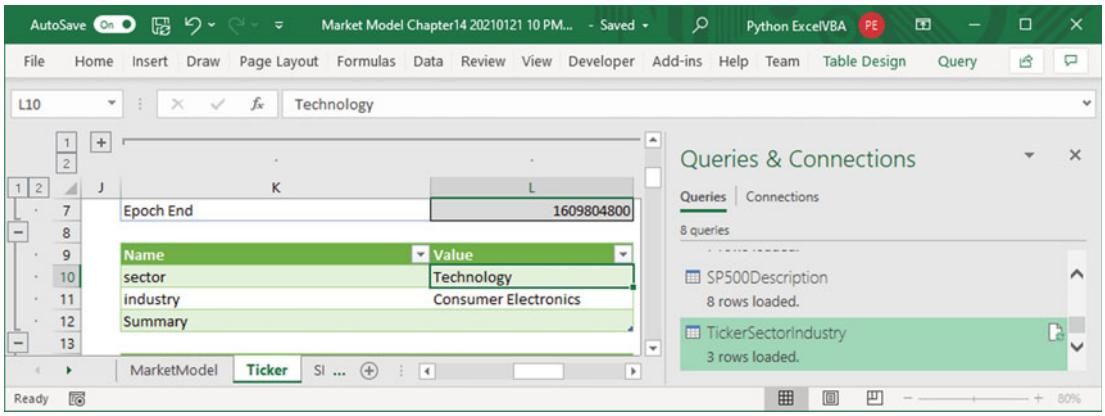
14.11.8.3 SP500Description



The *SP500Description* power query returns the security type for the S&P 500, the time and date of the last available price for the S&P 500, the exchange that published the S&P500 price, the time zone of the exchange and the name of the ticker.

The following is the *TickerDescription* power query m code.

14.11.8.4 Sectors and Industry



The *TickerSectorIndustry* power query returns the sector and industry of a ticker.

The following is the *TickerSectorIndusry* power query m code.

14.12 Market Model of the Clorox Company in Excel

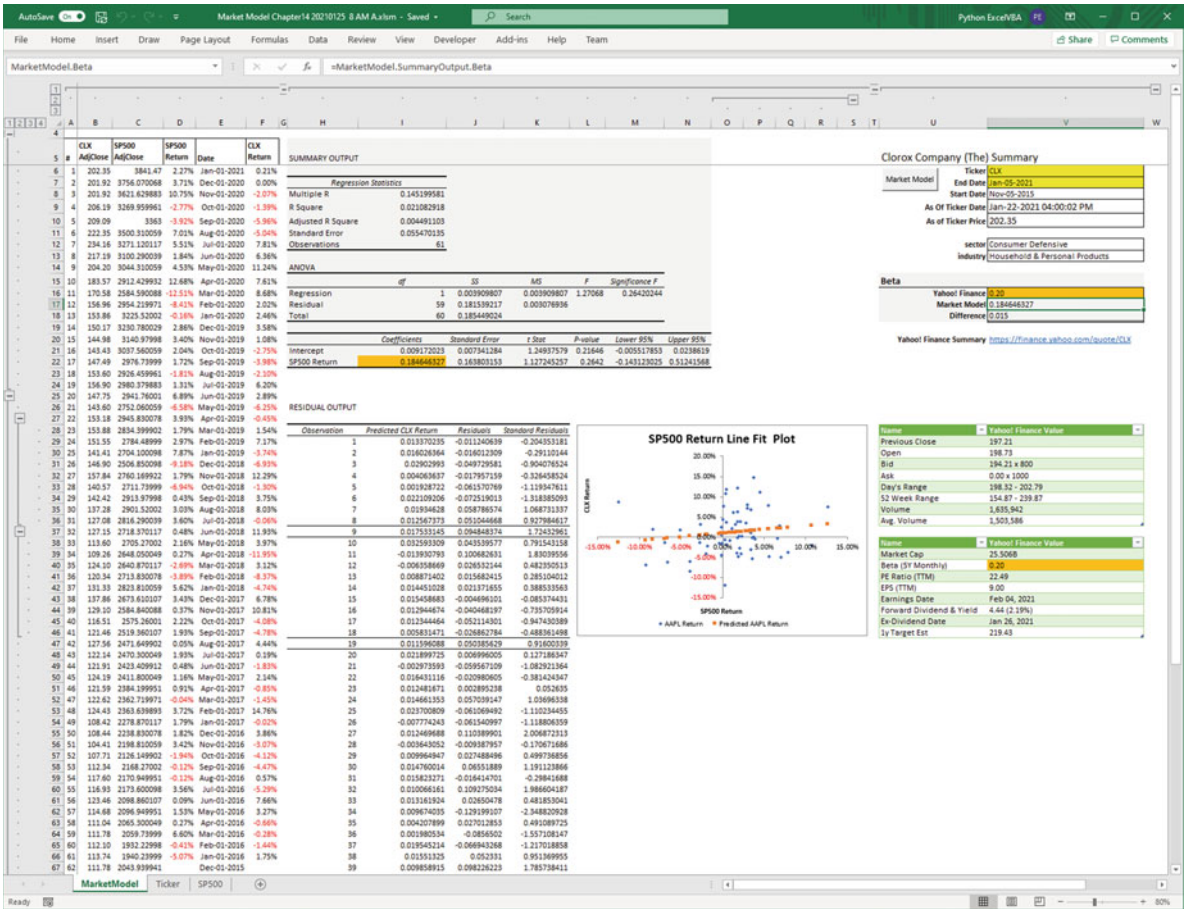
The web page https://finance.yahoo.com/quote/clx reported that the 5-year beta of The Clorox Company on January 2, 2021 was 0.2

14.12.1 Regression Report

The following is the URL for the 62-month S&P 500 pricing data from Yahoo! Finance. Notice that the ticker for the S&P 500 is *^GSPC*, https://query1.finance.yahoo.com/v7/finance/download/^GSPC?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

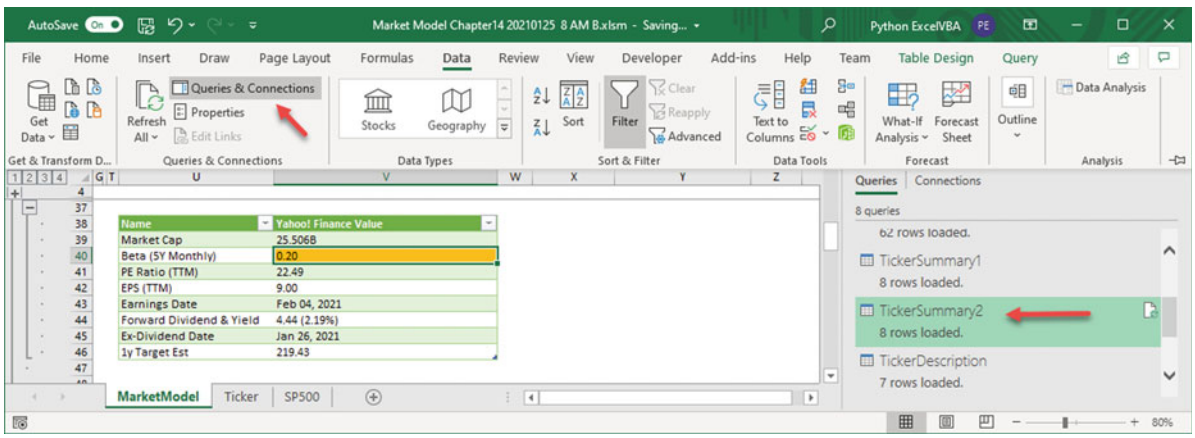
The following is the URL for the 62-month The Clorox Company pricing data from Yahoo! Finance, https://query1.finance.yahoo.com/v7/finance/download/CLX?period1=1446681600&period2=1609804800&interval=1mo&events=history&includeAdjustedClose=true.

The workbook below shows The Clorox Company's market model regression report.

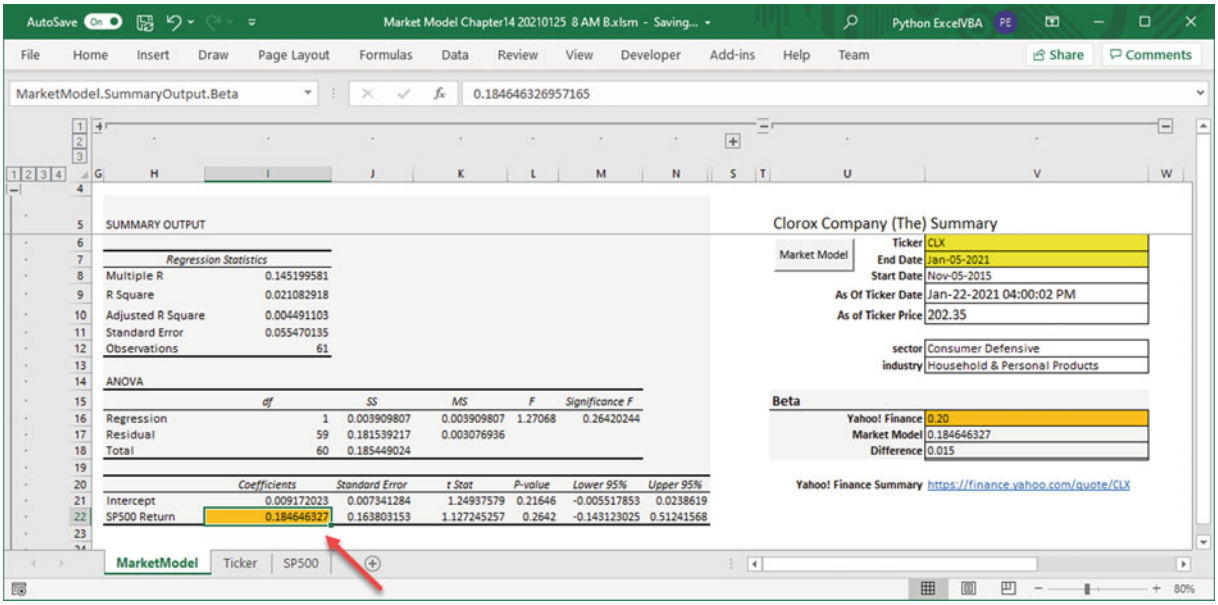


14.12.2 Yahoo! Finance Beta and the Market Model

Cell v40 shows the 0.20 beta value that Yahoo! Finance reported for The Clorox Company.



Cells U38:V36 is the red section shown below from the URL https://finance.yahoo.com/quote/clx. The market model calculated the beta as 0.1846463 for The Clorox Company.

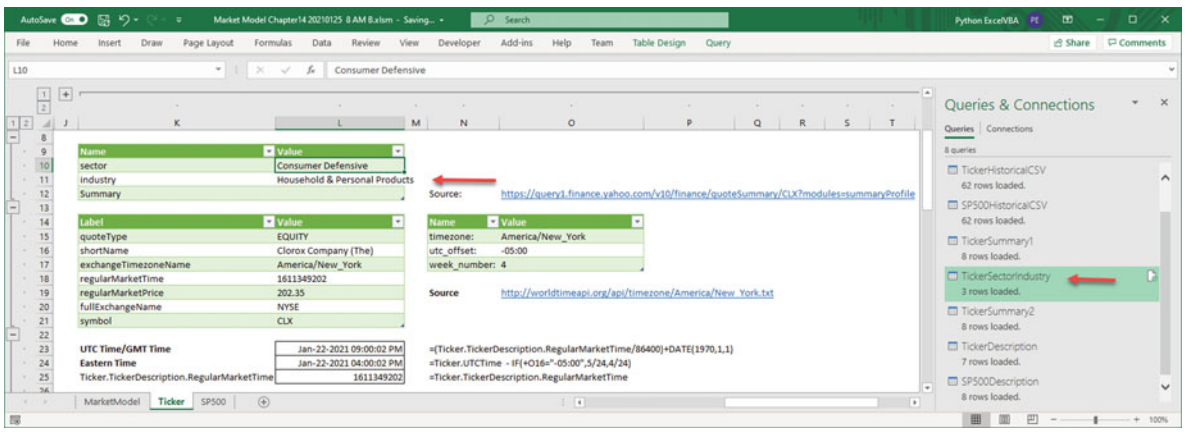


We interpret this to mean that for each 1% increase in the return of the S&P 500, the average increase in The Clorox Corporation's return is 0.18%.

Cell I9 of the regression report indicated that the coefficient of determination was 0.021. This means that 2.1% of the variation of Apple Inc.'s return is explained by the variation in the S&P 500 returns.

14.12.3 Sectors and Industry

Yahoo! Finance reported that The Clorox Company is in the *Consumer Defensive* sector and in the *Household & Personal Products* industry.



The URL https://query1.finance.yahoo.com/v10/finance/quoteSummary/CLX?modules=summaryProfile is the Yahoo! Finance API to retrieve the sector and industry of a ticker.

14.13 Market Model in Python

This section shows the Market Model in Python.

```
[1]  #Date: Apr-04-2021
     import pandas as pd
     import time
     import datetime as dt
```

```
[2]  def Usr_input():
       previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
       Default_end = previousday.strftime("%Y-%m-%d")
       years_ago = dt.date.today()-dt.timedelta(days=(5 * 365) + 60)
       Default_start = years_ago.strftime("%Y-%m-%d")
       Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
       Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or str(Default_start)
       End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or str(Default_end)
       Start = int(time.mktime(time.strptime(Start, "%Y-%m-%d")))
       End = int(time.mktime(time.strptime(End, "%Y-%m-%d")))
       T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'
       return Ticker, Start, End, T_interval
```

```
[3]  query = """https://query1.finance.yahoo.com/v7/finance/download/{stock}?period1={start}&period2={end}&interval={T}&events=history&includeAdjustedClose=true"""
     def calculate_ret(stock_a,stock_M,start,end_t,T_interval):
       Stock_price_a = pd.read_csv(query.format(stock=stock_a, start=start, end=end_t, T=T_interval), index_col=0)['Adj Close'].dropna()
       Ret_a = Stock_price_a.pct_change()[1:]
       Stock_price_M = pd.read_csv(query.format(stock=stock_M, start=start, end=end_t, T=T_interval), index_col=0)['Adj Close'].dropna()
       Ret_M = Stock_price_M.pct_change()[1:]
       Ret = pd.DataFrame([Ret_a,Ret_M])
       Ret = Ret.T
       Ret.columns = [stock_a,'SNP']
       return Ret
```

```
[4]  t, s, e, T = Usr_input()
     returns = calculate_ret(t, '^GSPC', s, e, T)
     market_return = returns['SNP']

     Please enter ticker(Default = JNJ)-->AAPL
     Enter start date(Default = 5 Years ago, 2016-02-05)-->2015-11-05
     Enter end date(Default = Yesterday, 2021-04-03)-->2021-01-05
     Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo
```

```
import statsmodels.api as sm

X = sm.add_constant(market_return)
model = sm.OLS(returns[t], X).fit()
print(model.summary())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecate
  import pandas.util.testing as tm
                            OLS Regression Results
==============================================================================
Dep. Variable:                   AAPL   R-squared:                       0.423
Model:                            OLS   Adj. R-squared:                  0.414
Method:                 Least Squares   F-statistic:                     43.30
Date:                Sun, 04 Apr 2021   Prob (F-statistic):           1.38e-08
Time:                        18:06:59   Log-Likelihood:                 80.223
No. Observations:                  61   AIC:                            -156.4
Df Residuals:                      59   BIC:                            -152.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0179      0.009      2.049      0.045       0.000       0.035
SNP            1.2831      0.195      6.581      0.000       0.893       1.673
==============================================================================
Omnibus:                        8.030   Durbin-Watson:                   2.353
Prob(Omnibus):                  0.018   Jarque-Bera (JB):                7.990
Skew:                          -0.628   Prob(JB):                       0.0184
Kurtosis:                       4.252   Cond. No.                         23.1
==============================================================================
```

```
# ANOVA test

from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

tmp = ols(t+' ~ SNP',returns).fit()
anova_table = anova_lm(tmp)
print(anova_table)
```

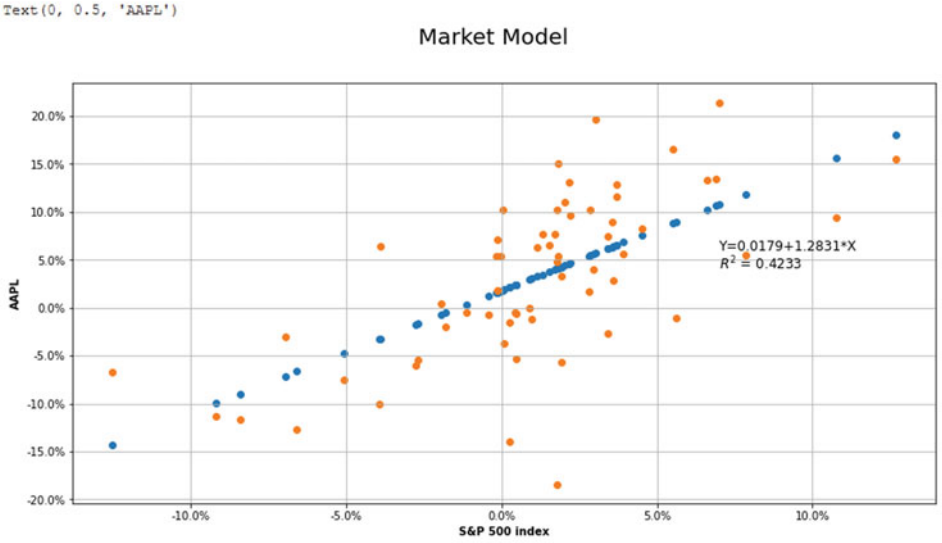
```
              df    sum_sq   mean_sq          F        PR(>F)
SNP          1.0  0.186636  0.186636  42.321983  1.847943e-08
Residual    59.0  0.260185  0.004410        NaN           NaN
```

```
# Plot the result

import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import numpy as np

plt.figure(figsize=(14,7))
plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter())
plt.gca().xaxis.set_major_formatter(mtick.PercentFormatter())
(a,b) = model.params
R_sq = model.rsquared
predict = a+np.multiply(market_return,b)
plt.scatter(market_return*100,predict*100)
plt.scatter(market_return*100,returns[t]*100)
plt.text(7, 6, 'Y={:.4f}+{:.4f}*X'.format(a,b), fontsize=12)
plt.text(7, 4, r'$R^2$ = {:.4f}'.format(R_sq), fontsize=12)
plt.suptitle(' Market model'  ,fontsize=20)
plt.grid(True)
plt.xlabel('S&P 500 index',fontweight='bold')
plt.ylabel(t,fontweight='bold')
```

```
Text(0, 0.5, 'AAPL')
```



```
returns
```

Date	AAPL	SNP
2016-01-01	-0.075242	-0.050735
2016-02-01	-0.006678	-0.004128
2016-03-01	0.133327	0.065991
2016-04-01	-0.139921	0.002699
2016-05-01	0.065287	0.015325
...	...	...
2020-09-01	-0.100908	-0.039228
2020-10-01	-0.060012	-0.027666
2020-11-01	0.093606	0.107546
2020-12-01	0.116497	0.037121
2021-01-01	0.077097	0.026435

61 rows × 2 columns

14.14 Summary

In this chapter, we extended our use of regression analysis. We determined whether the β coefficient would be significantly different from zero or not. We tested this by seeing if the *p*-value calculated for the *b* coefficient was less than the alpha value that we were interested in. We calculated the confidence interval for β. We found the regression line for a given data set. We made predictions using the regression line and found the confidence interval for those predictions.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba." Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language." BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data." O'Reilly Media, 2017.

Multiple Linear Regression

15

15.1 Introduction

In the previous two chapters, we examined the relationship between two variables. In this chapter, we will look at the relationship among three or more variables using regression analysis.

When we looked at the relationship between two variables using regression analysis, one variable explained the other variable of interest. Now we will have two or more variables explaining the variable of interest. The variables that are doing the explaining are called independent variables. The variable that is being explained is called the dependent variable. The mathematical formula for the regressed equation looks like the following:

$$y = a + b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n$$

The x's are the independent variables. The y is the dependent variable. The b's are the coefficients of the independent variables.

A regression that contains more than one independent variable is called a multiple linear regression. We will see that many of the analyses and tests are done in simple linear regression which extend over to multiple linear regression.

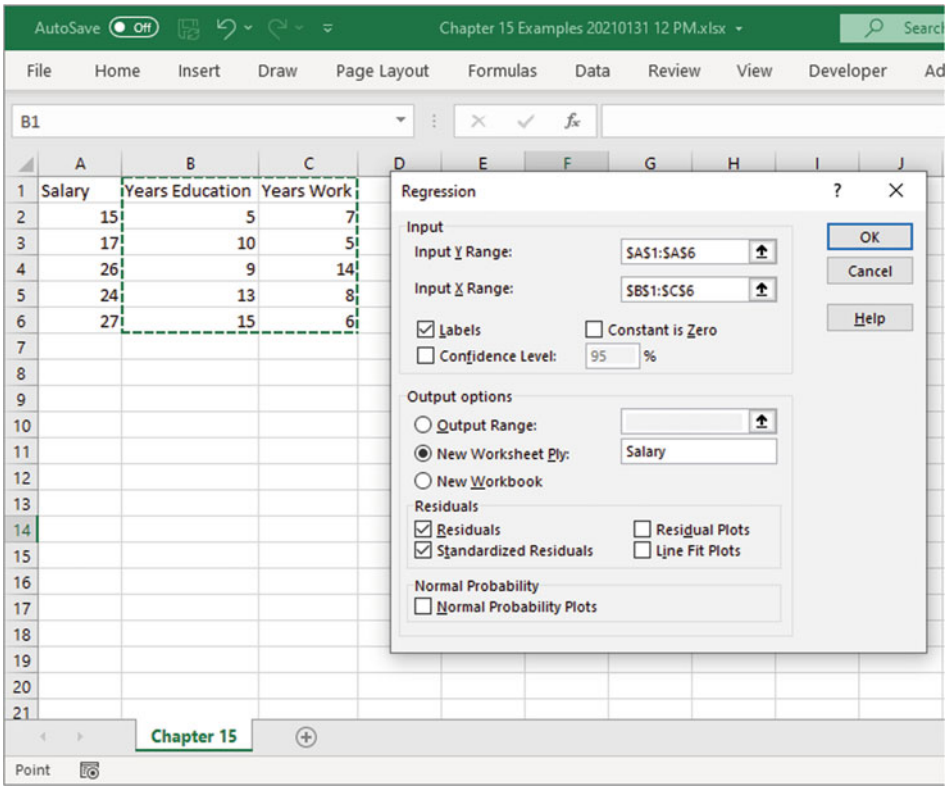
Suppose we believe that years of education and years of work experience can explain the amount of annual salary. To test this, we have the following data:

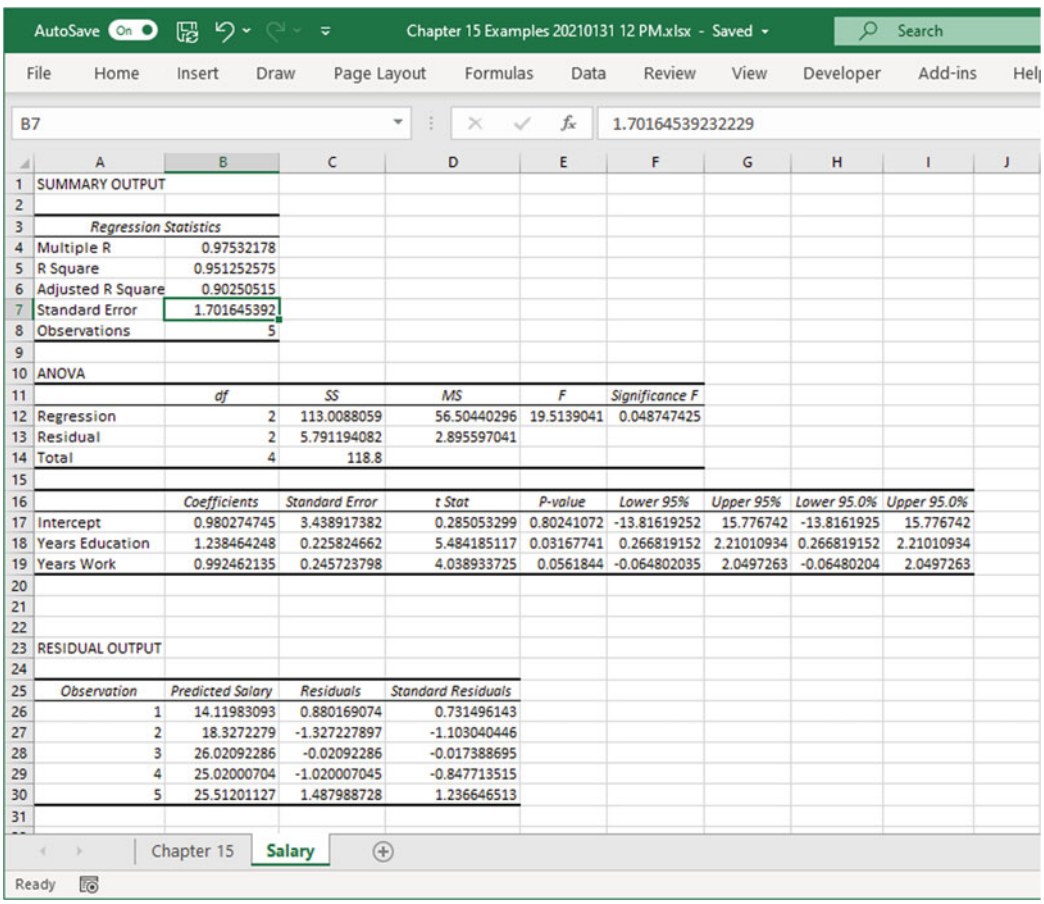
y	x1	x2
Annual salary (thousands)	Years of education	Years of work experience
15	5	7
17	10	5
26	9	14
24	13	8
27	15	6

To test our belief, we will use Excel to do a regression analysis on the above data.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_15


AutoSave Off
Chapter 15 Examples 20210131 12 PM.xlsx
File Home Insert Draw Page Layout Formulas Data Review View Developer
B1
Salary
Years Education
Years Work
15 5 7
17 10 5
26 9 14
24 13 8
27 15 6
Regression
Input
Input Y Range: $A$1:$A$6
Input X Range: $B$1:$C$6
Labels
Constant is Zero
Confidence Level: 95 %
OK
Cancel
Help
Output options
Output Range:
New Worksheet Ply: Salary
New Workbook
Residuals
Residuals
Residual Plots
Standardized Residuals
Line Fit Plots
Normal Probability
Normal Probability Plots
Chapter 15
Point


AutoSave On
Chapter 15 Examples 20210131 12 PM.xlsx - Saved
Search
File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins
B7
1.70164539232229
SUMMARY OUTPUT
Regression Statistics
Multiple R 0.97532178
R Square 0.951252575
Adjusted R Square 0.90250515
Standard Error 1.701645392
Observations 5
ANOVA
df SS MS F Significance F
Regression 2 113.0088059 56.50440296 19.5139041 0.048747425
Residual 2 5.791194082 2.895597041
Total 4 118.8
Coefficients Standard Error t Stat P-value Lower 95% Upper 95% Lower 95.0% Upper 95.0%
Intercept 0.980274745 3.438917382 0.285053299 0.80241072 -13.81619252 15.776742 -13.8161925 15.776742
Years Education 1.238464248 0.225824662 5.484185117 0.03167741 0.266819152 2.21010934 0.266819152 2.21010934
Years Work 0.992462135 0.245723798 4.038933725 0.0561844 -0.064802035 2.0497263 -0.06480204 2.0497263
RESIDUAL OUTPUT
Observation Predicted Salary Residuals Standard Residuals
1 14.11983093 0.880169074 0.731496143
2 18.3272279 -1.327227897 -1.103040446
3 26.02092286 -0.02092286 -0.017388695
4 25.02000704 -1.020007045 -0.847713515
5 25.51201127 1.487988728 1.236646513
Chapter 15
Salary
Ready

This Excel output shows that the regression equation is

```
Salary = .98 + 1.238 Years Education + .992 Years Work
```

This equation, which gives the relationship between annual salary and years of education and years of work experience, suggests that there will be a $1.238 increase in annual salary for every additional year of education. It also suggests that there will be a $0.992 increase in annual salary for every year of additional work experience.

The chapter is broken down into the following sections. Section 15.2 discusses R-square, Sect. 15.3 discusses F-test, and Sect. 15.4 discusses confidence interval of β. In Sect. 15.5 we talk about t-test and in Sect. 15.6 we analyze the determination of price per share. The power query resource issue is discussed in Sect. 15.7. Excel 365 and OneDrive are discussed in Sect. 15.8. In Sect. 15.9, we use R to predict and in Sect. 15.10 we summarize the chapter.

15.2 R-Square

To see how well the independent variables (education and experience) explain the variance of the dependent variable (salary) in the preceding section, we should look at *R*-square. The following part of the Excel output shows that the independent variables explain 95.13% of the dependent variable's variation.

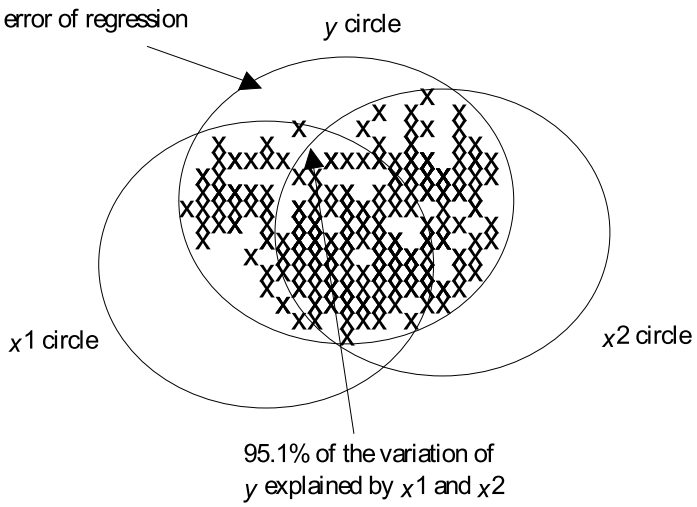
	A	B
1	SUMMARY OUTPUT	
2		
3	*Regression Statistics*	
4	Multiple R	0.97532178
5	R Square	0.951252575
6	Adjusted R Square	0.90250515
7	Standard Error	1.701645392
8	Observations	5

The *R*-square can also be calculated using the following part of the Excel output:

10	ANOVA					
11		*df*	*SS*	*MS*	*F*	*Significance F*
12	Regression	2	113.0088059	56.50440296	19.5139041	0.048747425
13	Residual	2	5.791194082	2.895597041		
14	Total	4	118.8			

$$R{-}square = \frac{113.009}{118.8} = 0.951$$

A Venn diagram of the *R*-square would look like the following:



15.3 *F*-Test

Many times in multiple regression, we use sample data for regression analysis instead of population data. For this reason, we are interested in knowing whether all the true population regression slope coefficients equal zero. To test this, we will use the *F*-test. The *F*-statistic looks like the following:

$$F_{k,n-k-1} = \frac{\frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{k}}{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-k-1}}$$

The hypotheses would be.

$$H_0: \beta_1 = \beta_1 = 0,$$

$$H_1: \text{at least one of } \beta_1 \text{ or } \beta_2 \text{ is not equal to zero,}$$

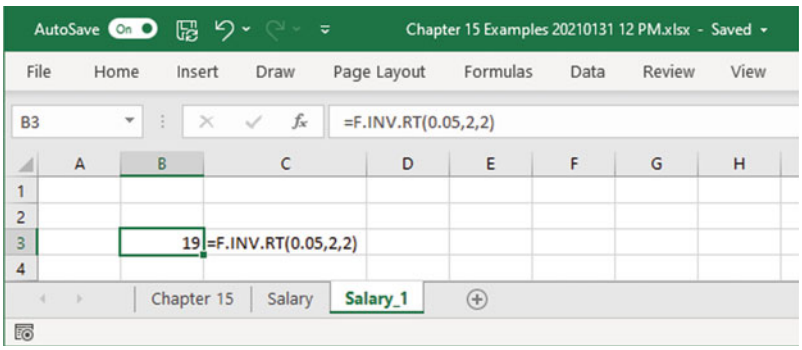
where β_1 and β_2 are population regression parameters.

A reproduction of the regression analysis part (from Sect. 15.1) that we will need for the *F*-test is given below:

10	ANOVA					
11		*df*	*SS*	*MS*	*F*	*Significance F*
12	Regression	2	113.0088059	56.50440296	19.5139041	0.048747425
13	Residual	2	5.791194082	2.895597041		
14	Total	4	118.8			

If we test the hypothesis at the 0.05 alpha level, there are two ways to see if we can reject or accept the null hypothesis.

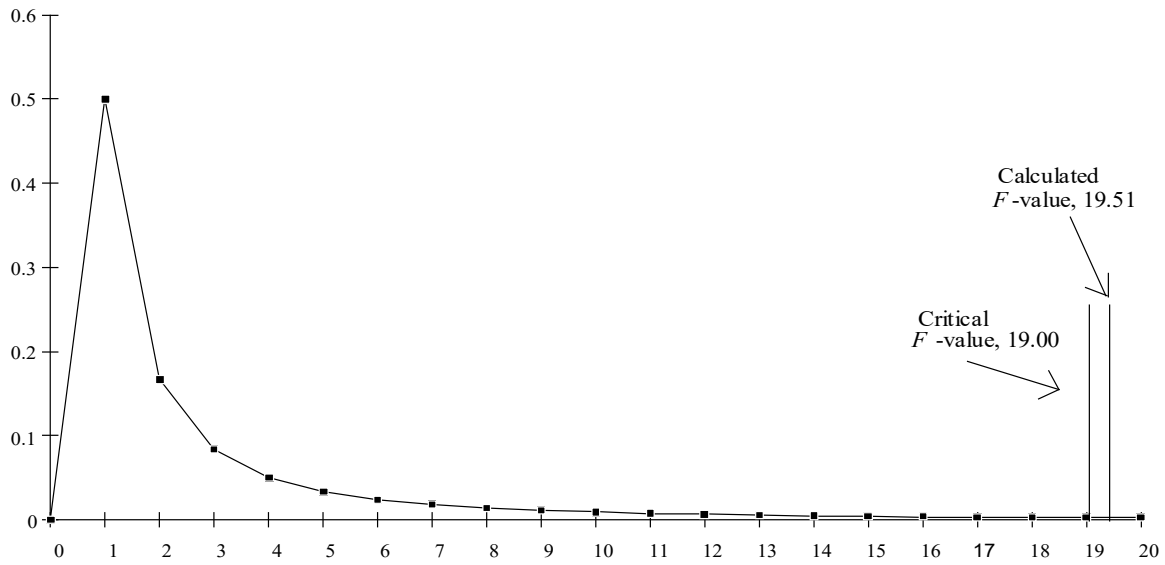
The first way to test the null hypothesis is to find the critical value for a significant level of alpha = 0.05. In Excel, we can use the *F.INV.RT* function to find this value, as illustrated below:



The Excel function shows that the alpha value of 0.05 corresponds with an *F*-value of 19. The first argument, "0.05," is a probability associated with the *F*-cumulative distribution. The second argument, "2," indicates the first degree of freedom for the *F*-test. This degree of freedom is equal to the number of independent variables. The third argument, "2," indicates the second degree of freedom for the *F*-test. This degree of freedom is obtained by taking the number of data items minus the number of independent variables minus one, or mathematically, $n - k - 1$. In this case, we have 5 data items minus 2 independent variables minus 1 equals 2.

We will reject the null hypothesis if the calculated *F*-value is greater than the *F*-value that corresponds with an alpha value of 0.05.

Since the *F*-value that corresponds with an alpha value of 0.05 is 19 and Excel calculated the *F*-value as 19.51, we can conclude that at least one of the regression coefficients is significantly different from zero. This is shown below.



The second way to test the null hypothesis is to find the *p*-value for the calculated *F*-value. In a one-tail test like the *F*-test, the *p*-value is the area to the calculated F-value's right. Excel calculated the *p*-value as 0.049. This means that the area to the right of the calculated *F*-value is 0.049. An alpha value of 0.05 means that there is an *F*-value that has an area to the right of 0.05. We reject the null hypothesis and accept the alternative hypothesis when the *p*-value is less than the alpha value. Since our *p*-value is less than the alpha value of 0.05, we reject the null hypothesis that all the population coefficients are equal to zero.

15.4 Confidence Interval of B

	A	B	C	D	E	F	G	H	I
15									
16		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
17	Intercept	0.980274745	3.438917382	0.285053299	0.80241072	-13.81619252	15.776742	-13.8161925	15.776742
18	Years Education	1.238464248	0.225824662	5.484185117	0.03167741	0.266819152	2.21010934	0.266819152	2.21010934
19	Years Work	0.992462135	0.245723798	4.038933725	0.0561844	-0.064802035	2.0497263	-0.06480204	2.0497263

From the above regression analysis, we can see that the 95% confidence interval for the *Years Education* independent variable constant is $0.2668 \leq \beta \leq 2.2101$.

From the above regression analysis, we can see that the 95% confidence interval for the *Years Work* independent variable constant is $-0.0648 \leq \beta \leq 2.0497$.

15.5 *t*-Test

From the *F*-test, we know that at least one of the two independent variables has a significant effect on the dependent variable. We will now use the *t*-test to see which of the independent variables has a significant effect.

Let us begin by testing if x_1, years of education, has a significant effect on annual salary. The *t*-test statistic is given below:

$$t = \frac{\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2} - 0}{\frac{s_e}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}}} = \frac{b-0}{s_b}$$

The null and alternative hypotheses would be

$$H_0 : \beta_1 = 0$$
$$H_1 : \beta_1 \neq 0$$

A reproduction of the calculated t-value from the regression output in Sect. 15.1 is given below. Excel calculates the t-statistic as a two-tail test.

	A	B	C	D	E	F	G	H	I
15									
16		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
17	Intercept	0.980274745	3.438917382	0.285053299	0.80241072	-13.81619252	15.776742	-13.8161925	15.776742
18	Years Education	1.238464248	0.225824662	5.484185117	0.03167741	0.266819152	2.21010934	0.266819152	2.21010934
19	Years Work	0.992462135	0.245723798	4.038933725	0.0561844	-0.064802035	2.0497263	-0.06480204	2.0497263

From this, we can see that the calculated t-statistic for x_1, or years of education, is 5.48, and the corresponding p-value 0.032. Since the p-value is less than the alpha value of 0.05, we will reject the null hypothesis that x_1 has no effect on the dependent variable. That is, we would reject the null hypothesis and accept the alternative hypothesis.

The hypotheses for analyzing the independent variable x_2, years of work experience, are

$$H_0 : \beta_2 = 0$$
$$H_0 : \beta_2 \neq 0$$

From the Excel output, we can see that the calculated p-value equals 0.056. Since the p-value is greater than the alpha value, we cannot reject the null hypothesis that this independent variable has no significant effect on the dependent variable.

Calculating t-statistics is one area where we can see that using Excel is very easy compared to doing the statistical analysis manually, which would take a very long time.

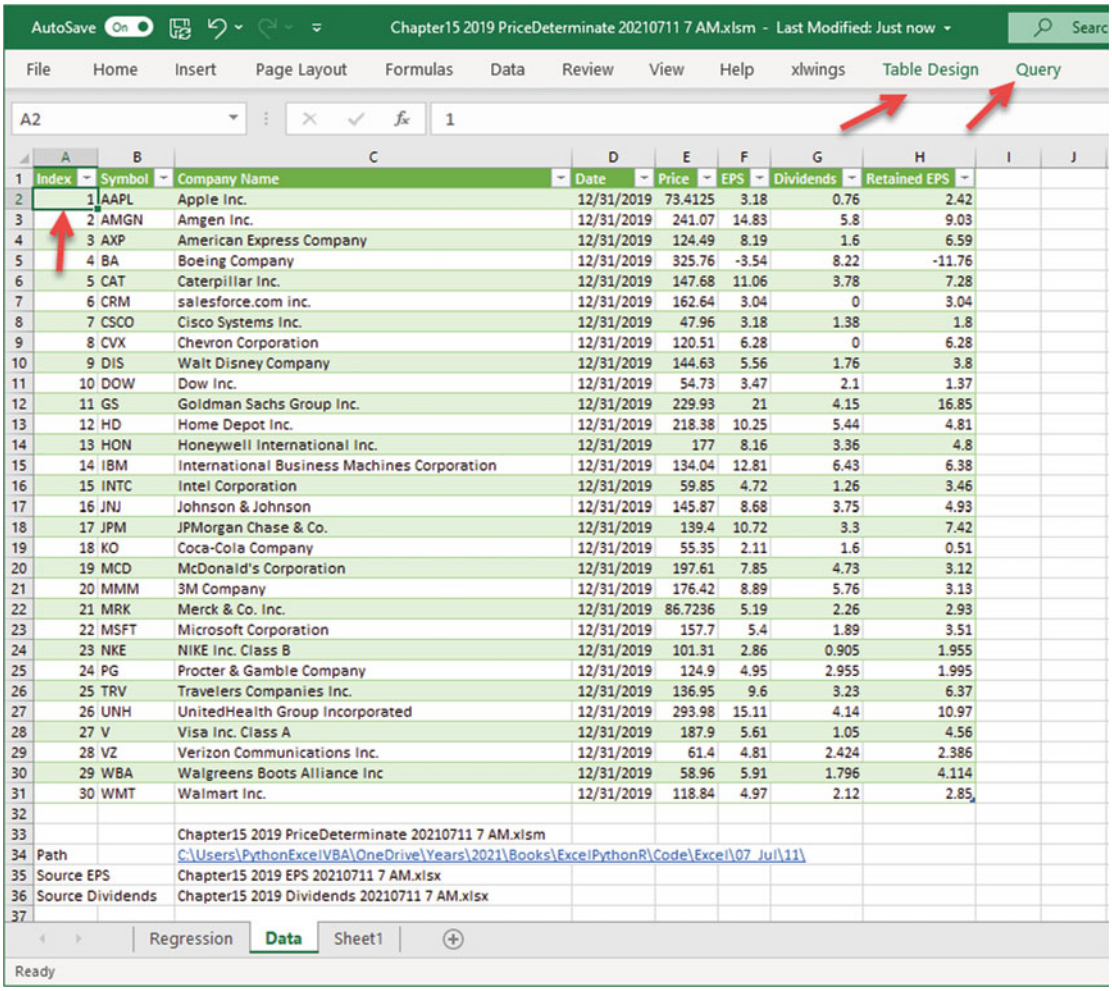
15.6 Analyzing the Determination of Price Per Share

In this section, we will do a *cross-sectional regression*. In a cross-sectional regression, all data come from a single period. The dependent variable in this regression is the price per share (P_j) of the 30 firms used to compile the Dow Jones Industrial Average for the year 2019. The independent variables are the dividend per share (DPS_j) and the retained earnings per share (EPS_j) for the 30 firms. Retained earnings per share are defined as earnings per share minus dividends per share. The price per share is the closing price of the end of the year 2019. The sample regression relationship is

$$P_j = a + b_1 DPS_j + b_2 EPS_j + e_j \quad (j = 1, 2, \ldots, 30)$$

15.6.1 Regression Analysis

The following shows the Dow 30 firm's price, dividends per share, and retained earnings per share for 2019:

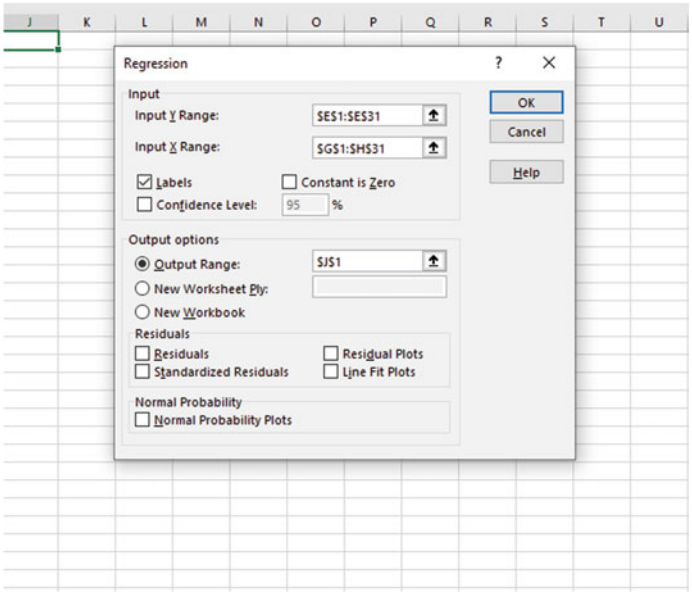


Index	Symbol	Company Name	Date	Price	EPS	Dividends	Retained EPS
1	AAPL	Apple Inc.	12/31/2019	73.4125	3.18	0.76	2.42
2	AMGN	Amgen Inc.	12/31/2019	241.07	14.83	5.8	9.03
3	AXP	American Express Company	12/31/2019	124.49	8.19	1.6	6.59
4	BA	Boeing Company	12/31/2019	325.76	-3.54	8.22	-11.76
5	CAT	Caterpillar Inc.	12/31/2019	147.68	11.06	3.78	7.28
6	CRM	salesforce.com inc.	12/31/2019	162.64	3.04	0	3.04
7	CSCO	Cisco Systems Inc.	12/31/2019	47.96	3.18	1.38	1.8
8	CVX	Chevron Corporation	12/31/2019	120.51	6.28	0	6.28
9	DIS	Walt Disney Company	12/31/2019	144.63	5.56	1.76	3.8
10	DOW	Dow Inc.	12/31/2019	54.73	3.47	2.1	1.37
11	GS	Goldman Sachs Group Inc.	12/31/2019	229.93	21	4.15	16.85
12	HD	Home Depot Inc.	12/31/2019	218.38	10.25	5.44	4.81
13	HON	Honeywell International Inc.	12/31/2019	177	8.16	3.36	4.8
14	IBM	International Business Machines Corporation	12/31/2019	134.04	12.81	6.43	6.38
15	INTC	Intel Corporation	12/31/2019	59.85	4.72	1.26	3.46
16	JNJ	Johnson & Johnson	12/31/2019	145.87	8.68	3.75	4.93
17	JPM	JPMorgan Chase & Co.	12/31/2019	139.4	10.72	3.3	7.42
18	KO	Coca-Cola Company	12/31/2019	55.35	2.11	1.6	0.51
19	MCD	McDonald's Corporation	12/31/2019	197.61	7.85	4.73	3.12
20	MMM	3M Company	12/31/2019	176.42	8.89	5.76	3.13
21	MRK	Merck & Co. Inc.	12/31/2019	86.7236	5.19	2.26	2.93
22	MSFT	Microsoft Corporation	12/31/2019	157.7	5.4	1.89	3.51
23	NKE	NIKE Inc. Class B	12/31/2019	101.31	2.86	0.905	1.955
24	PG	Procter & Gamble Company	12/31/2019	124.9	4.95	2.955	1.995
25	TRV	Travelers Companies Inc.	12/31/2019	136.95	9.6	3.23	6.37
26	UNH	UnitedHealth Group Incorporated	12/31/2019	293.98	15.11	4.14	10.97
27	V	Visa Inc. Class A	12/31/2019	187.9	5.61	1.05	4.56
28	VZ	Verizon Communications Inc.	12/31/2019	61.4	4.81	2.424	2.386
29	WBA	Walgreens Boots Alliance Inc	12/31/2019	58.96	5.91	1.796	4.114
30	WMT	Walmart Inc.	12/31/2019	118.84	4.97	2.12	2.85

	Chapter15 2019 PriceDeterminate 20210711 7 AM.xlsm
Path	C:\Users\PythonExcelVBA\OneDrive\Years\2021\Books\ExcelPythonR\Code\Excel\07_Jul\11\
Source EPS	Chapter15 2019 EPS 20210711 7 AM.xlsx
Source Dividends	Chapter15 2019 Dividends 20210711 7 AM.xlsx

The active cell is A2 which caused the *Table Design* and *Query* menu to show up. These two menus showing up indicate that the table is an Excel table that is the result of a power query.

The following is the regression screen:



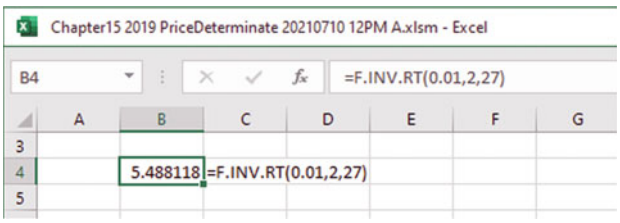
The following is the regression output:

J	K	L	M	N	O	P	Q	R
SUMMARY OUTPUT								
Regression Statistics								
Multiple R	0.691204671							
R Square	0.477763897							
Adjusted R Square	0.439079741							
Standard Error	52.72600997							
Observations	30							
ANOVA								
	df	*SS*	*MS*	*F*	*Significance F*			
Regression	2	68668.88819	34334.4441	12.35037673	0.000155311			
Residual	27	75060.86743	2780.032127					
Total	29	143729.7556						
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	62.35642487	20.22112533	3.083726739	0.004674873	20.86610284	103.8467469	20.86610284	103.8467469
Dividends	24.27697202	4.934680679	4.91966423	3.77953E-05	14.15184362	34.40210043	14.15184362	34.40210043
Retained EPS	2.360470168	2.201852105	1.072038473	0.293191906	-2.157357174	6.87829751	-2.157357174	6.87829751

The constant term is significant with a *t*-value of 3.0837. This result means that the intercept term is statistically different from zero, and the null hypothesis can be rejected at both a 10% and a 5% level. The retained earnings are not significant, with a *t*-value of 1.072 and a *p*-value of 0.2943. Thus, we cannot reject the null hypothesis equals to zero and that it does not contribute to the price per share.

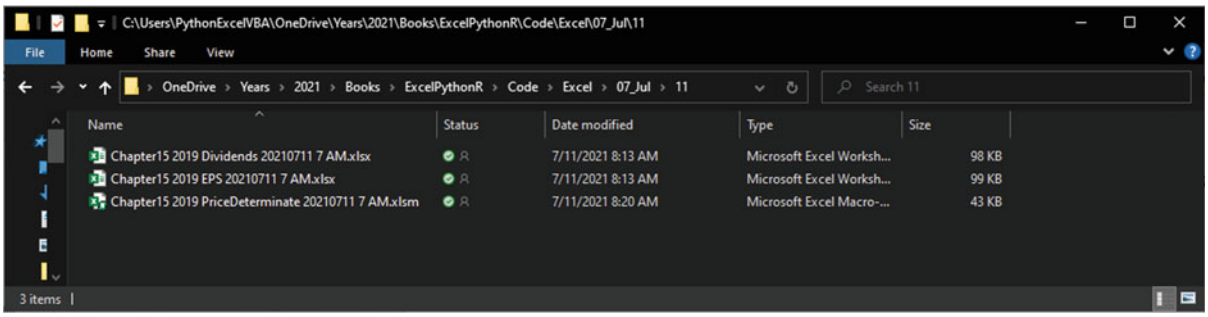
The dividends per share variable's coefficient have a t-value of 4.9196 and a p-value of 0; thus, the null hypothesis is rejected at both the 10% and 5% levels. The coefficient for dividend per share is 24.27. When the dividend increases by $1.00, the price per share tends to rise by $24.27.

The value of R^2 is 0.478, which means that the model explains 47.8% of the observed fluctuations in the price per share. The *F*-value for the regression is 12.35. The number of degrees of freedom for the regression and residuals is 2 and 27, respectively. Using the Excel function *F.INV.RT*, we can see that the critical value for *F* at a 1% level of significance is 5.488.

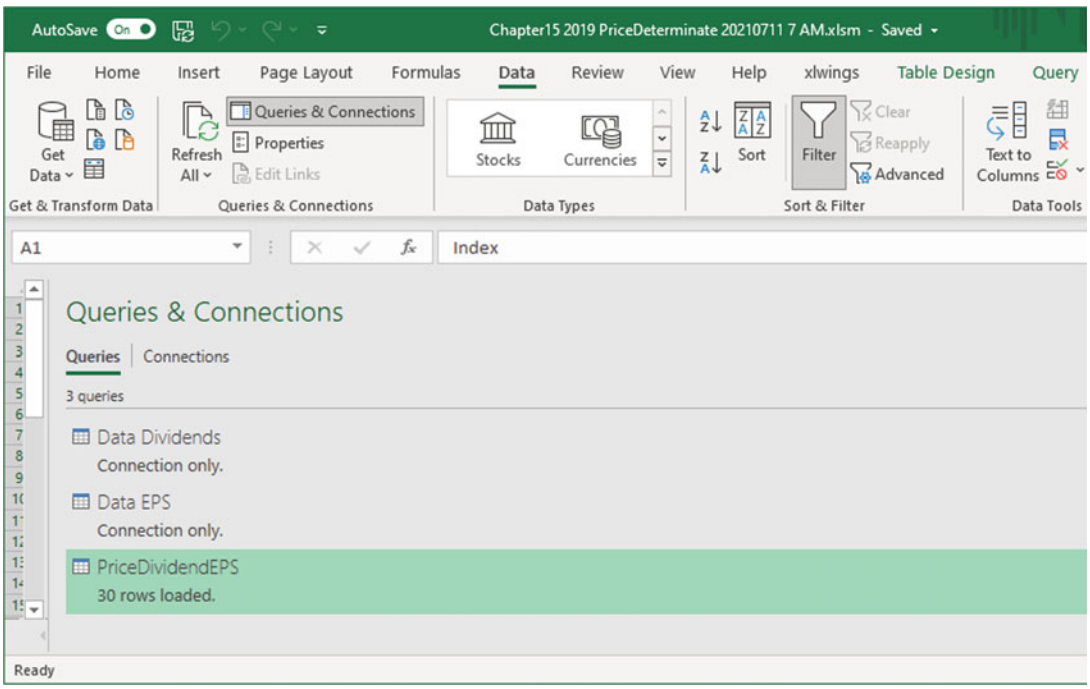


Because the regression *F*-value is greater than the critical value, the null hypothesis that the coefficients are equal to zero is rejected.

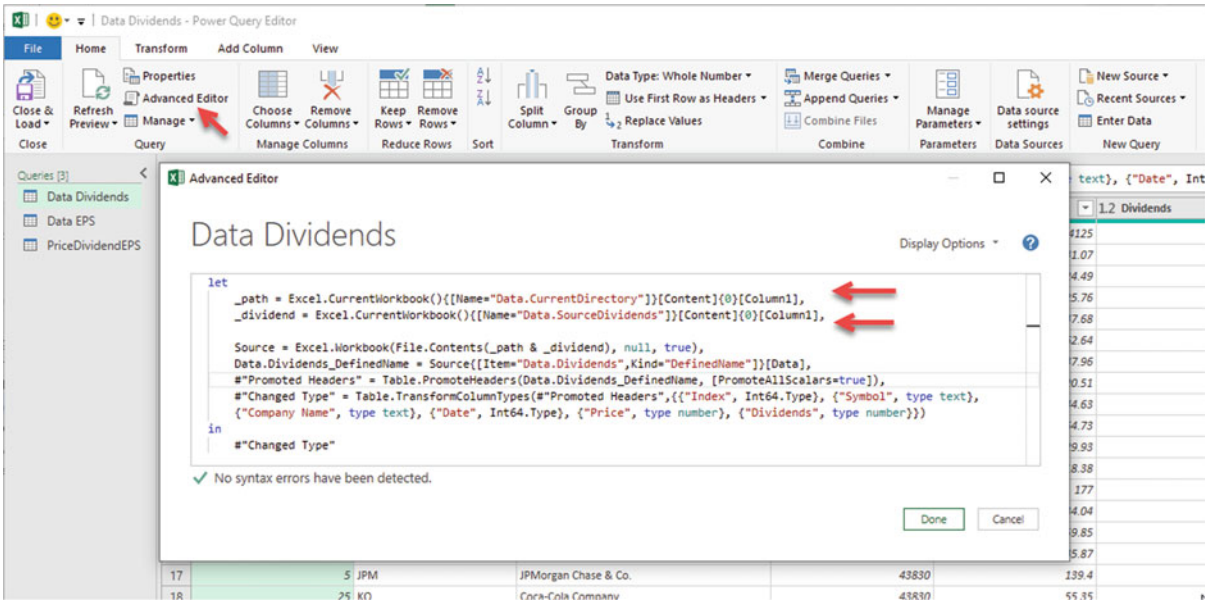
15.6.2 Workbook Sources



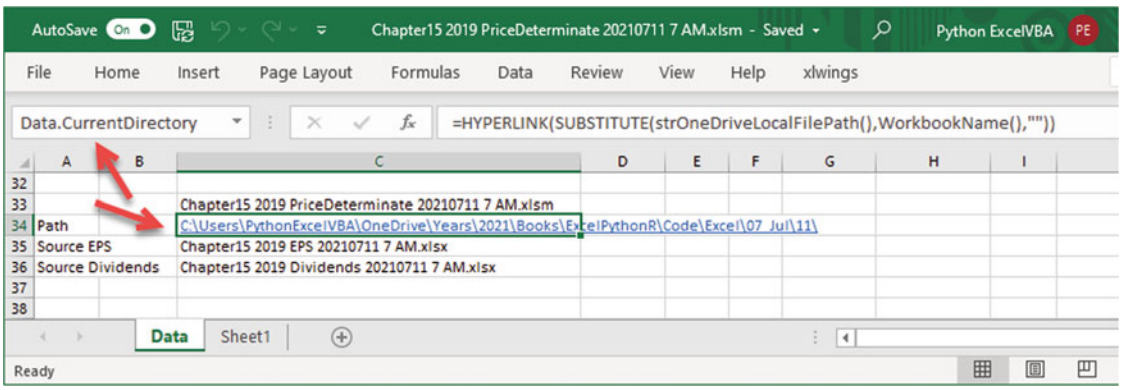
Three workbooks were created and used to produce the price determinate regression. The "PriceDeterminate" workbook is the consolidating workbook. Using power query this workbook pulled dividend information for the dow components from the "Dividends" workbook. Also using power query this workbook pulled eps information for the dow components from the "EPS" workbook.



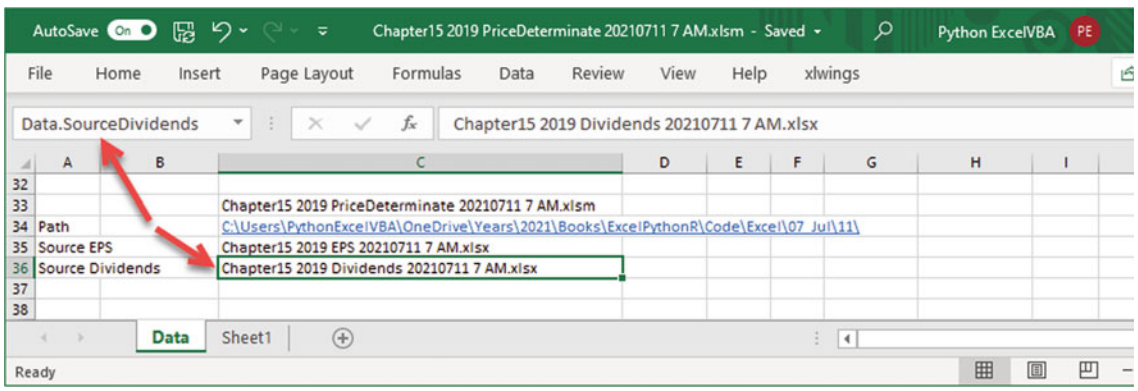
The following is the M Code for the "Data Dividends" power query. The first row reads the Excel range "Data. CurrentDirectory" to find the directory where the dividend workbook is located. The second row reads the Excel range "Data.Sourcedividends" to find the dividend workbook.



The Excel range "Data.CurrentDirectory" in the price determinate workbook is given below:

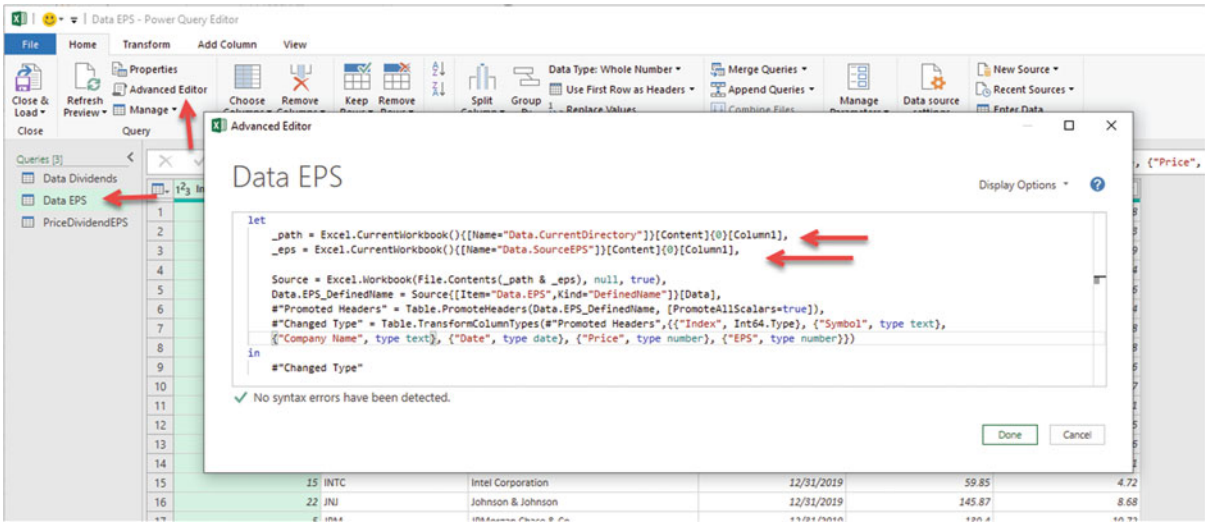


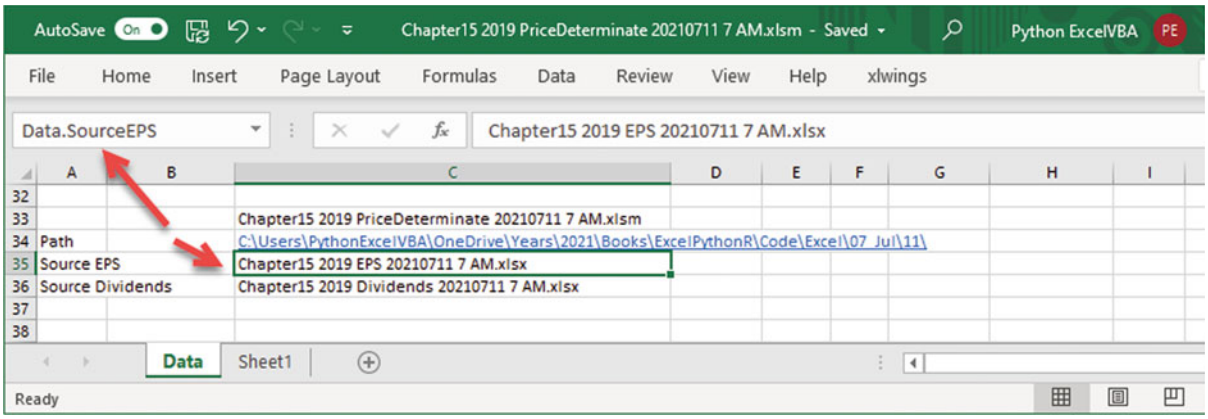
The Excel range "Data.SourceDividends" in the price determinate workbook is given below:



The Excel range "Data.SourceEPS" in the price determinate workbook is given below:

The following is the M Code for the "Data EPS" power query. The first row reads the Excel range "Data.CurrentDirectory" to find the directory where the dividend workbook is located. The second row reads the Excel range "Data.Sourcedividends" to find the dividend workbook.





15.6.3 Data Source

15.6.3.1 Dow 30 Components

The URL https://www.slickcharts.com/dowjones/performance is the source for the Dow 30 components. The Dow 30 components do change over time. The Dow 30 components on July 10, 2021 is shown below:

Dow Jones YTD Returns

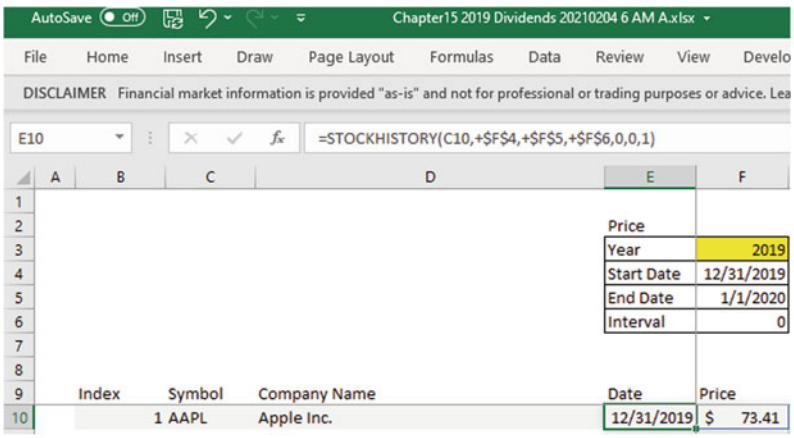
#	Company	Symbol	YTD Return
1	American Express Company	AXP	42.20%
2	Goldman Sachs Group Inc.	GS	40.97%
3	Microsoft Corporation	MSFT	24.96%
4	Chevron Corporation	CVX	23.23%
5	JPMorgan Chase & Co.	JPM	22.59%
6	Home Depot Inc.	HD	21.26%
7	Cisco Systems Inc.	CSCO	20.09%
8	Caterpillar Inc.	CAT	19.45%
9	Walgreens Boots Alliance Inc	WBA	18.88%
10	UnitedHealth Group Incorporated	UNH	17.52%
11	3M Company	MMM	15.00%
12	NIKE Inc. Class B	NKE	13.81%
13	Dow Inc.	DOW	13.66%
14	International Business Machines Corporation	IBM	12.42%
15	Intel Corporation	INTC	12.38%
16	Boeing Company	BA	11.93%
17	salesforce.com inc.	CRM	10.12%
18	McDonald's Corporation	MCD	9.83%
19	Apple Inc.	AAPL	9.36%
20	Travelers Companies Inc.	TRV	9.23%
21	Visa Inc. Class A	V	9.02%
22	Johnson & Johnson	JNJ	7.86%
23	Amgen Inc.	AMGN	6.65%
24	Honeywell International Inc.	HON	5.47%
25	Coca-Cola Company	KO	-0.69%
26	Procter & Gamble Company	PG	-1.52%
27	Walt Disney Company	DIS	-2.29%
28	Walmart Inc.	WMT	-2.67%
29	Verizon Communications Inc.	VZ	-4.56%
30	Merck & Co. Inc.	MRK	-4.66%

Power Query

The following is the power query M code to retrieve the Dow 30 components:

15.6.3.2 Stock Prices

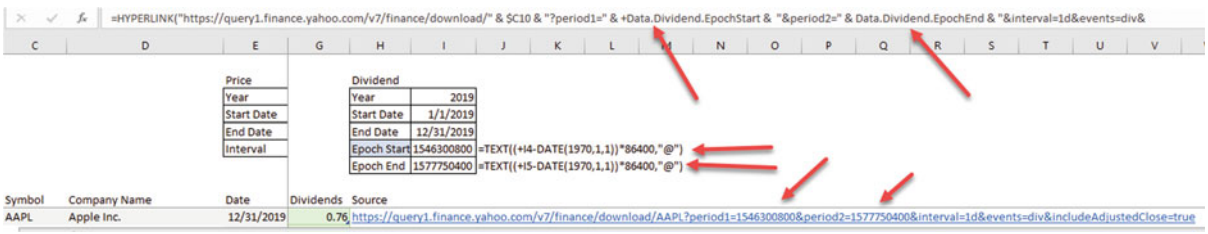
The Excel *STOCKHISTORY* function was used to retrieve the historical prices.



15.6.3.3 Dividends

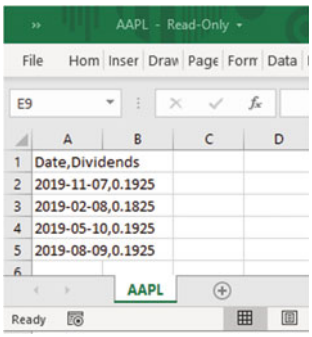
The following URL is the source for AAPL's dividend for 2019:

https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1546300800&period2=1577750400&interval=1d&events=div&includeAdjustedClose=true.



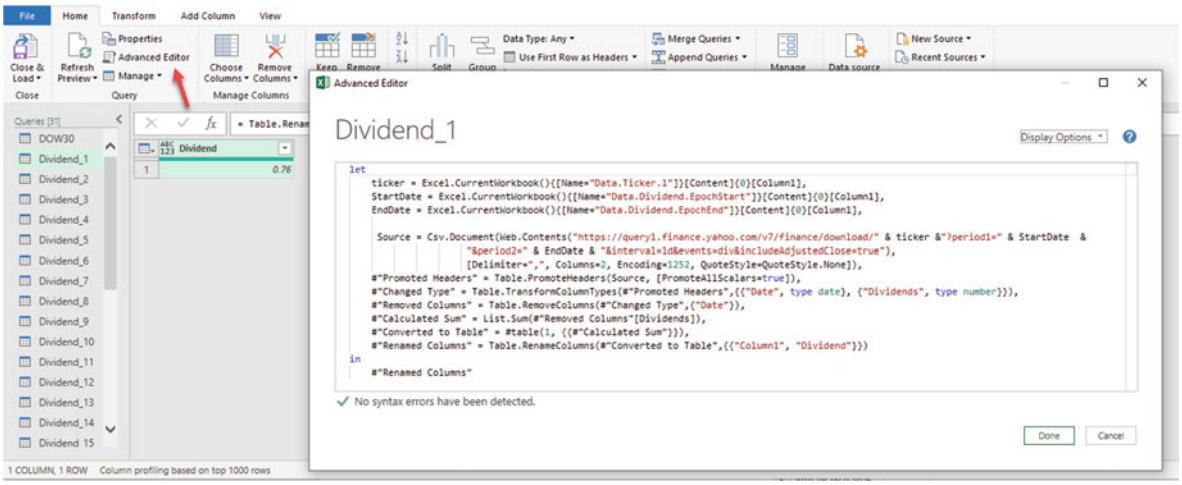
Periods 1 and 2 in the URL use *Epoch Time*. The formula "(+I4-DATE(1970,1,1))*86,400" converts the start date of January 1, 2019 to the Epoch time of 1,546,300,800. The formula "(+I5-DATE(1970,1,1))*86,400" converts the end date of December 31, 2019 to the Epoch time of 1,577,750,400.

Clicking on the URL would show the following four dividend payments for AAPL in 2019. The sum of the four payments equal 0.76.



Power Query

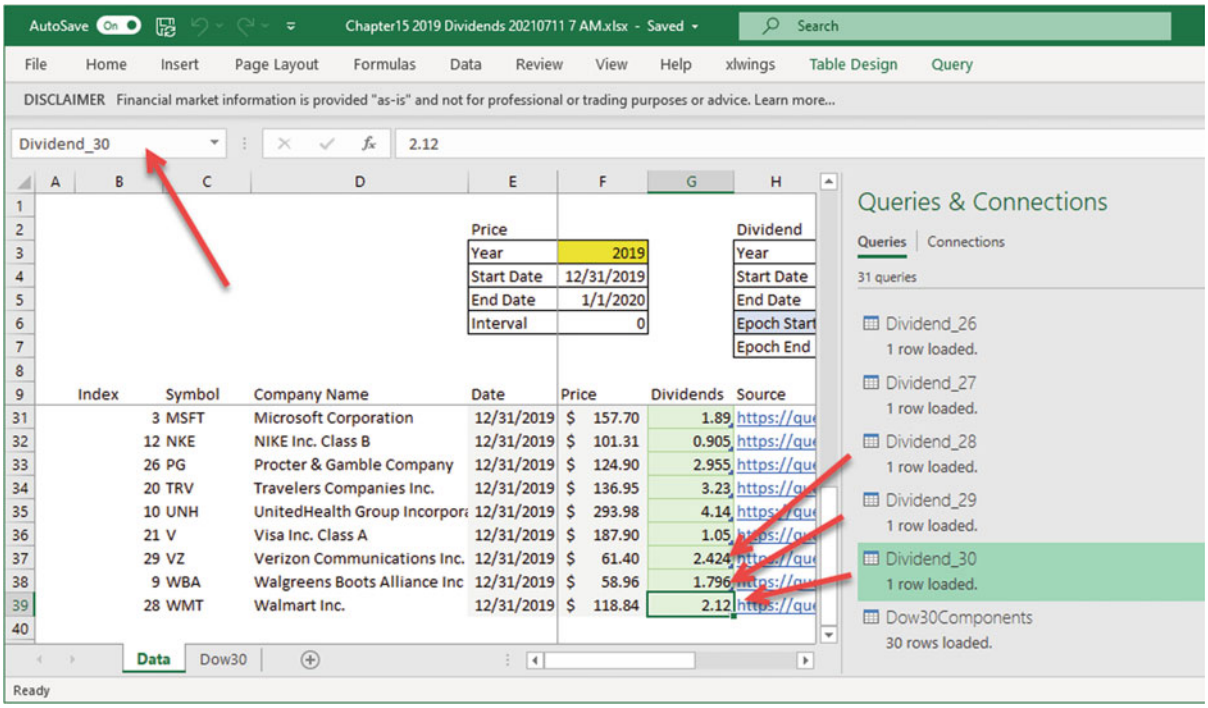
Power Query was used to retrieve the 2019 dividend for AAPL and sum the dividends. The following is the *Power Query* M code to do this.



The above technique was used to retrieve the other 29 dividends in the Dow 30.

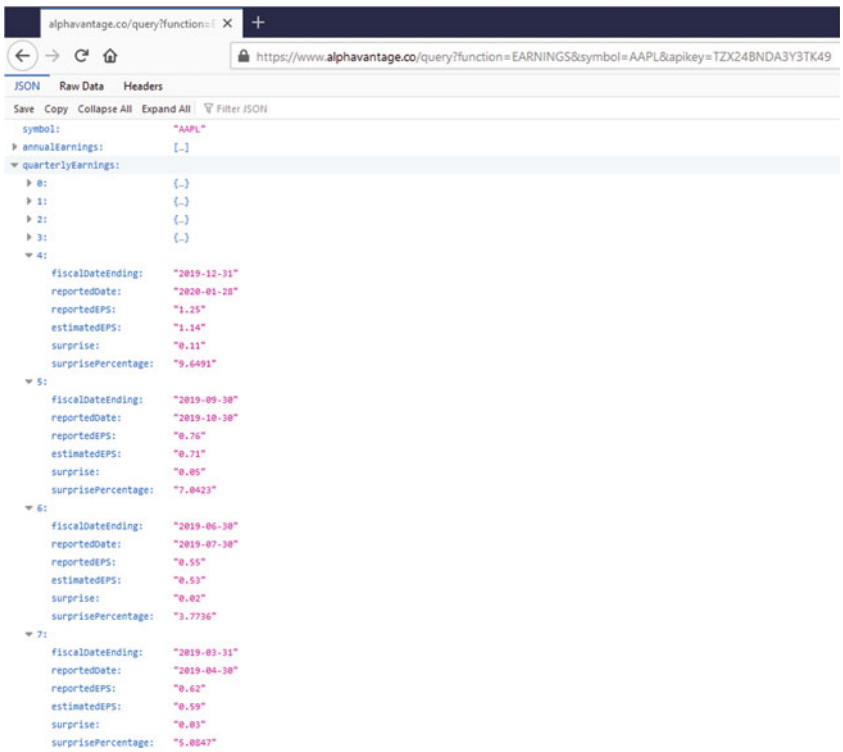
In the dividends workbook, there are 31 power queries. Thirty power were created to retrieve dividend information for each Dow component. One power query was created to retrieve the list of Dow components. The result of the *Dow30Componets* power query is saved in the worksheet *Dow30*.

The result of power query *Dividend_30* is saved in cell G39. The result of power query *Dividend_29* is saved in cell G38. The result of power query *Dividend_28* is saved in cell G37. The pattern for the other dividend power query is the same.



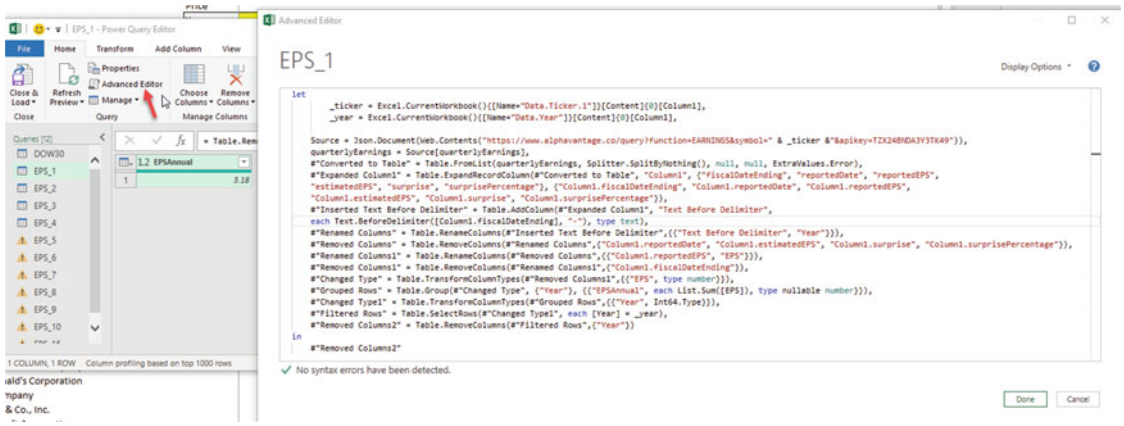
15.6.3.4 Earnings Per Share

The following URL is the source for AAPL's *earnings per share* for 2019:
https://www.alphavantage.co/query?function=EARNINGS&symbol=AAPL&apikey=TZX24BNDA3Y3TK49.
The data source in the Firefox web browser is given below:



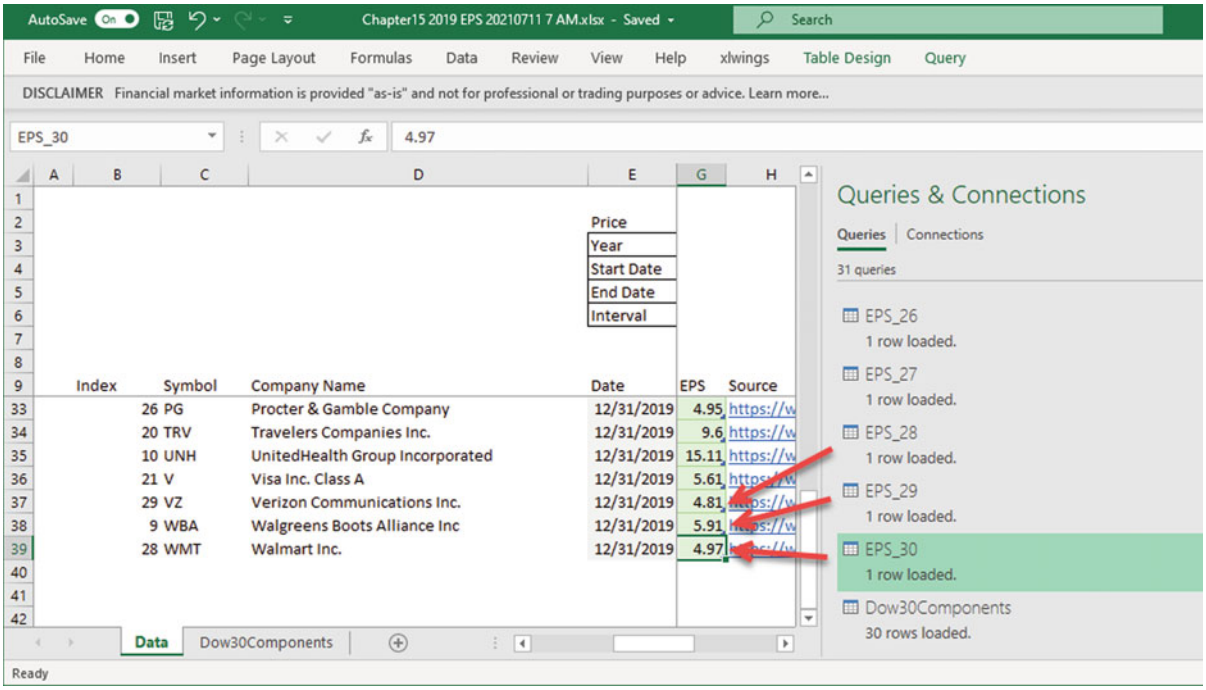
Power Query

Power Query was used to retrieve the earnings per share for AAPL. *Power Query* summed up the quarterly 2019 earnings per share. The following is the *Power Query* M code to do this:



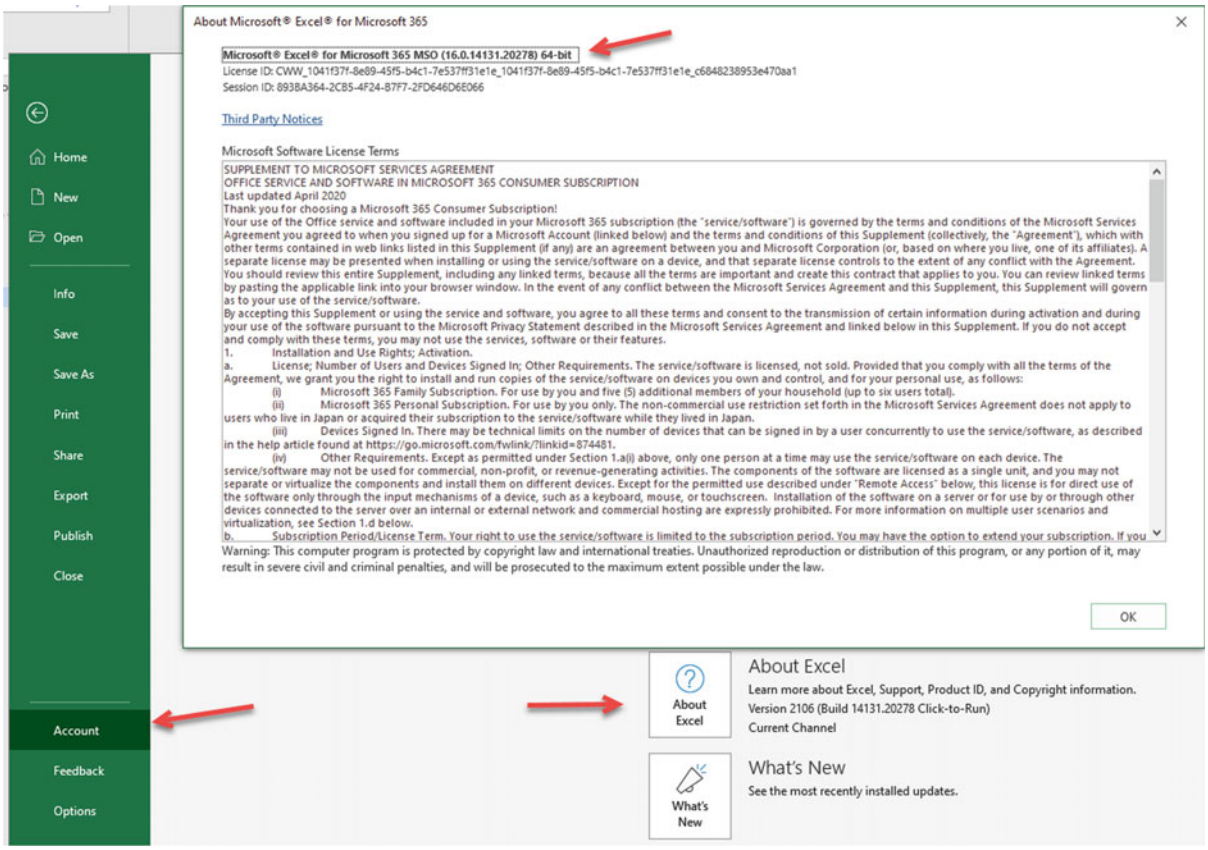
In the earnings per share workbook, there are 31 power queries. Thirty power were created to retrieve earnings per share information for each Dow component. One power query was created to retrieve the list of Dow components. The result of the *Dow30Componets* power query is saved in the worksheet *Dow30*.

The result of power query *EPS_30* is saved in cell G39. The result of power query *EPS_29* is saved in cell G38. The result of power query *PEs_28* is saved in cell G37. The pattern for the other dividend power query is the same.

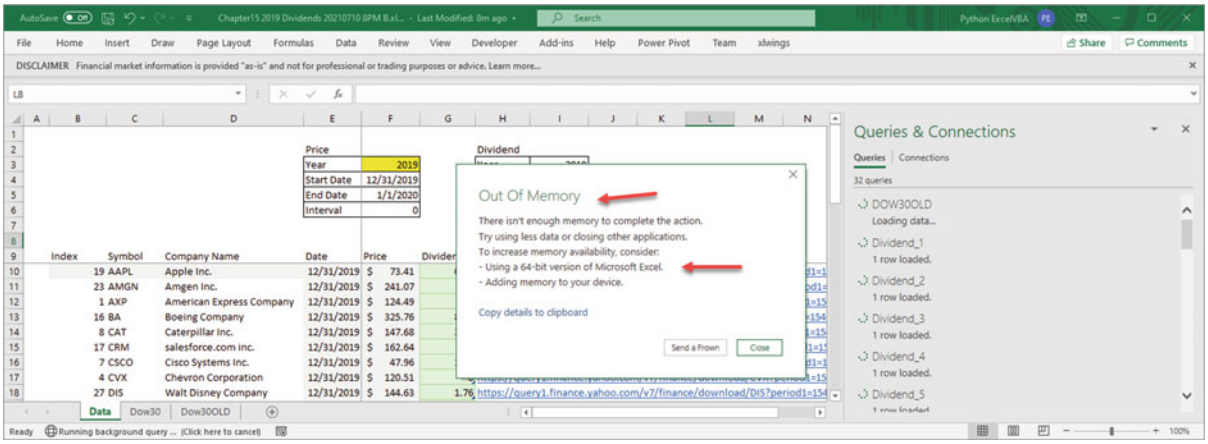


15.7 Power Query Resource Issue

There are two Excel 365 versions. There is an Excel 365 64-bit version and an Excel 365 32-bit version. Do File→Account→About Excel to determine which version that you have.



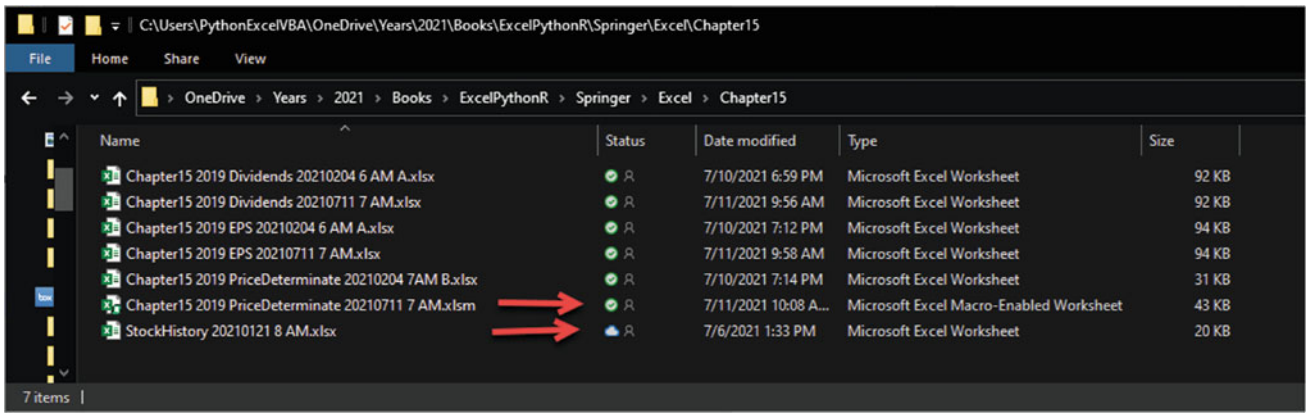
Power query uses a lot of Excel 365 resources. If you have a lot of power queries in a workbook you will need use the Excel 365 64-bit version. If you do not you will get the following *Out of Memory* error and tell you to use Excel 365 64-bit version.



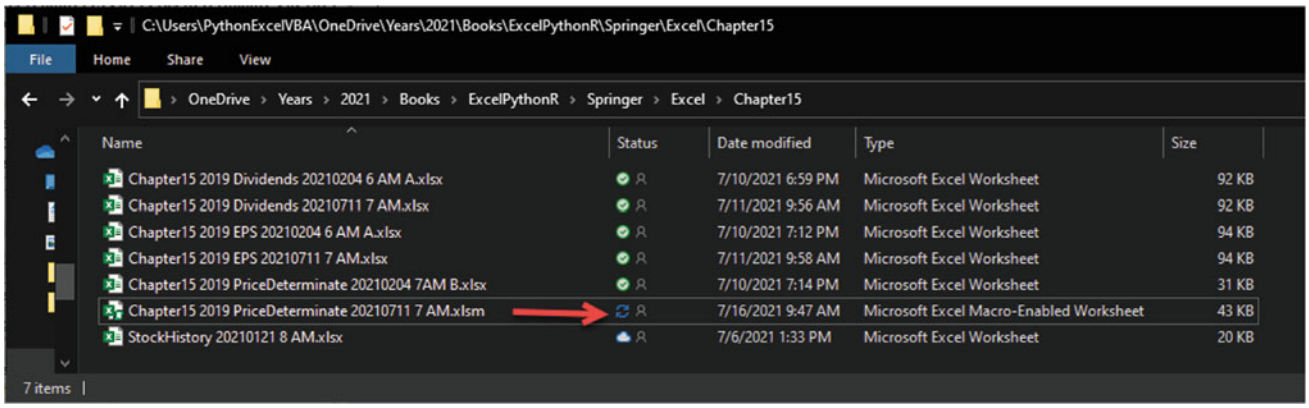
15.8 Excel 365 and OneDrive

With a subscription to Excel 365, you get 1 terabyte of OneDrive. The OneDrive icon will be shown on your Windows task bar.

By default your Excel files will be automatically sync to OneDrive.



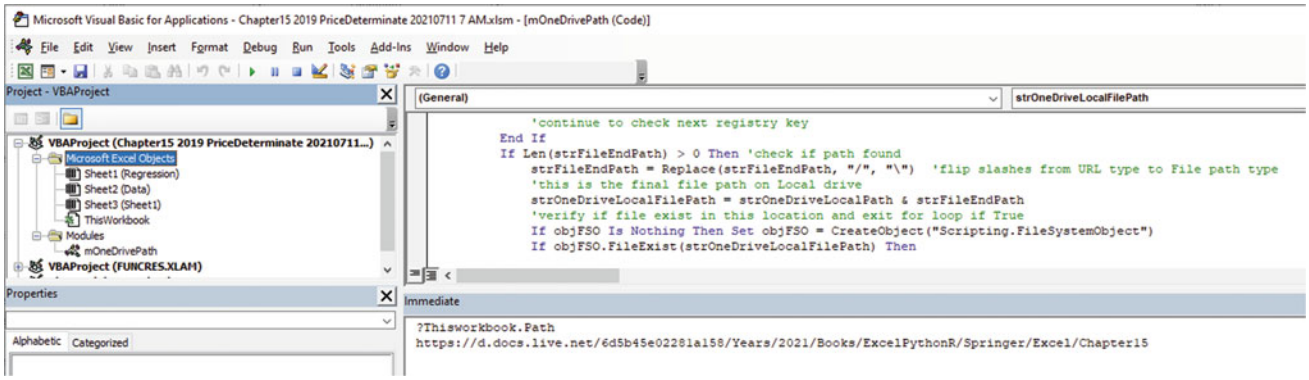
A green check mark icon for the status of a file indicates that the file has been sync and the copy of the file on OneDrive is the same as the copy on your "C" drive. A cloud icon next to the status of a file indicates that a copy of the file is on OneDrive but not on your "C" drive.



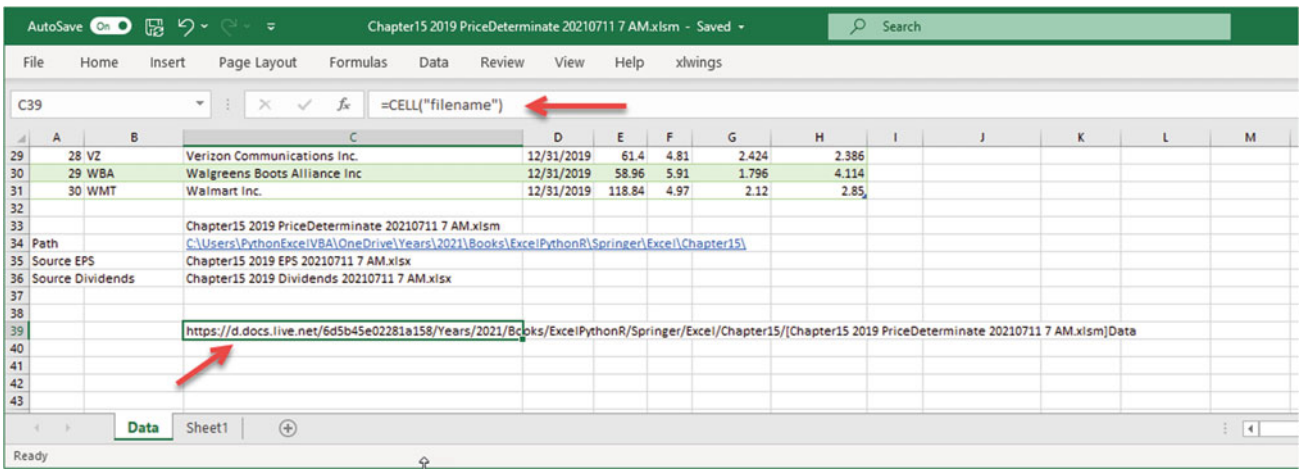
Blue circulating arrow icons for the status of a file indicate that the most current version of the file is on the "C" drive and not on OneDrive.

The "C" drive path for the file "Chapter15 2019 PriceDeterminate 20,210,711 7 AM.xlsm" is "C:\Users\PythonExcelVBA\OneDrive\Years\2021\Books\ExcelPythonR\Springer\Excel\Chapter15."

VBA indicated that the OneDrive path for this file is "https://d.docs.live.net/6d5b45e02281a158/Years/2021/Books/ExcelPythonR/Springer/Excel/Chapter15."



The Excel function *CELL* also shows the OneDrive path instead of the "C" drive path.



Most VBA and Excel functions cannot work with the OneDrive path. The URL https://social.msdn.microsoft.com/Forums/office/en-US/1331519b-1dd1-4aa0-8f4f-0453e1647f57/how-to-get-physical-path-instead-of-url-onedrive created a function that will convert the OnePath path to its equivalent "C" drive path."

```
'https://social.msdn.microsoft.com/Forums/office/en-US/1331519b-1dd1-4aa0-8f4f-0453e1647f57/how-to-get-physical-path-instead-of-url-onedrive
Function strOneDriveLocalFilePath() As String
  Dim ShellScript As Object
  Dim strOneDriveLocalPath As String
  Dim strFileURL As String
  Dim iTryCount As Integer
  Dim strRegKeyName As String
  Dim strFileEndPath As String
  Dim iDocumentsPosition As Integer
  Dim i4thSlashPosition As Integer
  Dim iSlashCount As Integer
  Dim blnFileExist As Boolean
  Dim objFSO As Object
```

```
On Error Resume Next 'invalid or non existin registry keys check would evaluate error

strFileURL = ThisWorkbook.FullName

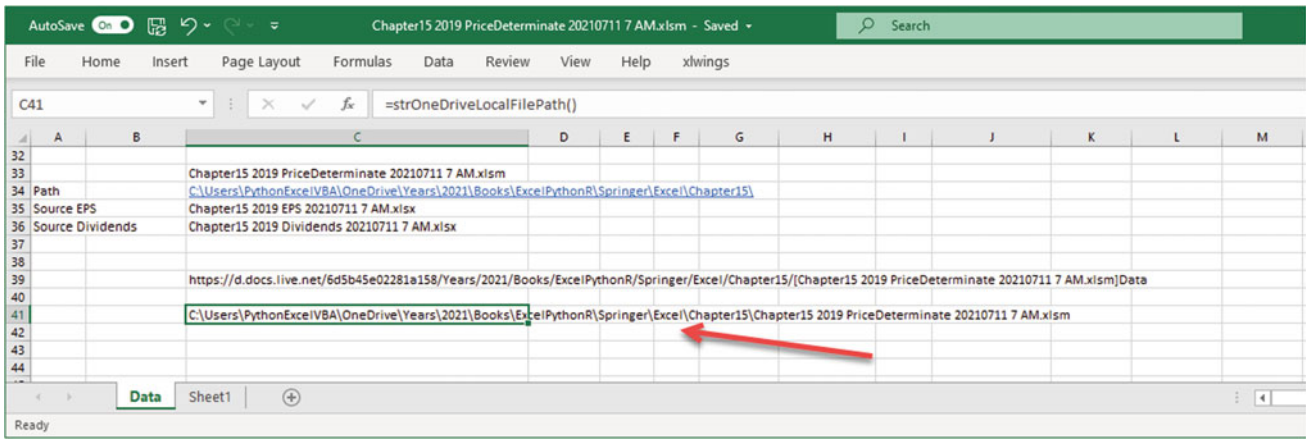
'get OneDrive local path from registry
Set ShellScript = CreateObject(``WScript.Shell'')
'3 possible registry keys to be checked
For iTryCount = 1 To 3
  Select Case (iTryCount)
    Case 1:
      strRegKeyName = ``OneDriveCommercial''
    Case 2:
      strRegKeyName = ``OneDriveConsumer''
    Case 3:
      strRegKeyName = ``OneDrive''
  End Select
  strOneDriveLocalPath = ShellScript.RegRead(``HKEY_CURRENT_USER\Environment\'' & strRegKeyName)
  'check if OneDrive location found
  If strOneDriveLocalPath <> vbNullString Then
    'for commercial OneDrive file path seems to be like
    '``https://companyName-my.sharepoint.com/personal/userName_domain_com/Documents'' & file.FullName)
    If InStr(1, strFileURL, ``my.sharepoint.com'') <> 0 Then
      'find ``/Documents'' in string and replace everything before the end with OneDrive local path
      iDocumentsPosition = InStr(1, strFileURL, ``/Documents'') + Len(``/Documents'')
      'get the ending file path without pointer in OneDrive
      strFileEndPath = Mid(strFileURL, iDocumentsPosition, Len(strFileURL) - iDocumentsPosition + 1)
    Else
      'do nothing
    End If
    'for personal onedrive it looks like ``https://d.docs.live.net/d7bbaa#######1/'' & file.FullName, _
    ' by replacing ``https..'' with OneDrive local path obtained from registry we can get local file path
    If InStr(1, strFileURL, ``d.docs.live.net'') <> 0 Then
      iSlashCount = 1
      i4thSlashPosition = 1
      Do Until iSlashCount > 4
        'loop 4 times, looking for ``''/`` after last found
        i4thSlashPosition = InStr(i4thSlashPosition + 1, strFileURL, ''/``)
        iSlashCount = iSlashCount + 1
      Loop
       'get the ending file path without pointer in OneDrive
      strFileEndPath = Mid(strFileURL, i4thSlashPosition, Len(strFileURL) - i4thSlashPosition + 1)
    Else
      'do nothing
    End If
  Else
    'continue to check next registry key
  End If
  If Len(strFileEndPath) > 0 Then 'check if path found
    strFileEndPath = Replace(strFileEndPath, ''/``, ''\``) 'flip slashes from URL type to File path type
    'this is the final file path on Local drive
    strOneDriveLocalFilePath = strOneDriveLocalPath & strFileEndPath
    'verify if file exist in this location and exit for loop if True
    If objFSO Is Nothing Then Set objFSO = CreateObject(''Scripting.FileSystemObject``)
    If objFSO.FileExist(strOneDriveLocalFilePath) Then
      blnFileExist = True 'that is it - WE GOT IT
```

```
        Exit For 'terminate for loop
      Else
        blnFileExist = False 'not there try another OneDrive type (personal/business)
      End If
    Else
      'continue to check next registry key
    End If
  Next iTryCount
  'display message if file could not be located in any OneDrive folders
  If Not blnFileExist Then MsgBox ''File could not be found in any OneDrive folders``



  'clean up
  Set ShellScript = Nothing
  Set objFSO = Nothing
End Function
```

The *strOneDriveLocalFilePath* custom function is demonstrated below:



15.9 Using *R* to Predict

To show how the multiple regression technique can be used by real estate appraisers, Andres and Ferguson (1986) use the data below to do a multiple regression analysis. They used this formula:

$$y_i = a + b_1 x_{1i} + b_2 x_{2i} + e_i$$

where

y_i = sale price for the ith house,
x_{1i} = home size for the ith house, and.
x_{2i} = condition rating for the ith house.

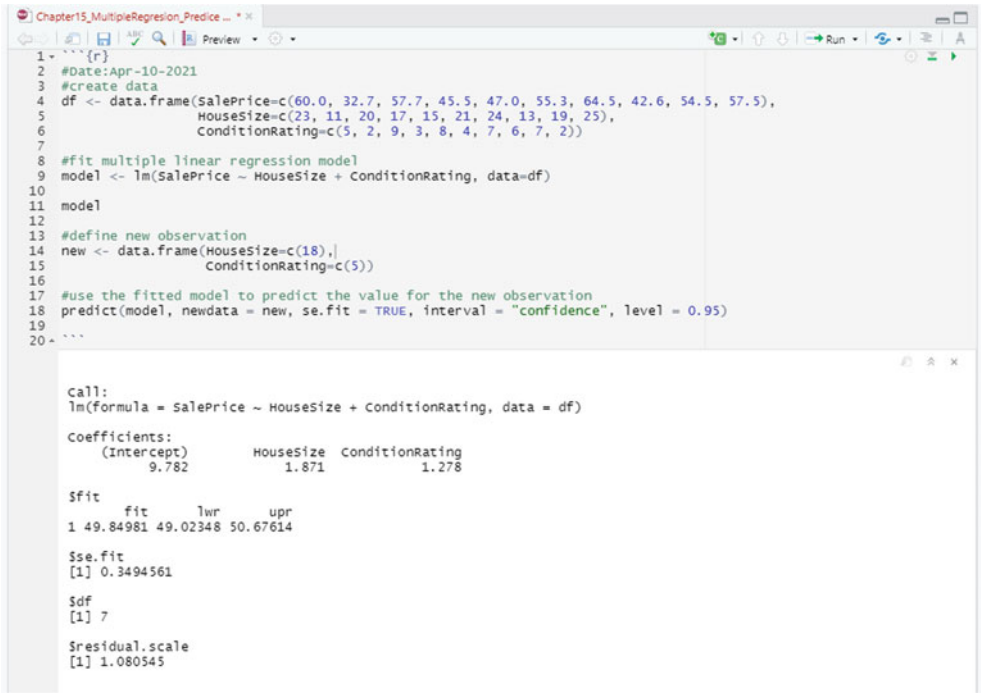
To show how the multiple regression technique can be used by real estate appraisers, Andres and Ferguson (1986) use the data below to do a multiple regression analysis. They used this formula:

$$y_i = a + b_1 x_{1i} + b_2 x_{2i} + e_i$$

where

y_i = sale price for the ith house,
x_{1i} = home size for the ith house, and.
x_{2i} = condition rating for the ith house.

Use *R* to do a regression on the data and predict what the sale price of a house would be if the home size was 1800 square feet and the condition rating was 5.



```r
#Date:Apr-10-2021
#create data
df <- data.frame(SalePrice=c(60.0, 32.7, 57.7, 45.5, 47.0, 55.3, 64.5, 42.6, 54.5, 57.5),
                 HouseSize=c(23, 11, 20, 17, 15, 21, 24, 13, 19, 25),
                 ConditionRating=c(5, 2, 9, 3, 8, 4, 7, 6, 7, 2))

#fit multiple linear regression model
model <- lm(SalePrice ~ HouseSize + ConditionRating, data=df)

model

#define new observation
new <- data.frame(HouseSize=c(18),
                  ConditionRating=c(5))

#use the fitted model to predict the value for the new observation
predict(model, newdata = new, se.fit = TRUE, interval = "confidence", level = 0.95)
```

```
Call:
lm(formula = SalePrice ~ HouseSize + ConditionRating, data = df)

Coefficients:
    (Intercept)        HouseSize  ConditionRating
          9.782            1.871            1.278

$fit
       fit      lwr      upr
1 49.84981 49.02348 50.67614

$se.fit
[1] 0.3494561

$df
[1] 7

$residual.scale
[1] 1.080545
```

From the *R* output, we can see that the regression equation is.

$$\texttt{price} = 9.78 + 1.87\ \texttt{size} + 1.28\ \texttt{rating}$$

This regression equation says that for every 100 square foot increase in size, a house's price will increase by $1,870. For every increase in rating, there will be a $1,290 increase in the price of a house.

From the regression output, we can see that the coefficients for size and rating do not equal zero at the 5% alpha level. We conclude this because the *p*-value calculated for both coefficients is zero, and this would be less than the 5% alpha level.

The regression predicts that the sale price would be $49,850 for a house with 1800 square feet and a rating of 5. The prediction interval for this prediction is from $47,164 to $52,536.

15.10 Summary

In this chapter, we looked at the relationship among three or more variables. We saw that many of the concepts of the relationship between two variables extend to the situations of three or more variables.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.
Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.
Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.
Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.
Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O'Reilly Media, 2017.

16 Residual and Regression Assumption Analysis

16.1 Introduction

When we do regression analysis, there are six underlying assumptions:

1. The dependent and independent variables have a linear relationship.
2. The expected value of the residual term is zero.
3. The variance of the residual term is constant.
4. The residual terms are independent.
5. The independent variables are uncorrelated.
6. The residuals are normally distributed.

To make the proper statistical inferences, these six assumptions must not be materially violated. In this chapter, we will look at the six major assumptions by studying an article that was published in *Accounting Review* (Benston 1966). G. J. Benston did a multiple regression that had the following dependent variable and independent variables:

y = hours of labor in ith week.
x_1 = thousands of pounds shipped.
x_2 = percentage of units shipped by truck.
x_3 = average number of pounds per shipment.

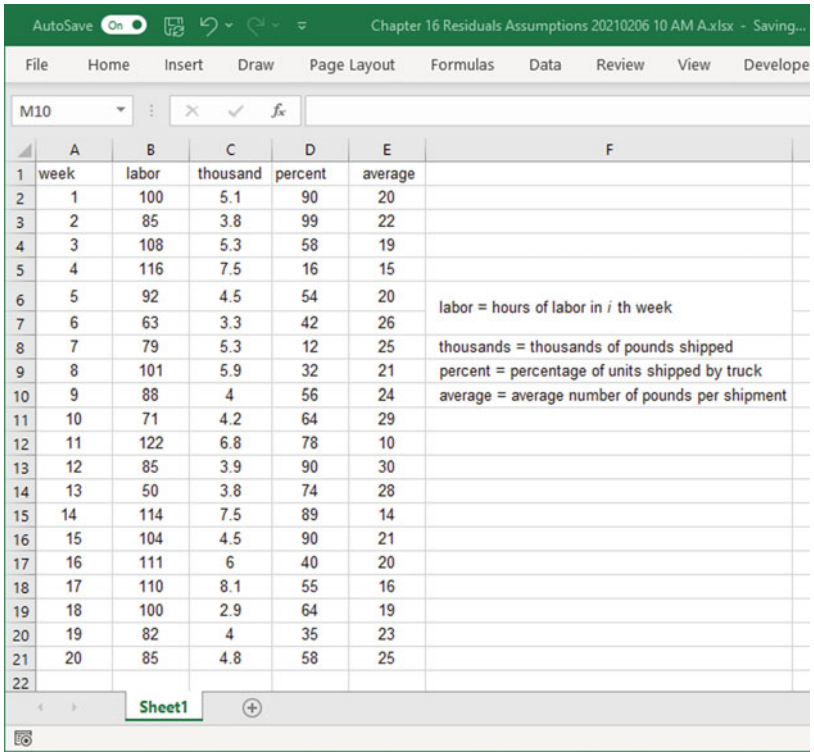
This chapter is broken down into the following sections. Section 16.2 discusses regression analysis, Sect. 16.3 discusses linearity, and Sect. 16.4 explains how the expected value of the residual term is zero. Section 16.5 explains how the variance of the error term is constant, and Sect. 16.6 discusses the autocorrelation Durbin–Watson test. In Sect. 16.7, we look at the autocorrelation of Walmart's dividend and EPS from 2019 to 2000. In Sect. 16.8, we use VBA to retrieve a ticker's name, and in Sect. 16.9, we discuss the Durbin–Watson test market model Python code. Section 16.10 discusses how independent variables are uncorrelated with multicollinearity, and Sect. 16.11 looks at the variance inflationary factor. Finally, in Sect. 16.12, we summarize the chapter.

16.2 Regression Analysis

From the above, we can conclude that Benston felt that thousands of pounds shipped, percentage of units shipped by truck, and the average number of pounds per shipment could explain the hours of labor in a particular week. The data for his model are displayed below.

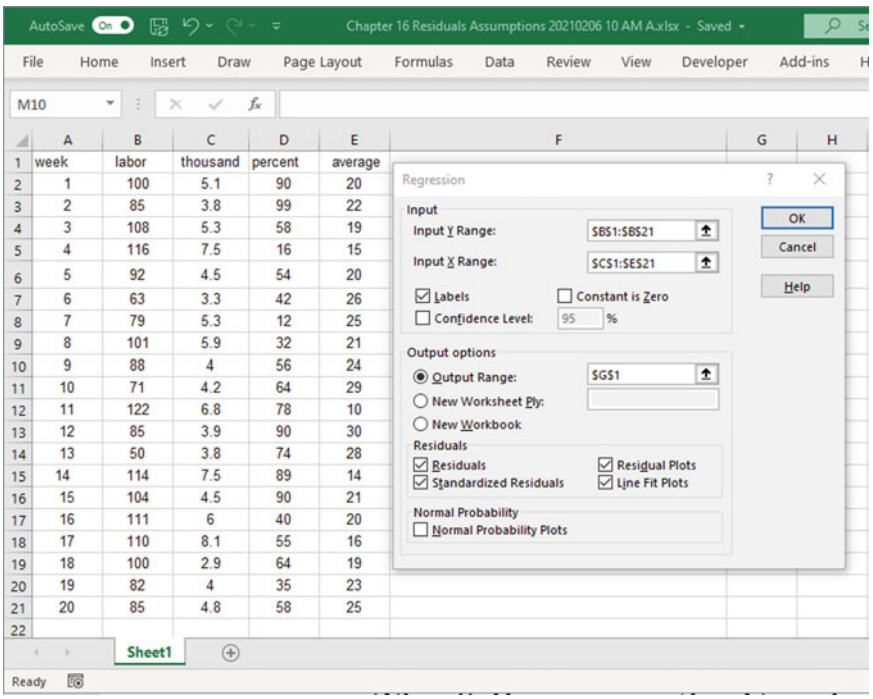
© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_16



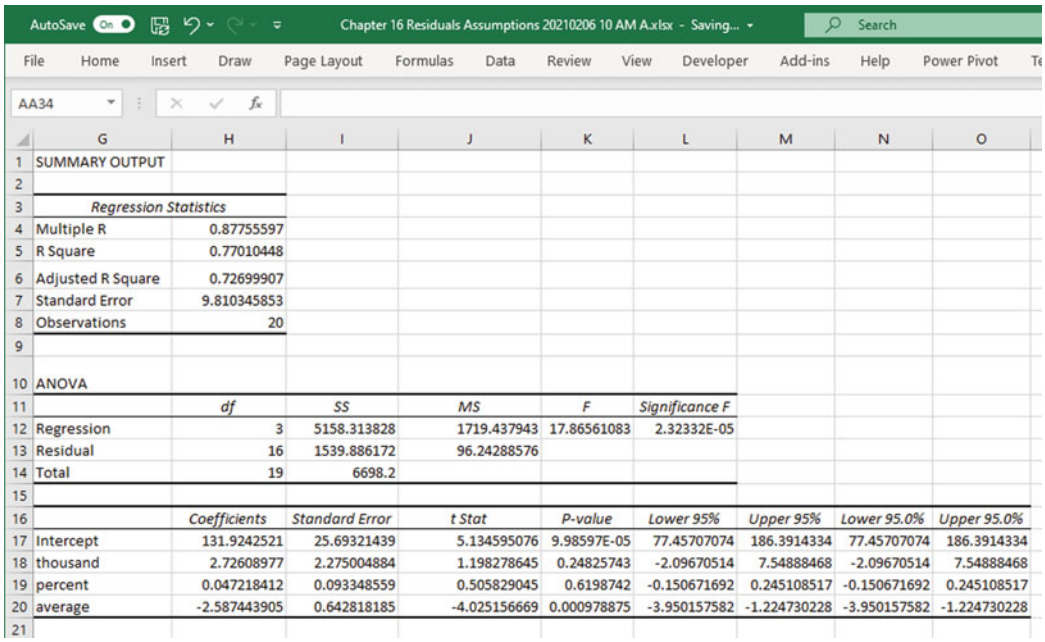
	A	B	C	D	E	F
1	week	labor	thousand	percent	average	
2	1	100	5.1	90	20	
3	2	85	3.8	99	22	
4	3	108	5.3	58	19	
5	4	116	7.5	16	15	
6	5	92	4.5	54	20	labor = hours of labor in *i* th week
7	6	63	3.3	42	26	
8	7	79	5.3	12	25	thousands = thousands of pounds shipped
9	8	101	5.9	32	21	percent = percentage of units shipped by truck
10	9	88	4	56	24	average = average number of pounds per shipment
11	10	71	4.2	64	29	
12	11	122	6.8	78	10	
13	12	85	3.9	90	30	
14	13	50	3.8	74	28	
15	14	114	7.5	89	14	
16	15	104	4.5	90	21	
17	16	111	6	40	20	
18	17	110	8.1	55	16	
19	18	100	2.9	64	19	
20	19	82	4	35	23	
21	20	85	4.8	58	25	
22						

The following is the Excel regression screen to perform the regression analysis on the data.

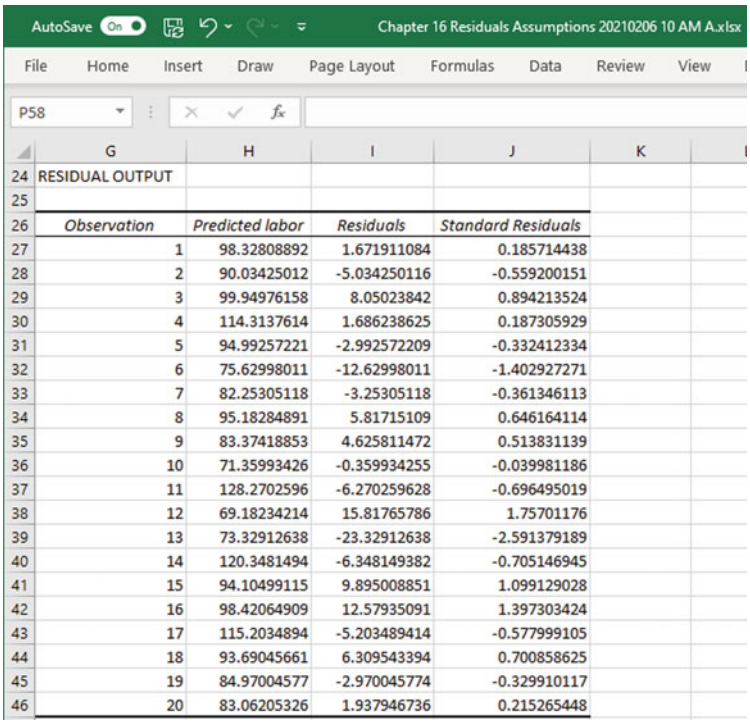


	A	B	C	D	E
1	week	labor	thousand	percent	average
2	1	100	5.1	90	20
3	2	85	3.8	99	22
4	3	108	5.3	58	19
5	4	116	7.5	16	15
6	5	92	4.5	54	20
7	6	63	3.3	42	26
8	7	79	5.3	12	25
9	8	101	5.9	32	21
10	9	88	4	56	24
11	10	71	4.2	64	29
12	11	122	6.8	78	10
13	12	85	3.9	90	30
14	13	50	3.8	74	28
15	14	114	7.5	89	14
16	15	104	4.5	90	21
17	16	111	6	40	20
18	17	110	8.1	55	16
19	18	100	2.9	64	19
20	19	82	4	35	23
21	20	85	4.8	58	25
22					

Below shows the regression report.



	G	H	I	J	K	L	M	N	O
1	SUMMARY OUTPUT								
2									
3	*Regression Statistics*								
4	Multiple R	0.87755597							
5	R Square	0.77010448							
6	Adjusted R Square	0.72699907							
7	Standard Error	9.810345853							
8	Observations	20							
9									
10	ANOVA								
11		*df*	*SS*	*MS*	*F*	*Significance F*			
12	Regression	3	5158.313828	1719.437943	17.86561083	2.32332E-05			
13	Residual	16	1539.886172	96.24288576					
14	Total	19	6698.2						
15									
16		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
17	Intercept	131.9242521	25.69321439	5.134595076	9.98597E-05	77.45707074	186.3914334	77.45707074	186.3914334
18	thousand	2.72608977	2.275004884	1.198278645	0.24825743	-2.09670514	7.54888468	-2.09670514	7.54888468
19	percent	0.047218412	0.093348559	0.505829045	0.6198742	-0.150671692	0.245108517	-0.150671692	0.245108517
20	average	-2.587443905	0.642818185	-4.025156669	0.000978875	-3.950157582	-1.224730228	-3.950157582	-1.224730228
21									



	G	H	I	J	K
24	RESIDUAL OUTPUT				
25					
26	*Observation*	*Predicted labor*	*Residuals*	*Standard Residuals*	
27	1	98.32808892	1.671911084	0.185714438	
28	2	90.03425012	-5.034250116	-0.559200151	
29	3	99.94976158	8.05023842	0.894213524	
30	4	114.3137614	1.686238625	0.187305929	
31	5	94.99257221	-2.992572209	-0.332412334	
32	6	75.62998011	-12.62998011	-1.402927271	
33	7	82.25305118	-3.25305118	-0.361346113	
34	8	95.18284891	5.81715109	0.646164114	
35	9	83.37418853	4.625811472	0.513831139	
36	10	71.35993426	-0.359934255	-0.039981186	
37	11	128.2702596	-6.270259628	-0.696495019	
38	12	69.18234214	15.81765786	1.75701176	
39	13	73.32912638	-23.32912638	-2.591379189	
40	14	120.3481494	-6.348149382	-0.705146945	
41	15	94.10499115	9.895008851	1.099129028	
42	16	98.42064909	12.57935091	1.397303424	
43	17	115.2034894	-5.203489414	-0.577999105	
44	18	93.69045661	6.309543394	0.700858625	
45	19	84.97004577	-2.970045774	-0.329910117	
46	20	83.06205326	1.937946736	0.215265448	

From the regression output, we can see that the regressed equation is

$$\text{labor} = 131.9 + 2.73 \text{ thousand} + 0.0472\% - 2.587 \text{ average}$$

This equation tells us that:

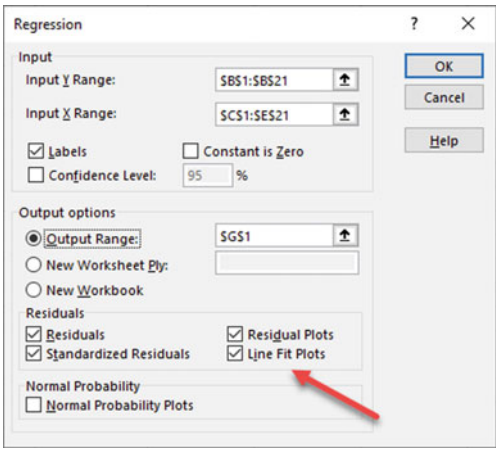
1. For every thousand-pound increase, there will be 2.73 h increase in labor.
2. For every 1% increase in units shipped, there will be 0.0472 h increase in labor.
3. For every average pound increase per shipment, there will be 2.59 h decrease in labor.

From the R-square calculation of 72.7%, we can see that the independent variables explain 72.7% of the variability of the dependent variable.

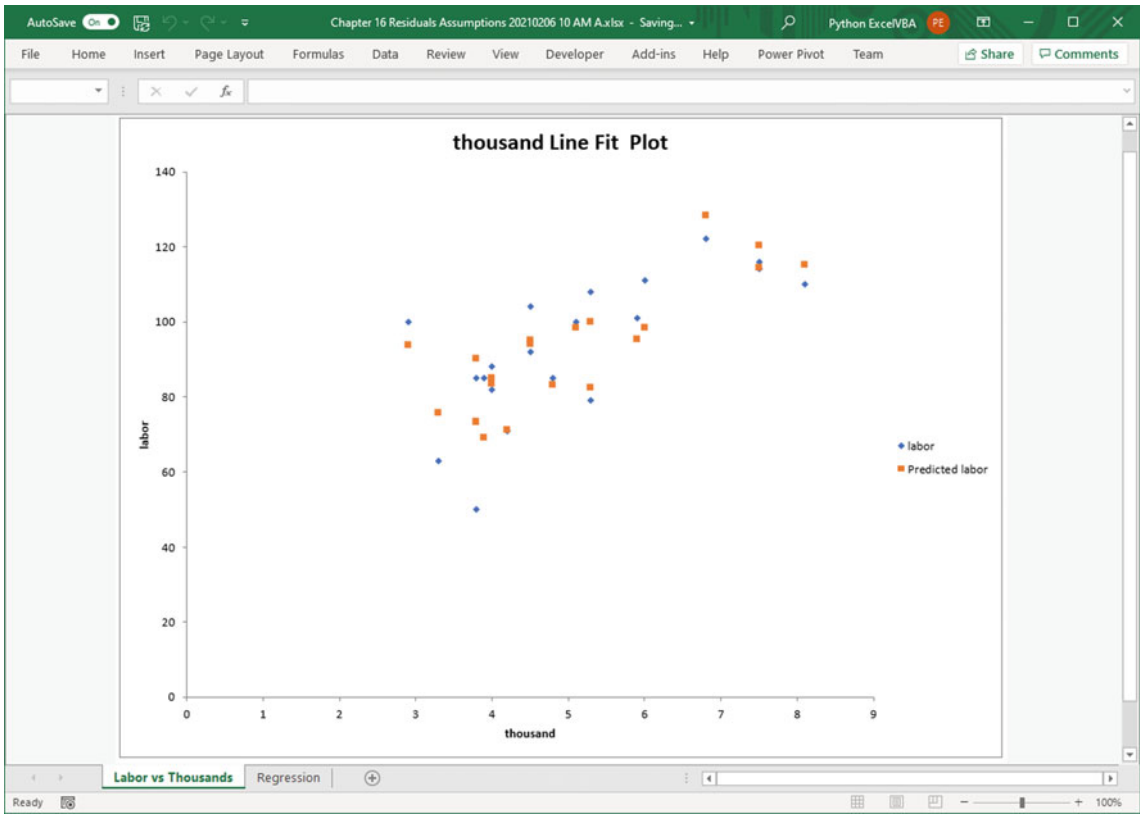
16.3 Linearity

One way to check linearity is to graph the y variable with each x variable to check if the graph looks linear. If one of these graphs does not look linear, then the linearity assumption might be violated. Below are three graphs of y with each of the three x variables. For each graph, we will also look at the strength of the relationship between y and x.

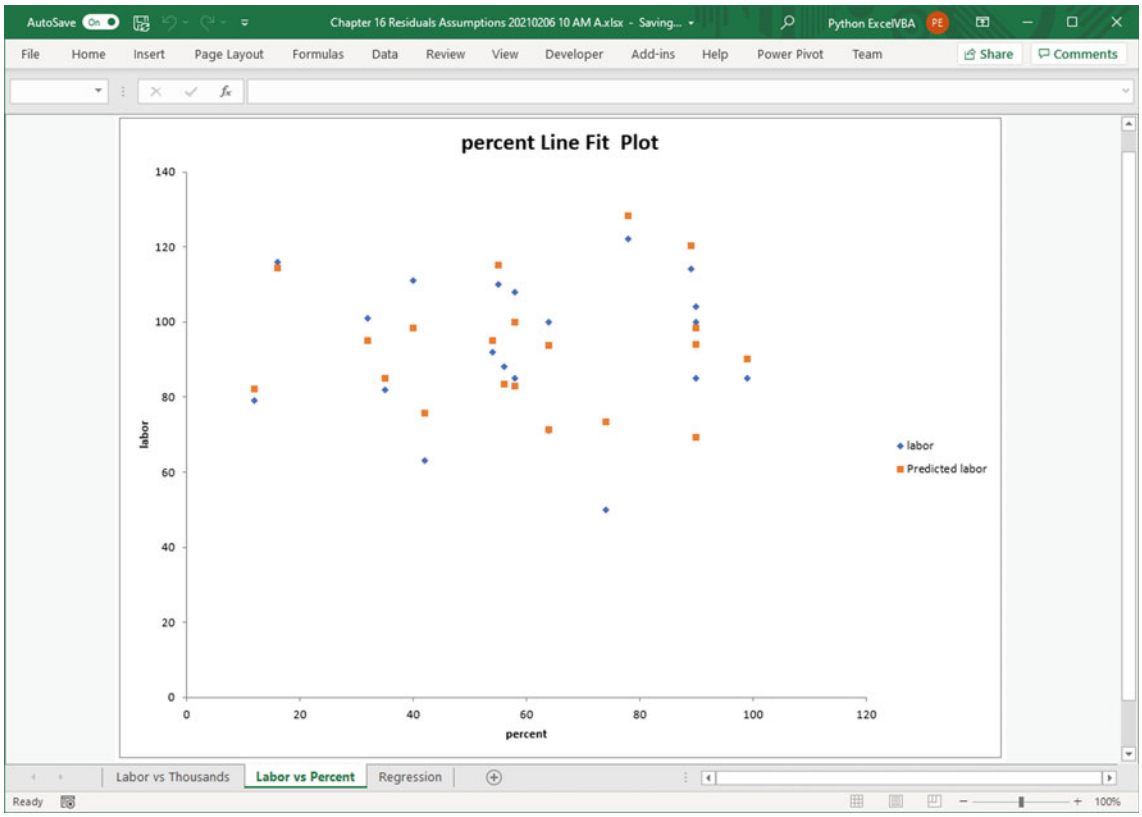
In the Excel *Regression* screen, there is a *Line Fit Plots* option.



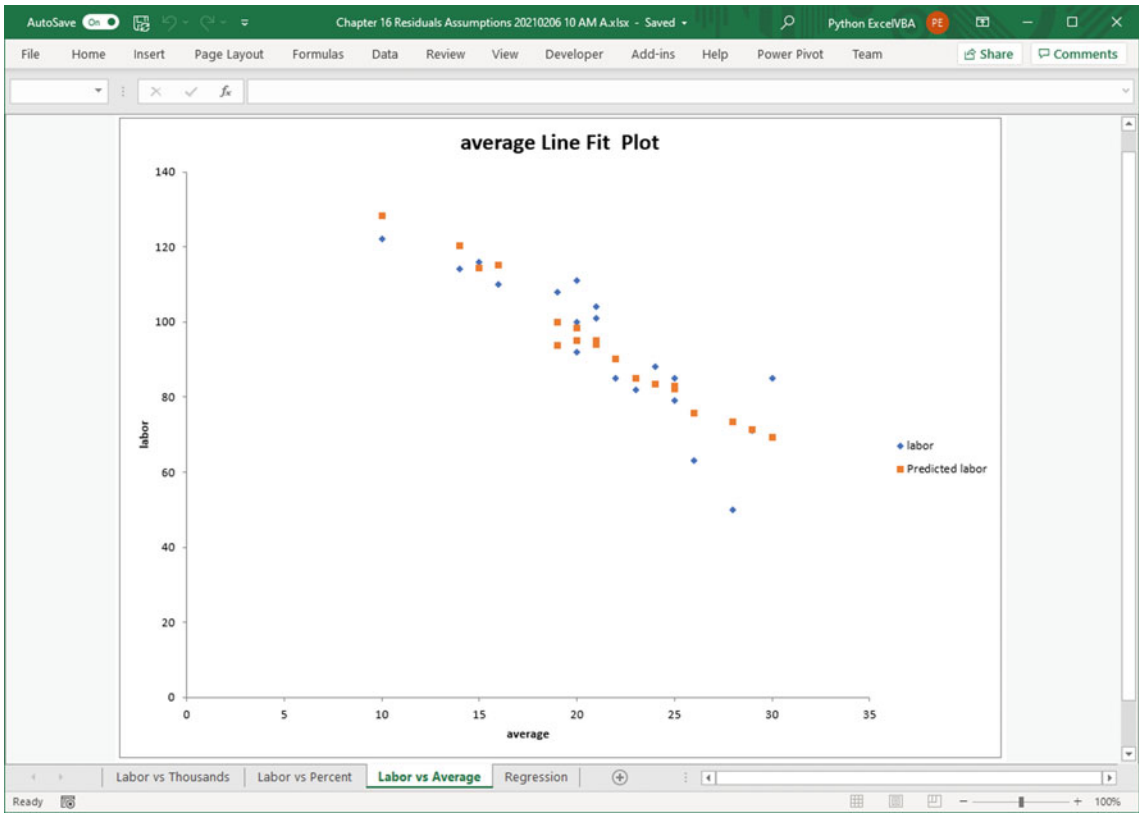
The *Line Fit Plots* option plots the y variable with each x variable. The following is the graph of labor versus thousands.



The following is the graph of labor versus percent.



The following is the graph of labor versus average.

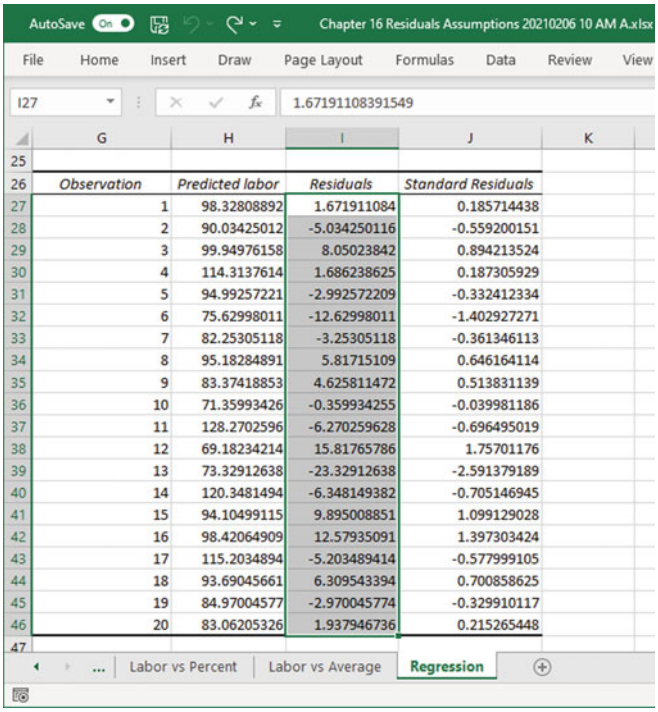


Of the three plots, the second one, *labor vs. percent*, might indicate that the linear assumption is being violated. In this plot, we can see that there is no general linear relationship. Also, from the correlation number of 0.031 shown below, we can see that there is only a weak association between the two variables.

16.4 The Expected Value of the Residual Term is Zero

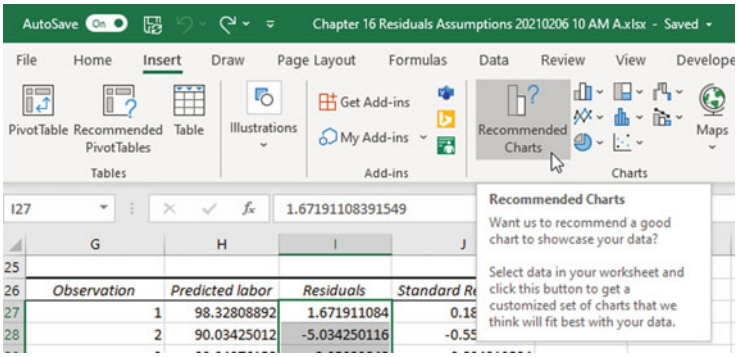
One way to check this assumption is to produce a boxplot of the residual terms and see if the midpoint is zero.

Do the following to produce a boxplot of the residual terms. First, select the residuals from the regression report.

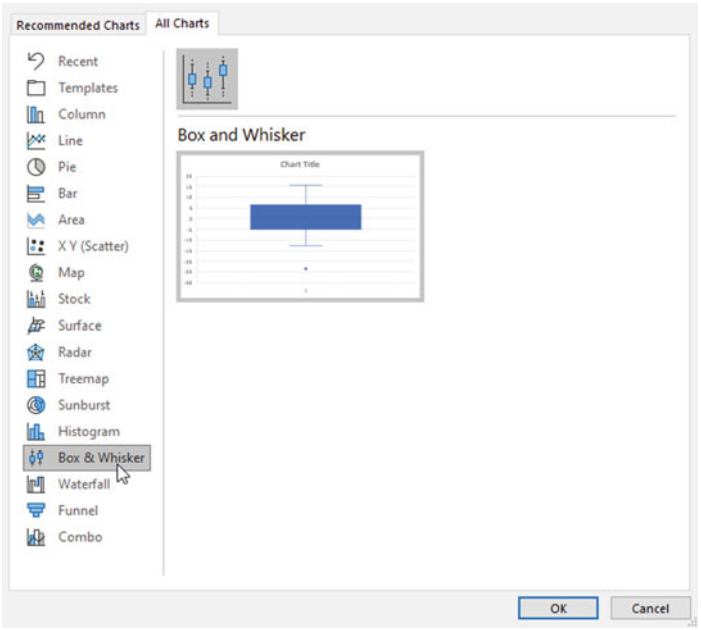


	G	H	I	J	K
25					
26	*Observation*	*Predicted labor*	*Residuals*	*Standard Residuals*	
27	1	98.32808892	1.671911084	0.185714438	
28	2	90.03425012	-5.034250116	-0.559200151	
29	3	99.94976158	8.05023842	0.894213524	
30	4	114.3137614	1.686238625	0.187305929	
31	5	94.99257221	-2.992572209	-0.332412334	
32	6	75.62998011	-12.62998011	-1.402927271	
33	7	82.25305118	-3.25305118	-0.361346113	
34	8	95.18284891	5.81715109	0.646164114	
35	9	83.37418853	4.625811472	0.513831139	
36	10	71.35993426	-0.359934255	-0.039981186	
37	11	128.2702596	-6.270259628	-0.696495019	
38	12	69.18234214	15.81765786	1.75701176	
39	13	73.32912638	-23.32912638	-2.591379189	
40	14	120.3481494	-6.348149382	-0.705146945	
41	15	94.10499115	9.895008851	1.099129028	
42	16	98.42064909	12.57935091	1.397303424	
43	17	115.2034894	-5.203489414	-0.577999105	
44	18	93.69045661	6.309543394	0.700858625	
45	19	84.97004577	-2.970045774	-0.329910117	
46	20	83.06205326	1.937946736	0.215265448	
47					

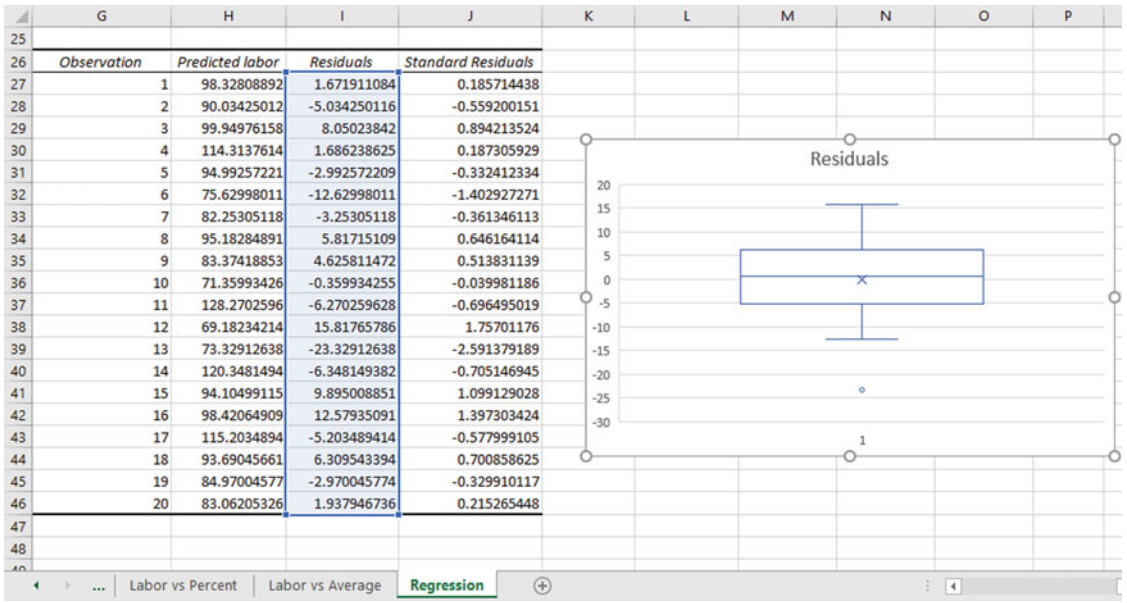
Next, click on *Recommended Charts* on the *Inserts* tab.



Next click on the *Box and Wisker* on the *All Charts* tab.



Below is the boxplot for the residuals.

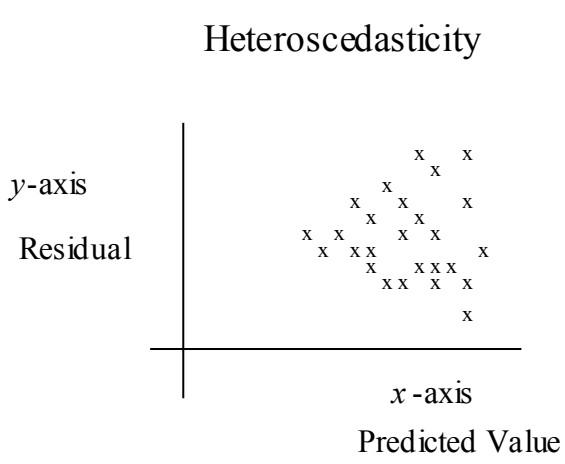


	G	H	I	J
25				
26	*Observation*	*Predicted labor*	*Residuals*	*Standard Residuals*
27	1	98.32808892	1.671911084	0.185714438
28	2	90.03425012	-5.034250116	-0.559200151
29	3	99.94976158	8.05023842	0.894213524
30	4	114.3137614	1.686238625	0.187305929
31	5	94.99257221	-2.992572209	-0.332412334
32	6	75.62998011	-12.62998011	-1.402927271
33	7	82.25305118	-3.25305118	-0.361346113
34	8	95.18284891	5.81715109	0.646164114
35	9	83.37418853	4.625811472	0.513831139
36	10	71.35993426	-0.359934255	-0.039981186
37	11	128.2702596	-6.270259628	-0.696495019
38	12	69.18234214	15.81765786	1.75701176
39	13	73.32912638	-23.32912638	-2.591379189
40	14	120.3481494	-6.348149382	-0.705146945
41	15	94.10499115	9.895008851	1.099129028
42	16	98.42064909	12.57935091	1.397303424
43	17	115.2034894	-5.203489414	-0.577999105
44	18	93.69045661	6.309543394	0.700858625
45	19	84.97004577	-2.970045774	-0.329910117
46	20	83.06205326	1.937946736	0.215265448
47				
48				

From the above boxplot, we can see that the midpoint is very close to zero.

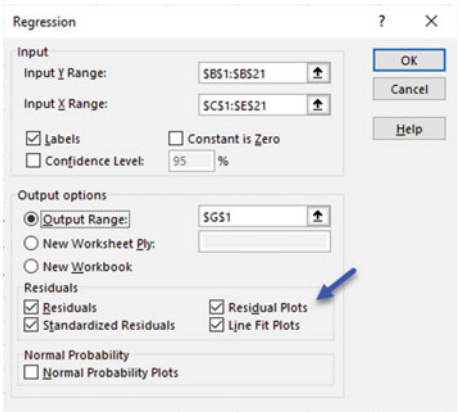
16.5 The Variance of the Error Term is Constant

One quick way to check to see if the variance of the error term is constant is to see if the graph of the residuals looks like the following.

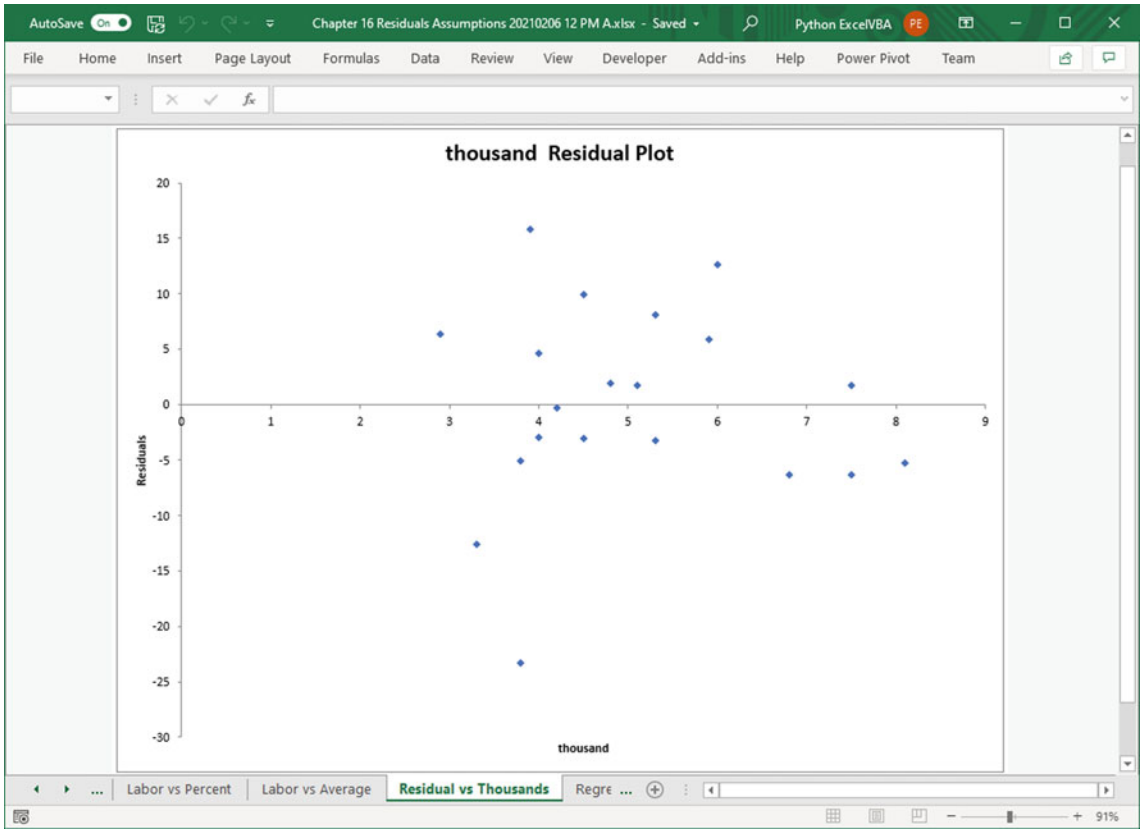


If the graph looks like the above, then the assumption that the variance of the error terms is constant is violated. This problem is called *heteroscedasticity*. Below we will graph the residuals against the independent variables and graph the residuals against the expected values.

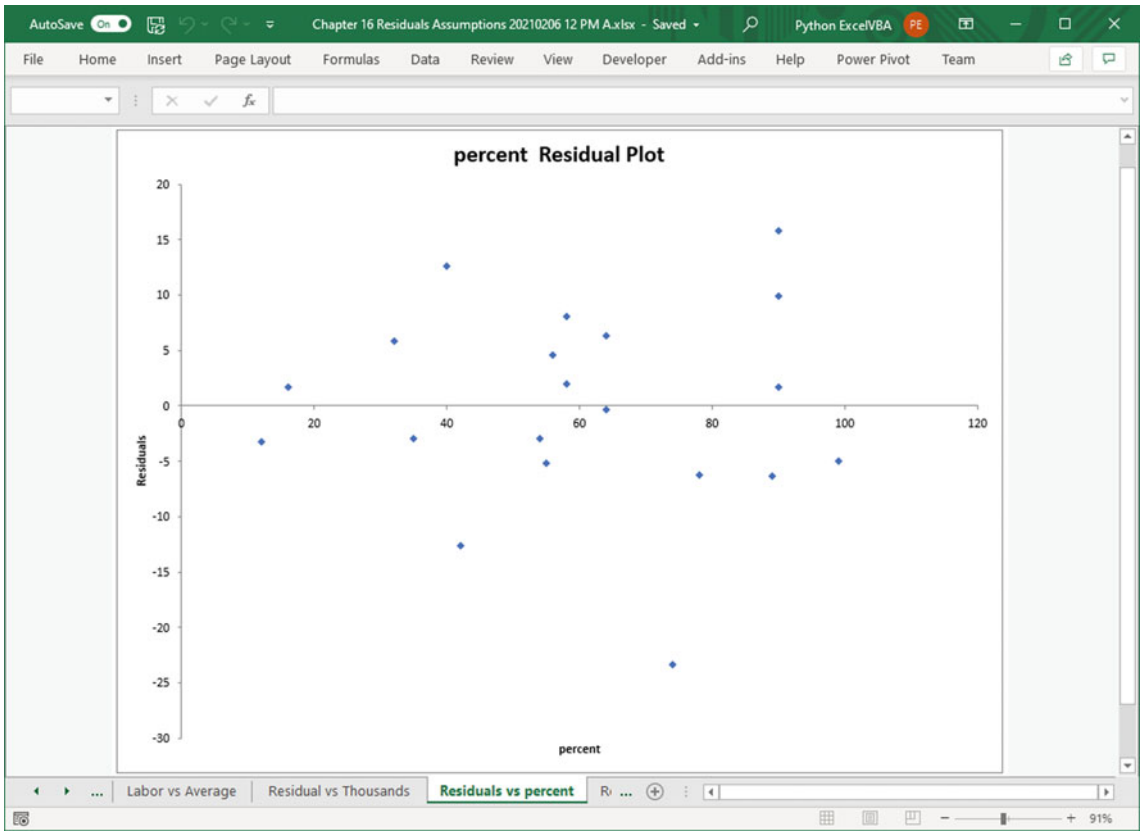
In the Excel *Regression* screen, there is a *Residual Plots* option.



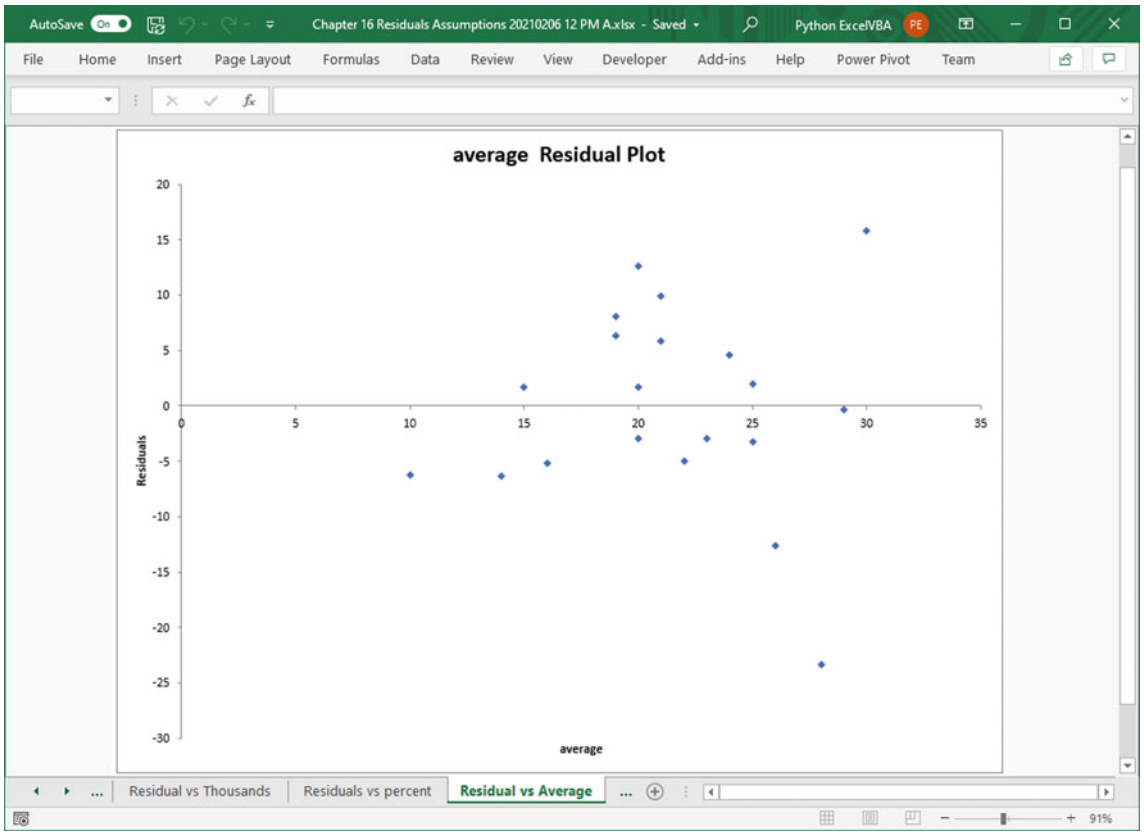
The *Residual Plots* option graphs the residuals with each independent variable. The following is residuals versus thousands.



The following is residuals versus percent.



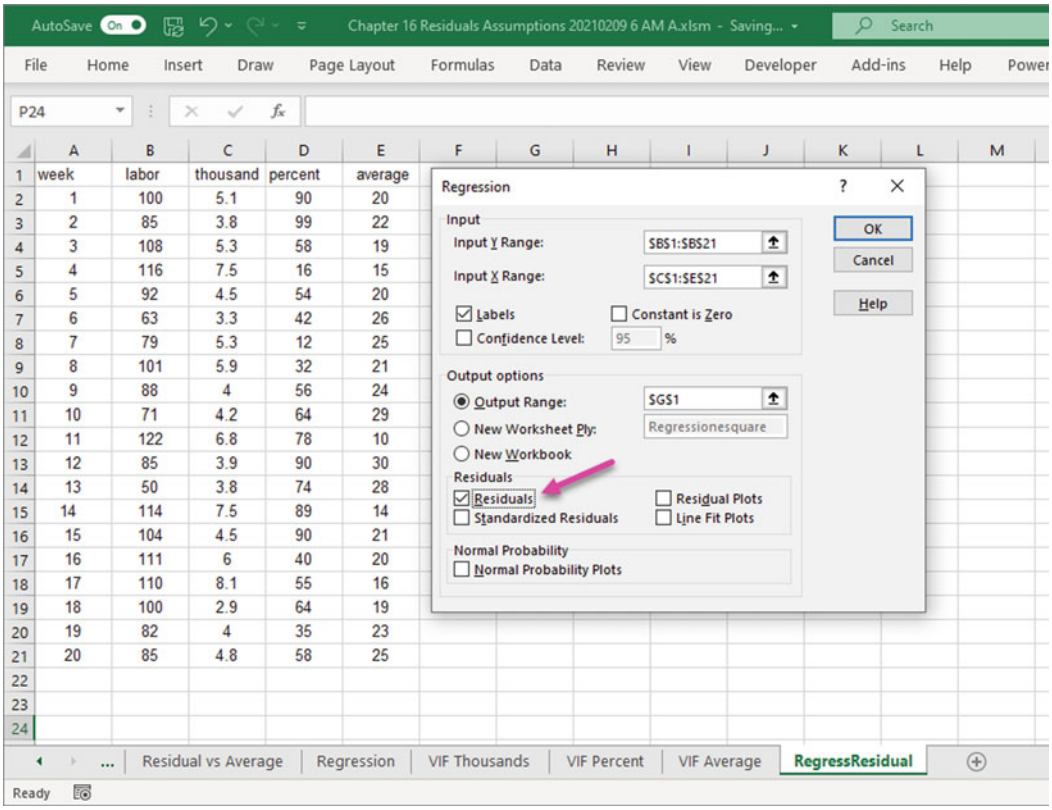
The following is residuals versus average.



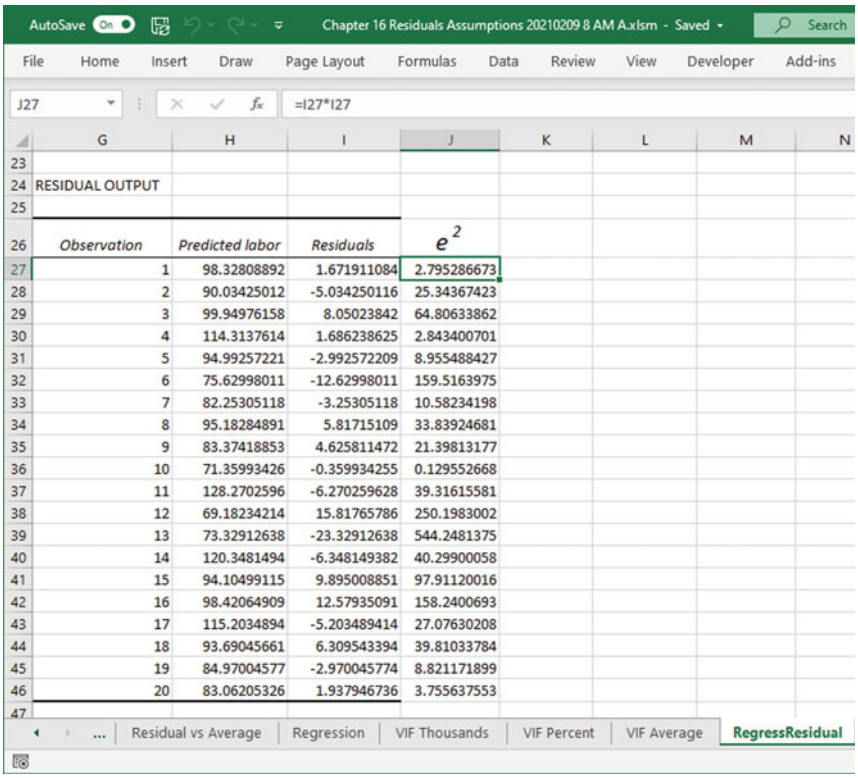
We can also test for heteroscedasticity by calculating the nR^2 statistics.
The following are the steps to calculate for the existence of heteroscedasticity.

(1) Run a standard regression
(2) Calculate the residuals, residual = expected − actual
(3) Run a regression using the square of the residuals as the dependent variable and the estimated dependent variable as the independent variable.
(4) Estimate nR^2, where n is the sample size and R^2 is the coefficient of determination.
(5) Using χ^2 statistics with 3 degrees of freedom to test whether nR^2 is significantly different from zero.

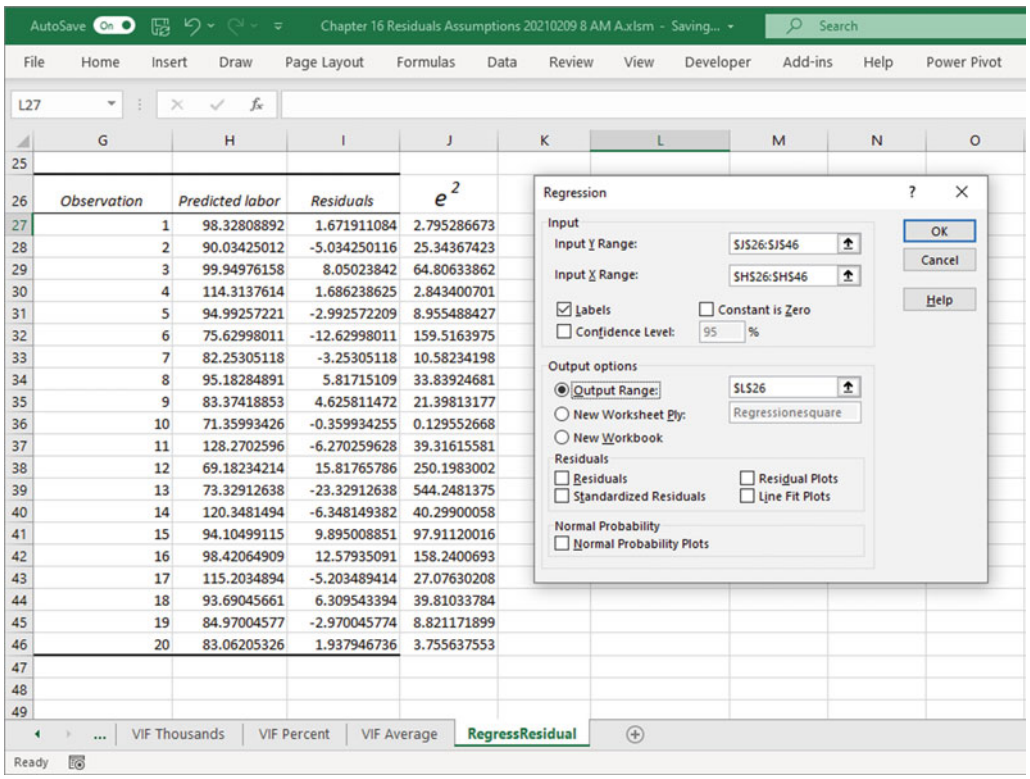
The following runs a regular regression with the *Residual* option selected.



In the *RESIDUAL OUTPUT* section, the regression report created a residual square column.

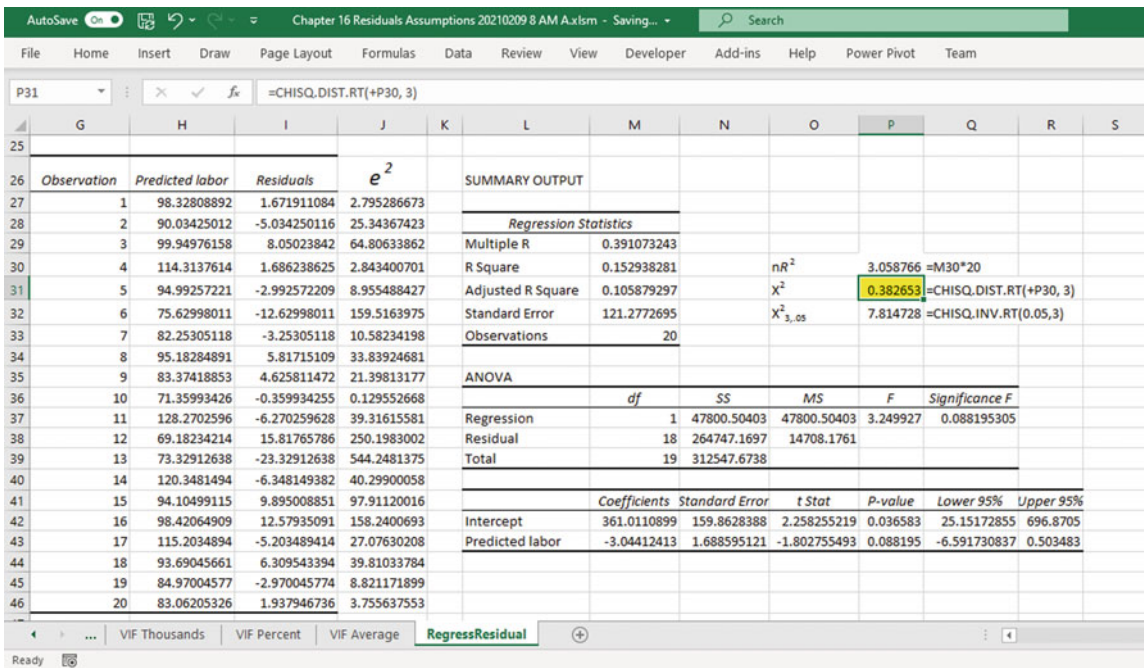


Run a regression where the residual square is the dependent variable, and the predicted labor value is the independent variable.



Observation	Predicted labor	Residuals	e^2
1	98.32808892	1.671911084	2.795286673
2	90.03425012	-5.034250116	25.34367423
3	99.94976158	8.05023842	64.80633862
4	114.3137614	1.686238625	2.843400701
5	94.99257221	-2.992572209	8.955488427
6	75.62998011	-12.62998011	159.5163975
7	82.25305118	-3.25305118	10.58234198
8	95.18284891	5.81715109	33.83924681
9	83.37418853	4.625811472	21.39813177
10	71.35993426	-0.359934255	0.129552668
11	128.2702596	-6.270259628	39.31615581
12	69.18234214	15.81765786	250.1983002
13	73.32912638	-23.32912638	544.2481375
14	120.3481494	-6.348149382	40.29900058
15	94.10499115	9.895008851	97.91120016
16	98.42064909	12.57935091	158.2400693
17	115.2034894	-5.203489414	27.07630208
18	93.69045661	6.309543394	39.81033784
19	84.97004577	-2.970045774	8.821171899
20	83.06205326	1.937946736	3.755637553

Calculate nR^2 and X^2



Observation	Predicted labor	Residuals	e^2
1	98.32808892	1.671911084	2.795286673
2	90.03425012	-5.034250116	25.34367423
3	99.94976158	8.05023842	64.80633862
4	114.3137614	1.686238625	2.843400701
5	94.99257221	-2.992572209	8.955488427
6	75.62998011	-12.62998011	159.5163975
7	82.25305118	-3.25305118	10.58234198
8	95.18284891	5.81715109	33.83924681
9	83.37418853	4.625811472	21.39813177
10	71.35993426	-0.359934255	0.129552668
11	128.2702596	-6.270259628	39.31615581
12	69.18234214	15.81765786	250.1983002
13	73.32912638	-23.32912638	544.2481375
14	120.3481494	-6.348149382	40.29900058
15	94.10499115	9.895008851	97.91120016
16	98.42064909	12.57935091	158.2400693
17	115.2034894	-5.203489414	27.07630208
18	93.69045661	6.309543394	39.81033784
19	84.97004577	-2.970045774	8.821171899
20	83.06205326	1.937946736	3.755637553

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.391073243
R Square	0.152938281
Adjusted R Square	0.105879297
Standard Error	121.2772695
Observations	20

ANOVA	df	SS	MS	F	Significance F
Regression	1	47800.50403	47800.50403	3.249927	0.088195305
Residual	18	264747.1697	14708.1761		
Total	19	312547.6738			

	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%
Intercept	361.0110899	159.8628388	2.258255219	0.036583	25.15172855	696.8705
Predicted labor	-3.04412413	1.688595121	-1.802755493	0.088195	-6.591730837	0.503483

nR^2	3.058766	=M30*20
X^2	0.382653	=CHISQ.DIST.RT(+P30, 3)
$X^2_{3,.05}$	7.814728	=CHISQ.INV.RT(0.05,3)

X^2 was calculated as 0.382. X^2 at alpha 0.05 and 3 degrees of freedom is 7.81. Therefore, we can conclude that the residuals of the regression of labor have the same variance.

16.6 Autocorrelation Durbin–Watson Test

If the residual terms are not independent, we have a situation called *autocorrelation*. The Durbin–Watson test is a test statistic for detecting autocorrelation.

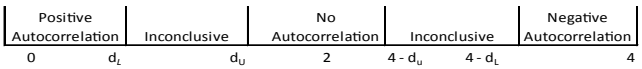
The DW statistic has a value between 0 and 4. A 0 means very strong positive autocorrelation. A 4 means very strong negative autocorrelation. A value of 2 means no or very little autocorrelation.

According to James Stevens:

> This is a very important assumption. . .. if independence is violated only mildly then the probability of a type one error will be *several* times greater than the level the experimenter thinks he or she is working at. Thus, instead of rejecting falsely 5% of the time, the experimenter may be rejecting falsely 25% or 30% of the time (Stevens 1986).

The definition of the DW statistics is

$$DW = \frac{\sum_{t=2}^{n} (e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$



We will create a custom Excel VBA function to calculate the Durban–Watson test.

16.6.1 VBA Code

The following is the custom Excel VBA function that calculates the Durban–Watson test. You need to pass the range of the residual data. The range needs to include the header.

```
'rResidual: residual data column. Include header
'Sample =SUMXMY2(I27:I46,I26:I45)/SUMSQ(I26:I46)
Function DurbanWatson(rResidual As Excel.Range) As Double

    Dim numerator As Double
    Dim denominator As Double
    Dim rFirstArgument As Excel.Range
    Dim rSecondArgument As Excel.Range

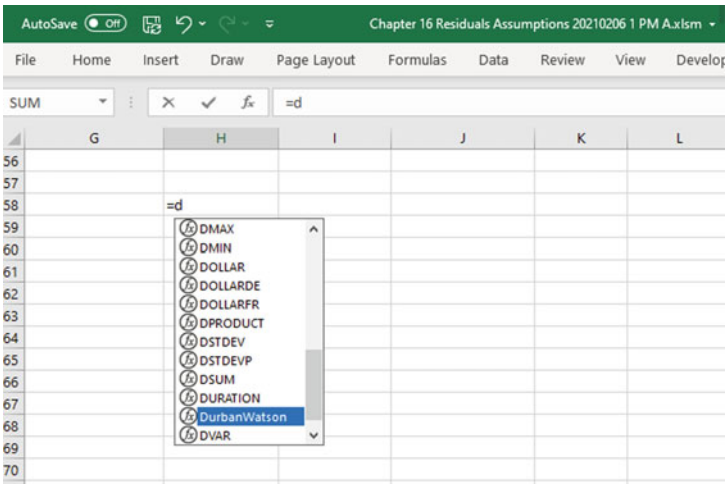
    'SUMXMY2(I27:I46,I26:I45)
    'first argument no header
    'second argument header minus last row
    Set rFirstArgument = rResidual.Resize(rResidual.Rows.Count - 1, 1).Offset(1, 0)
    Set rSecondArgument = rResidual.Resize(rResidual.Rows.Count - 1, 1)
    numerator = WorksheetFunction.SumXMY2(rFirstArgument, rSecondArgument)

    'SUMSQ(I26:I46)
    'No adjustment
    denominator = WorksheetFunction.SumSq(rResidual)

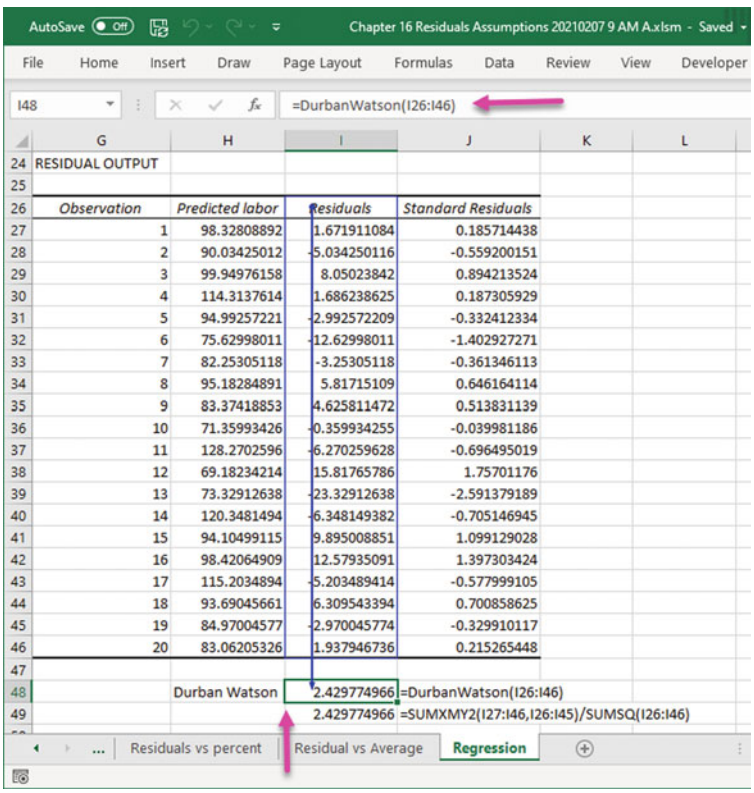
    DurbanWatson = numerator / denominator

End Function
```

The custom function *DurbanWatson* will show up in the Excel drop-down list.

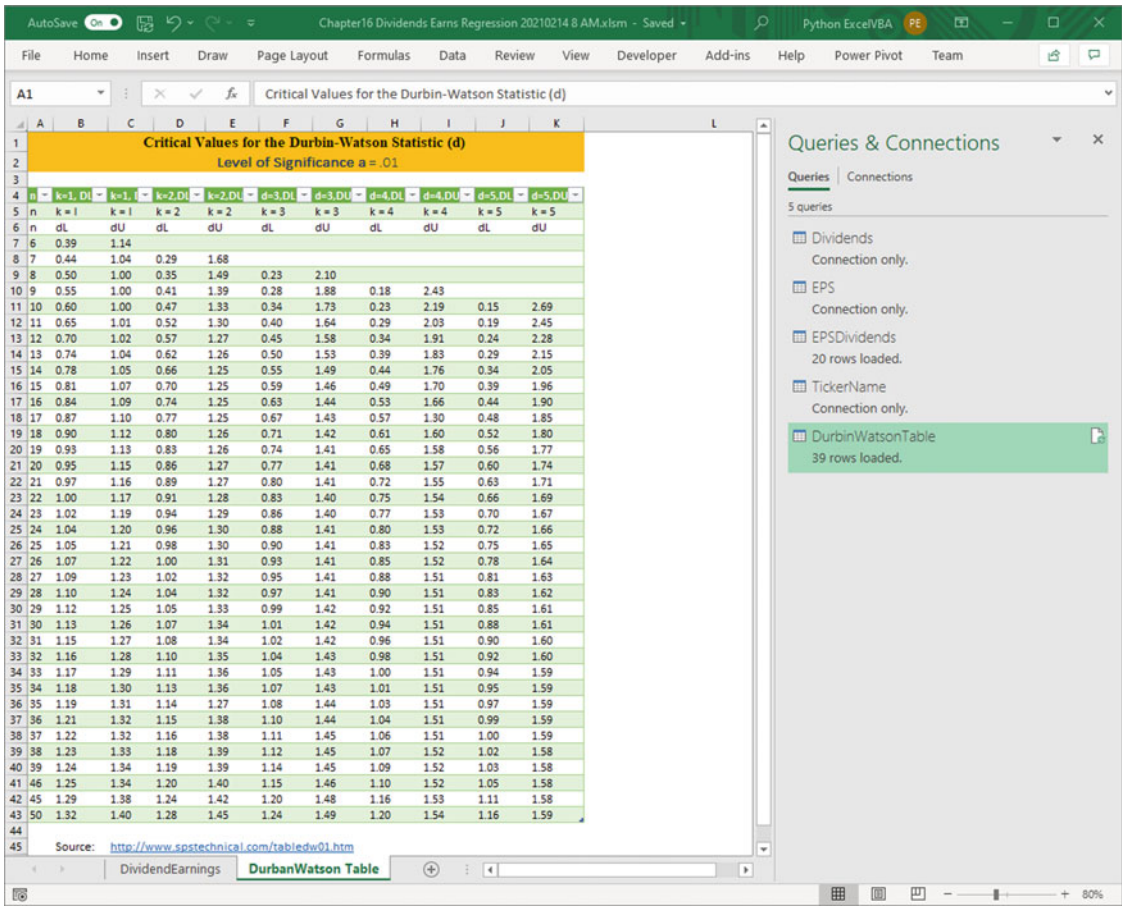


Below shows the use of the VBA function *DurbanWatson* and the equivalent Excel formula.



16.6.2 Durbin–Watson 1% Table

The URL http://www.spstechnical.com/tabledw01.htm shows a 1% Durbin–Watson table.



Critical Values for the Durbin-Watson Statistic (d)
Level of Significance a = .01

n	k=1, DL	k=1, DU	k=2,DL	k=2,DU	d=3,DL	d=3,DU	d=4,DL	d=4,DU	d=5,DL	d=5,DU
n	k = 1	k = 1	k = 2	k = 2	k = 3	k = 3	k = 4	k = 4	k = 5	k = 5
n	dL	dU	dL	dU	dL	dU	dL	dU	dL	dU
6	0.39	1.14								
7	0.44	1.04	0.29	1.68						
8	0.50	1.00	0.35	1.49	0.23	2.10				
9	0.55	1.00	0.41	1.39	0.28	1.88	0.18	2.43		
10	0.60	1.00	0.47	1.33	0.34	1.73	0.23	2.19	0.15	2.69
11	0.65	1.01	0.52	1.30	0.40	1.64	0.29	2.03	0.19	2.45
12	0.70	1.02	0.57	1.27	0.45	1.58	0.34	1.91	0.24	2.28
13	0.74	1.04	0.62	1.26	0.50	1.53	0.39	1.83	0.29	2.15
14	0.78	1.05	0.66	1.25	0.55	1.49	0.44	1.76	0.34	2.05
15	0.81	1.07	0.70	1.25	0.59	1.46	0.49	1.70	0.39	1.96
16	0.84	1.09	0.74	1.25	0.63	1.44	0.53	1.66	0.44	1.90
17	0.87	1.10	0.77	1.25	0.67	1.43	0.57	1.30	0.48	1.85
18	0.90	1.12	0.80	1.26	0.71	1.42	0.61	1.60	0.52	1.80
19	0.93	1.13	0.83	1.26	0.74	1.41	0.65	1.58	0.56	1.77
20	0.95	1.15	0.86	1.27	0.77	1.41	0.68	1.57	0.60	1.74
21	0.97	1.16	0.89	1.27	0.80	1.41	0.72	1.55	0.63	1.71
22	1.00	1.17	0.91	1.28	0.83	1.40	0.75	1.54	0.66	1.69
23	1.02	1.19	0.94	1.29	0.86	1.40	0.77	1.53	0.70	1.67
24	1.04	1.20	0.96	1.30	0.88	1.41	0.80	1.53	0.72	1.66
25	1.05	1.21	0.98	1.30	0.90	1.41	0.83	1.52	0.75	1.65
26	1.07	1.22	1.00	1.31	0.93	1.41	0.85	1.52	0.78	1.64
27	1.09	1.23	1.02	1.32	0.95	1.41	0.88	1.51	0.81	1.63
28	1.10	1.24	1.04	1.32	0.97	1.41	0.90	1.51	0.83	1.62
29	1.12	1.25	1.05	1.33	0.99	1.42	0.92	1.51	0.85	1.61
30	1.13	1.26	1.07	1.34	1.01	1.42	0.94	1.51	0.88	1.61
31	1.15	1.27	1.08	1.34	1.02	1.42	0.96	1.51	0.90	1.60
32	1.16	1.28	1.10	1.35	1.04	1.43	0.98	1.51	0.92	1.60
33	1.17	1.29	1.11	1.36	1.05	1.43	1.00	1.51	0.94	1.59
34	1.18	1.30	1.13	1.36	1.07	1.43	1.01	1.51	0.95	1.59
35	1.19	1.31	1.14	1.27	1.08	1.44	1.03	1.51	0.97	1.59
36	1.21	1.32	1.15	1.38	1.10	1.44	1.04	1.51	0.99	1.59
37	1.22	1.32	1.16	1.38	1.11	1.45	1.06	1.51	1.00	1.59
38	1.23	1.33	1.18	1.39	1.12	1.45	1.07	1.52	1.02	1.58
39	1.24	1.34	1.19	1.39	1.14	1.45	1.09	1.52	1.03	1.58
41	46	1.25	1.34	1.20	1.40	1.15	1.46	1.10	1.52	1.05	1.58
42	45	1.29	1.38	1.24	1.42	1.20	1.48	1.16	1.53	1.11	1.58
43	50	1.32	1.40	1.28	1.45	1.24	1.49	1.20	1.54	1.16	1.59

Source: http://www.spstechnical.com/tabledw01.htm

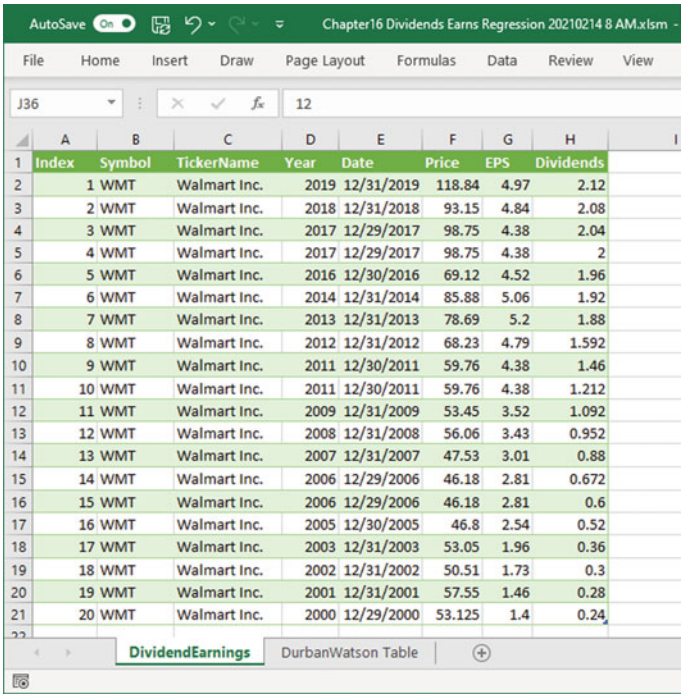
This table gives two values, d_L and d_U, for different sample sizes n and number of independent variable k. If the DW statistics is between d_U and $4 - d_U$, the null hypothesis that the correlation between lagged errors is equal to zero is accepted. If the statistic is less than d_L, that null hypothesis is rejected in favor of positive autocorrelation. Negative autocorrelation exists if the DW is greater than $4 - d_L$. There are two zones in which no judgment can be made. The two zones are between d_L and d_U and between $4 - d_U$ and $4 - d_L$.

The following is the *Power Query* M code to retrieve and format the Durban–Watson table.

```
let
    Source = Web.Page(Web.Contents("http://www.spstechnical.com/tabledw01.htm")),
    Data0 = Source{0}[Data],
    #"Changed Type" = Table.TransformColumnTypes(Data0,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text},
    {"Column5", type text}, {"Column6", type text}, {"Column7", type text}, {"Column8", type text}, {"Column9", type text}, {"Column10", type text},
    {"Column11", type text}}),
    #"Removed Top Rows" = Table.Skip(#"Changed Type",2),
    #"Removed Bottom Rows" = Table.RemoveLastN(#"Removed Top Rows",1),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Bottom Rows",{{"Column1", "n"}, {"Column2", "k=1, DL"}, {"Column3", "k=1, DU"},
    {"Column4", "k=2,DL"}, {"Column5", "k=2,DU"}, {"Column6", "d=3,DL"}, {"Column7", "d=3,DU"}, {"Column8", "d=4,DL"}, {"Column9", "d=4,DU"},
    {"Column10", "d=5,DL"}, {"Column11", "d=5,DU"}}),
    #"Removed Bottom Rows1" = Table.RemoveLastN(#"Renamed Columns",12)
in
    #"Removed Bottom Rows1"
```

16.7 Autocorrelation Walmart's Dividend and EPS from 2019 to 2000

The dividend per share (DPS_t) and earnings per share (EPS_t) from 2000 to 2019 for Walmart is shown below.



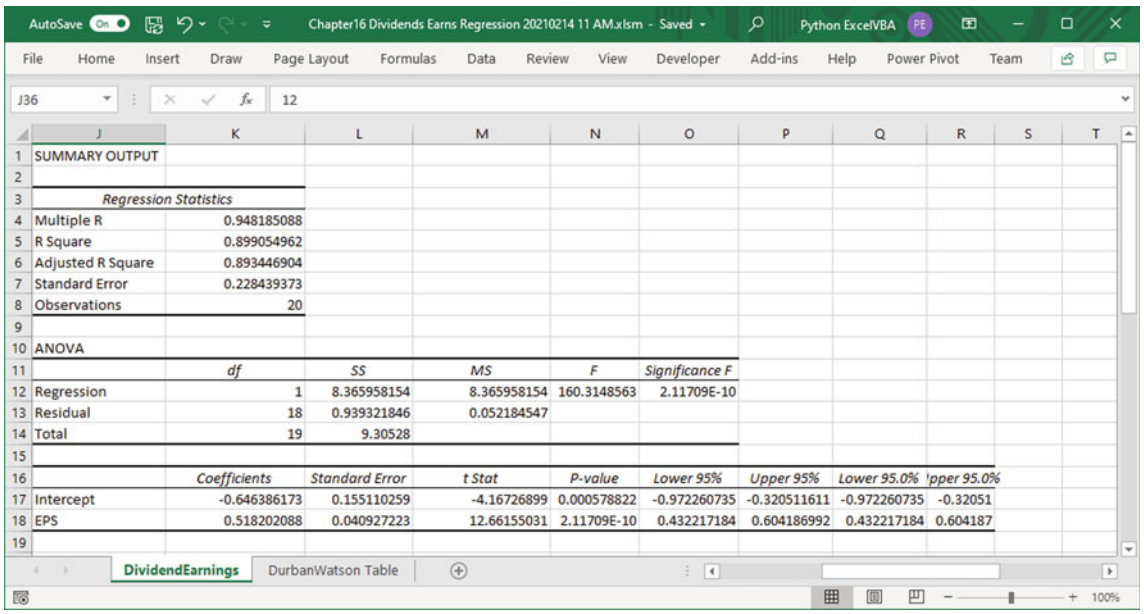
Index	Symbol	TickerName	Year	Date	Price	EPS	Dividends
1	WMT	Walmart Inc.	2019	12/31/2019	118.84	4.97	2.12
2	WMT	Walmart Inc.	2018	12/31/2018	93.15	4.84	2.08
3	WMT	Walmart Inc.	2017	12/29/2017	98.75	4.38	2.04
4	WMT	Walmart Inc.	2017	12/29/2017	98.75	4.38	2
5	WMT	Walmart Inc.	2016	12/30/2016	69.12	4.52	1.96
6	WMT	Walmart Inc.	2014	12/31/2014	85.88	5.06	1.92
7	WMT	Walmart Inc.	2013	12/31/2013	78.69	5.2	1.88
8	WMT	Walmart Inc.	2012	12/31/2012	68.23	4.79	1.592
9	WMT	Walmart Inc.	2011	12/30/2011	59.76	4.38	1.46
10	WMT	Walmart Inc.	2011	12/30/2011	59.76	4.38	1.212
11	WMT	Walmart Inc.	2009	12/31/2009	53.45	3.52	1.092
12	WMT	Walmart Inc.	2008	12/31/2008	56.06	3.43	0.952
13	WMT	Walmart Inc.	2007	12/31/2007	47.53	3.01	0.88
14	WMT	Walmart Inc.	2006	12/29/2006	46.18	2.81	0.672
15	WMT	Walmart Inc.	2006	12/29/2006	46.18	2.81	0.6
16	WMT	Walmart Inc.	2005	12/30/2005	46.8	2.54	0.52
17	WMT	Walmart Inc.	2003	12/31/2003	53.05	1.96	0.36
18	WMT	Walmart Inc.	2002	12/31/2002	50.51	1.73	0.3
19	WMT	Walmart Inc.	2001	12/31/2001	57.55	1.46	0.28
20	WMT	Walmart Inc.	2000	12/29/2000	53.125	1.4	0.24

The above data will be used to estimate the following regression equation,

$$\mathrm{DPS}_{i,t} = a_i + b_i \mathrm{EPS}_{it} + e_{it}$$

where $\mathrm{DPS}_{i,t}$ and EPS_{it} are dividends per share and earnings per share for the ith firm and in period t.

Excel's regression report is shown below,



SUMMARY OUTPUT									
Regression Statistics									
Multiple R	0.948185088								
R Square	0.899054962								
Adjusted R Square	0.893446904								
Standard Error	0.228439373								
Observations	20								
ANOVA									
	df	*SS*	*MS*	*F*	*Significance F*				
Regression	1	8.365958154	8.365958154	160.3148563	2.11709E-10				
Residual	18	0.939321846	0.052184547						
Total	19	9.30528							
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*pper 95.0%*	
Intercept	-0.646386173	0.155110259	-4.16726899	0.000578822	-0.972260735	-0.320511611	-0.972260735	-0.32051	
EPS	0.518202088	0.040927223	12.66155031	2.11709E-10	0.432217184	0.604186992	0.432217184	0.604187	

Looking up the value in the above DW 1% table, we find that for one independent variable and 20 observations, we find that d_L is 0.95 and d_U is 1.15. Because Walmart's DW number 0.46 is less than d_L, we reject the null hypothesis in favor of positive autocorrelation.

16.7.1 Data Source

16.7.1.1 Earnings Per Share

The data source for earnings per share was https://www.alphavantage.co. You will need to register for a few API key at https://www.alphavantage.co/support/#api-key.

The API to get WMT's earnings per share was,
https://www.alphavantage.co/query?function=EARNINGS&symbol=WMT&apikey=YourAPI.

A power query was created for each year's EPS. The following is the M code for year 2019.

The M code for the other year's EPS is similar.

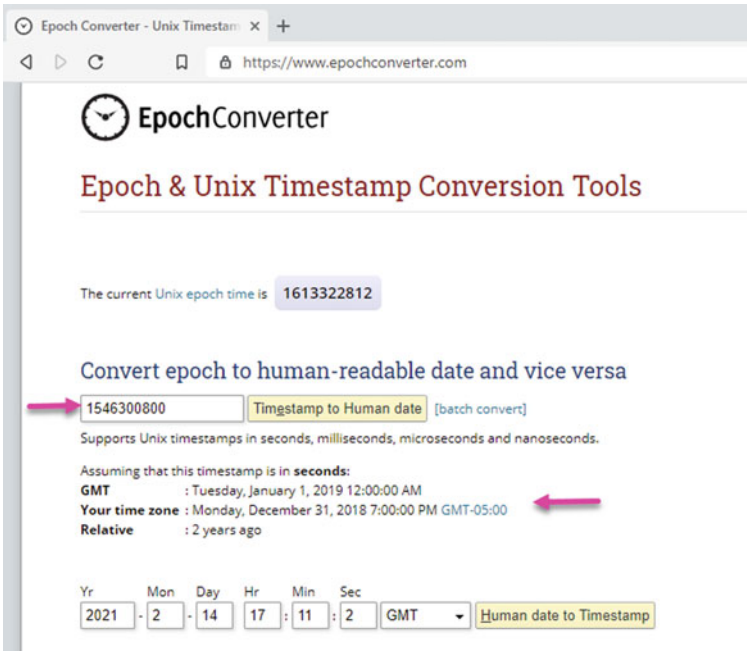
16.7.1.2 Dividends Per Share

The data source for dividends per share was https://finance.yahoo.com/quote/WMT/history.

The API to get Walmart's dividends for 2019 was,
https://query1.finance.yahoo.com/v7/finance/download/WMT?period1=1546300800&period2=1577750400&interval=1d&events=div&includeAdjustedClose=true.

The number used for period 1 and period 2 was epoch time. Go to https://www.epochconverter.com/ to convert from Epoch time to regular time.

EeochConverter.com indicated that 1,546,300,800 epoch time is the same as Monday, December 31, 2018, at 7 PM.



A power query was created for each year's DPS. The following is the M code for year 2019.

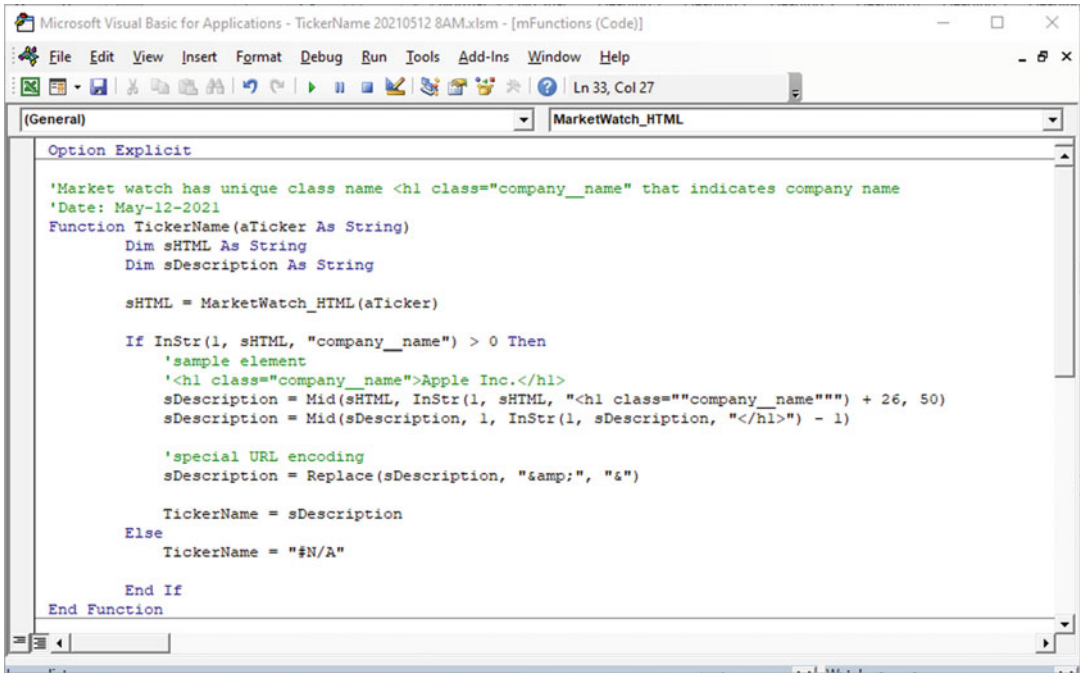
The M code for the other year's DPS is similar.

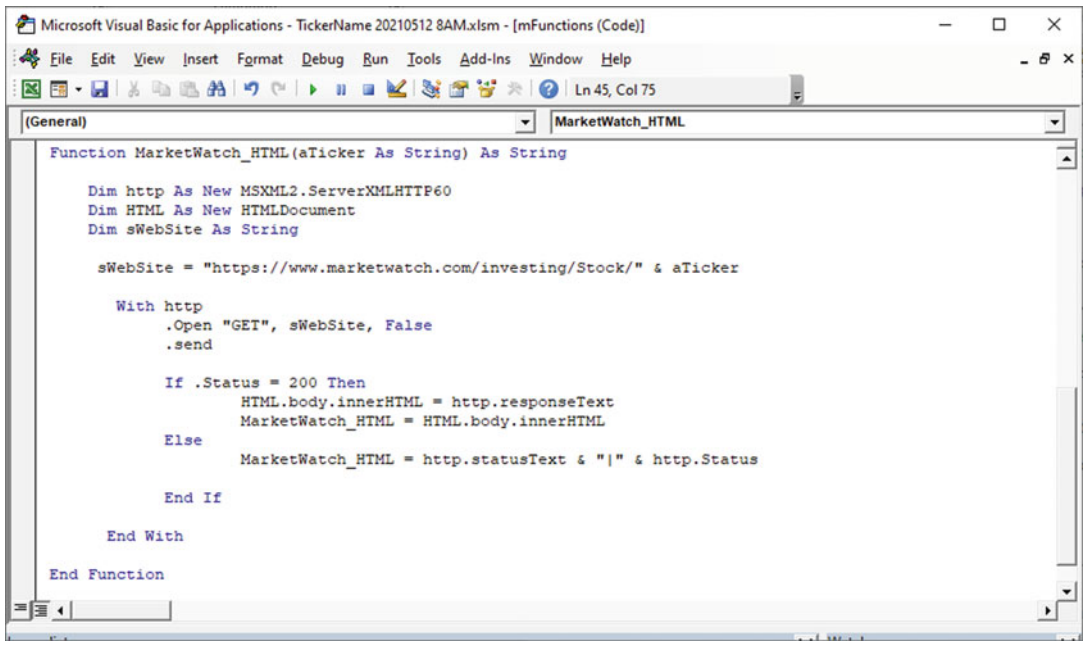
16.8 Using VBA to Retrieve a Ticker's Name

In this section, we will show VBA code and discuss how the VBA code retrieves a ticker's name from www.marketwatch.com.

The URL for WMT's ticker name is https://www.marketwatch.com/investing/stock/wmt.

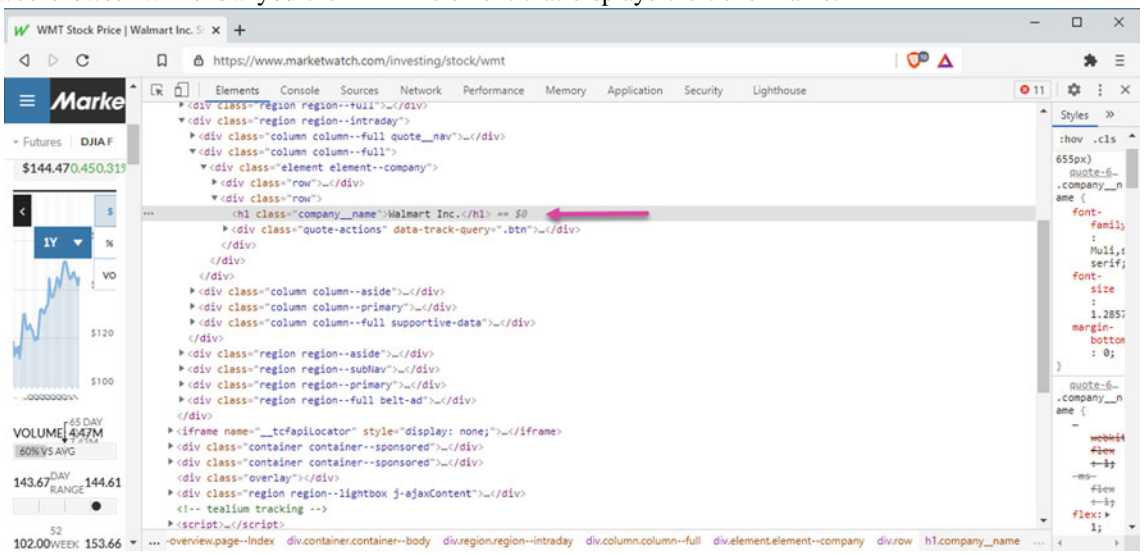
Below shows the VBA function *TickerName* and *MarketWatch_HTML*. The function *TickerName* uses the function *MarketWatch*.





Hypertext Markup Language, HTML, tells a web page on how to construct and display its contents. Right-mouse click on the ticker name and click on *Inspect*.

The web browser will show you the HTML element that displays the ticker name.

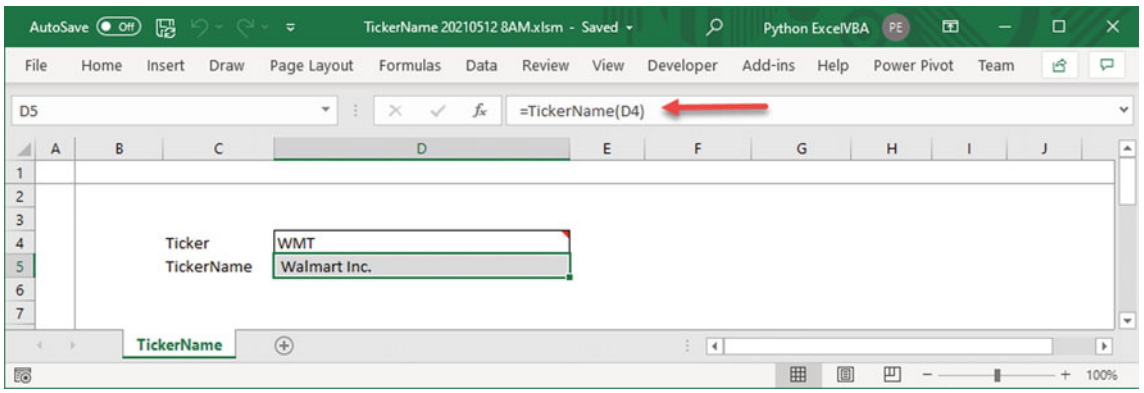


The element that contains the ticker name has an attribute called *class*. The class attribute has the value of *company__name*. If you search the HTML of this web page you will find out this is the only place in the HTML that has the text *company__name*.

The VBA function *MarketWatch_HTML* retrieves the HTML from marketwatch.com and gives it to the function *TickerName*. The function *TickerName* uses the *InStr* VBA function to find the text *company__name* in the HTML so it can return the ticker name.

```
'sample element
'<h1 class="company__name">Apple Inc.</h1>
sDescription = Mid(sHTML, InStr(1, sHTML, "<h1 class=""company__name""") + 26, 50)
sDescription = Mid(sDescription, 1, InStr(1, sDescription, "</h1>") - 1)
```

Below shows the function *TickerName* returning the ticker name for the ticker *WMT*.



16.9 Durbin–Watson Test Market Model Python Code

The following Python code performs the Durbin–Watson test on the market model to test where there is an autocorrelation for AAPL for the period March 7, 2016, to March 31, 2021.

```
[38] pip install requests_html
```

```
[39] pip install yfinance
```

```
[40] pip install requests
```

```
[41] pip install bs4
```

```
[42] #Date: Apr-05-2021
     import yfinance as yf
     import matplotlib.pyplot as plt
     import numpy as np
     import pandas as pd
     from scipy.stats import kurtosis, skew
     from tabulate import tabulate
     import datetime as dt#get current date
     import statsmodels.api as sm
     from statsmodels.formula.api import ols
     import requests
     from bs4 import BeautifulSoup
     import matplotlib.ticker as mtick
```

```
[43] def Usr_input():
       previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
       Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
       years_ago = dt.date.today()-dt.timedelta(days=(5 * 365) + 30)
       Default_start = years_ago.strftime("%Y-%m-%d")
       Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
       Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
       End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
       T_interval = input('Enter date interval(1d,1wk,1mo Default = 1mo)-->') or '1mo'
       return Ticker, Start, End, T_interval
```

```
[44] def calculate_ret(stock_a,stock_M,start,end_t,T_interval):
       Stock_price_a = yf.download(stock_a,start=start,end=end_t,interval = T_interval)['Adj Close'].dropna()
       Ret_a = Stock_price_a.pct_change()[1:]
       Stock_price_M = yf.download(stock_M,start=start,end=end_t,interval = T_interval)['Adj Close'].dropna()
       Ret_M = Stock_price_M.pct_change()[1:]
       Ret = pd.DataFrame([Ret_a,Ret_M])
       Ret = Ret.T
       Ret.columns = [stock_a,'SNP']
       return Ret
```

```
def html_beautifulsoup(url):
    response = requests.get(url)
    html = response.content

    #beautifulsoup html will not look like soure
    #review beautifulsoup html to use find and findall
    soup = BeautifulSoup(html,"lxml")
    return soup

#soup_html: soup html is not same as source html
def yahoo_description(soup_html):
    element = soup_html.find('h1',class_='D(ib) Fz(18px)')
    return element.text
```

```
(Ticker,macro_start,macro_end,macro_T_interval) = Usr_input()
soup_html = html_beautifulsoup("https://finance.yahoo.com/quote/"  +Ticker)
Ticker_shortN = yahoo_description(soup_html)
```

```
Please enter ticker(Default = JNJ)-->AAPL
Enter start date(Default = 5 Years ago, 2016-03-07)-->2016-03-07
Enter end date(Default = Yesterday, 2021-04-04)-->2021-03-31
Enter date interval(1d,1wk,1mo Default = 1mo)-->1mo
```

```
ret = calculate_ret(Ticker,'^GSPC',macro_start,macro_end,macro_T_interval)
ret_M = ret['SNP']
ret_a = ret[Ticker]

ret_Mc = sm.add_constant(ret_M)
results = sm.OLS(ret_a,ret_Mc).fit()
print('                              Market Model')
print(results.summary())
```

```
[**********100%**********]  1 of 1 completed
[**********100%**********]  1 of 1 completed
                            Market Model
                          OLS Regression Results
==============================================================================
Dep. Variable:                   AAPL   R-squared:                       0.385
Model:                            OLS   Adj. R-squared:                  0.374
Method:                 Least Squares   F-statistic:                     35.70
Date:                Mon, 05 Apr 2021   Prob (F-statistic):           1.58e-07
Time:                        11:19:28   Log-Likelihood:                 77.457
No. Observations:                  59   AIC:                            -150.9
Df Residuals:                      57   BIC:                            -146.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0186      0.009      2.071      0.043       0.001       0.036
SNP            1.2001      0.201      5.975      0.000       0.798       1.602
==============================================================================
Omnibus:                        5.668   Durbin-Watson:                   2.150
Prob(Omnibus):                  0.059   Jarque-Bera (JB):                4.945
Skew:                          -0.508   Prob(JB):                       0.0844
Kurtosis:                       3.989   Cond. No.                         23.3
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
DW = np.sum( np.diff( results.resid.values )**2.0 ) / results.ssr
DW = round(DW, 3)
```

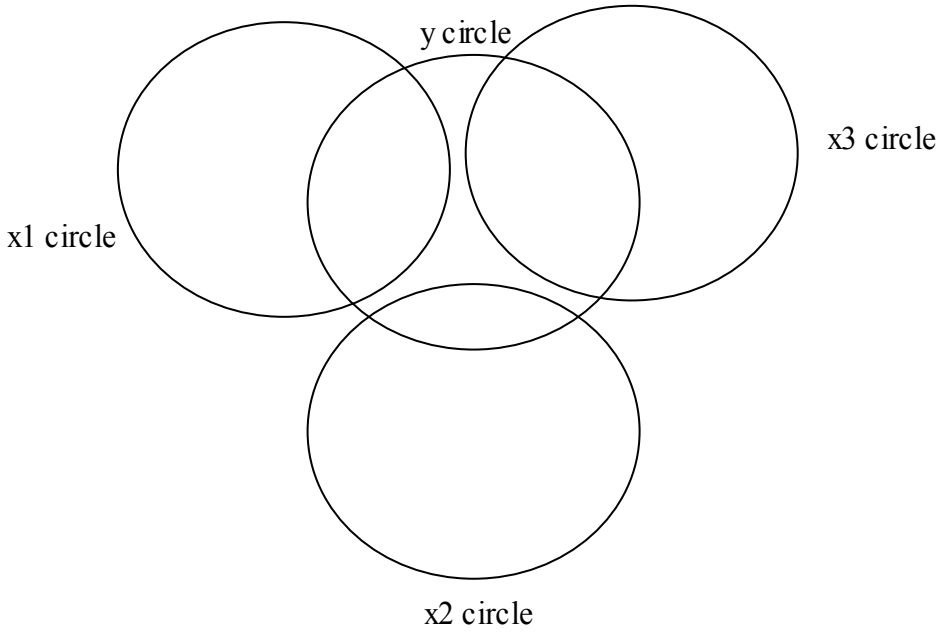
```
##Using significant level as 5 %, we find that dL is 1.20 and dU is 1.41.
dL = 1.2
dU = 1.41

if DW >0 and DW < dL:
  print("Durbin-Watson statistic(DW) is " + str(DW) + " , we have dL as 1.2, dU as 1.41. \
  \nBecause DW value is greater than zero and less than dL, our test result is positive correlation")
elif DW > dL and DW < dU:
  print("Durbin-Watson statistic(DW) is " + str(DW) + " , we have dL as 1.2, dU as 1.41. \
  \nBecause DW value is greater than dL and less than dU, our test result is inconclusive")
elif DW > dU and DW < 4 - dU:
  print("Durbin-Watson statistic(DW) is " + str(DW) + " , we have dL as 1.2, dU as 1.41. \
  \nBecause DW value is greater than dU and less than 4 - dU, our test result is no correlation")
elif DW > 4 - dU and DW < 4 - dL:
  print("Durbin-Watson statistic(DW) is " + str(DW) + " , we have dL as 1.2, dU as 1.41. \
  \nBecause DW value is greater than 4 - dU and less than 4 - dL, our test result is inconclusive")
else:
  print("Durbin-Watson statistic(DW) is " + str(DW) + " , we have dL as 1.2, dU as 1.41. \
  \nBecause DW value is greater than 4 - dL and less than 4, our test result is negative correlation")
```

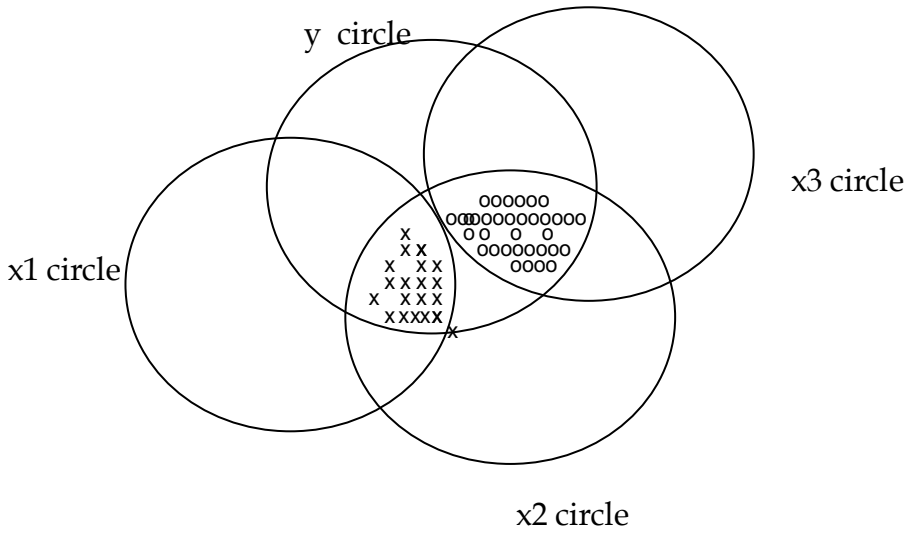
```
Durbin-Watson statistic(DW) is 2.15 , we have dL as 1.2, dU as 1.41.
Because DW value is greater than dU and less than 4 - dU, our test result is no correlation
```

16.10 The Independent Variables Are Uncorrelated: Multicollinearity

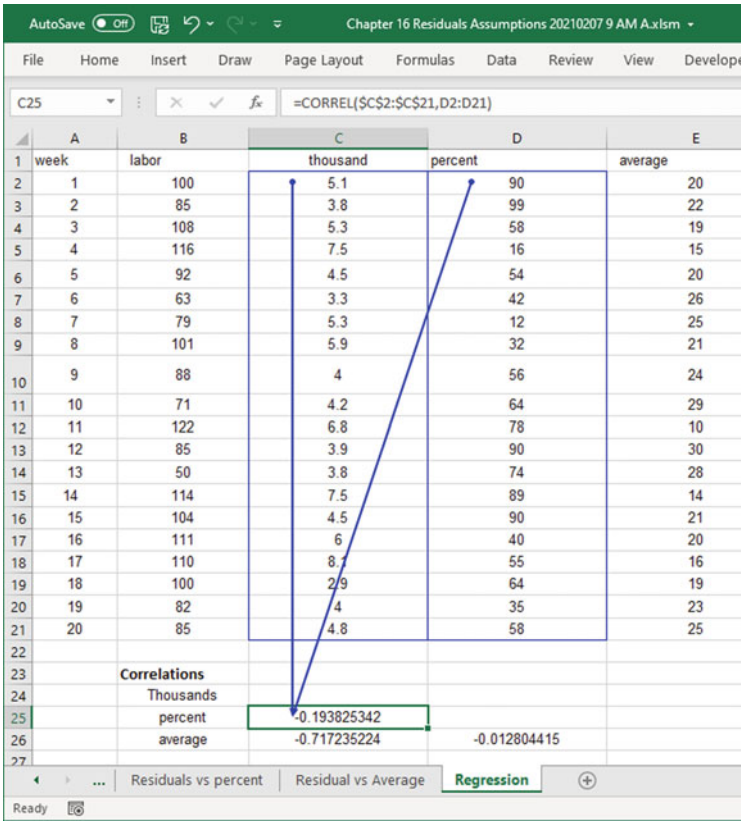
The next thing we would like to see is how correlated the independent variables are. The more correlated the independent variables, the less sure we are about how much each independent variable explains the dependent variable. Multicollinearity can cause problems like inflating regression coefficients, producing a t statistic that is too small (Kvanli et al. 1992). The ideal situation would be no correlation at all between the independent variables. This would look like the following, using a Venn diagram.



In this diagram, none of the independent variables intersects any other independent variable. But most of the time, this is not the case. Usually, there are intersections among the independent variables, indicating some degree of correlation. This would look like the following in a Venn diagram.



The area with small x's indicates multicollinearity between circles × 1 and × 2. This means that × 1 and × 2 are both explaining the same part of *y*. Similarly, we can see that there is multicollinearity between × 2 and × 3 by the small o's. This means that × 2 and × 3 are explaining the same part of *y*. We can see therefore that circle × 2 is not very useful in the regression because the part of *y* that it is explaining is also being explained by × 1 and × 3.



	A	B	C	D	E
1	week	labor	thousand	percent	average
2	1	100	5.1	90	20
3	2	85	3.8	99	22
4	3	108	5.3	58	19
5	4	116	7.5	16	15
6	5	92	4.5	54	20
7	6	63	3.3	42	26
8	7	79	5.3	12	25
9	8	101	5.9	32	21
10	9	88	4	56	24
11	10	71	4.2	64	29
12	11	122	6.8	78	10
13	12	85	3.9	90	30
14	13	50	3.8	74	28
15	14	114	7.5	89	14
16	15	104	4.5	90	21
17	16	111	6	40	20
18	17	110	8.1	55	16
19	18	100	2.9	64	19
20	19	82	4	35	23
21	20	85	4.8	58	25
22					
23		Correlations			
24		Thousands			
25		percent	-0.193825342		
26		average	-0.717235224	-0.012804415	
27					

From the above Excel multicollinearity table, we can see that there is strong multicollinearity between thousands of pounds shipped and the average number of pounds per shipment.

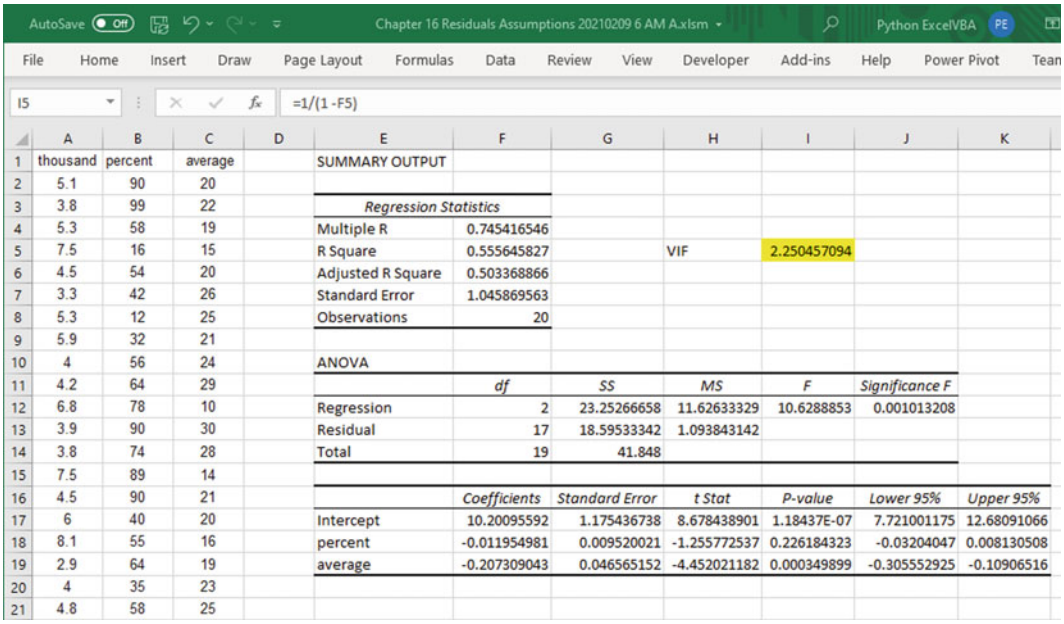
16.11 Variance Inflationary Factor (VIF)

The correlation matrix only measures the correlation between one independent variable with another independent variable and not the possible correlation between any one independent variable and all others taken as a group. A statistic called the *variance inflationary factor (VIF)* measures the correlation between any one independent variable and all others were taken as a group. Researchers have used VIF = 10 as a critical value rule of thumb to determine whether too much correlation exists between an independent variable and the other independent variables taken as a group. The VIF is defined for each independent variable as

$$VIF = \frac{1}{1 - R^2}$$

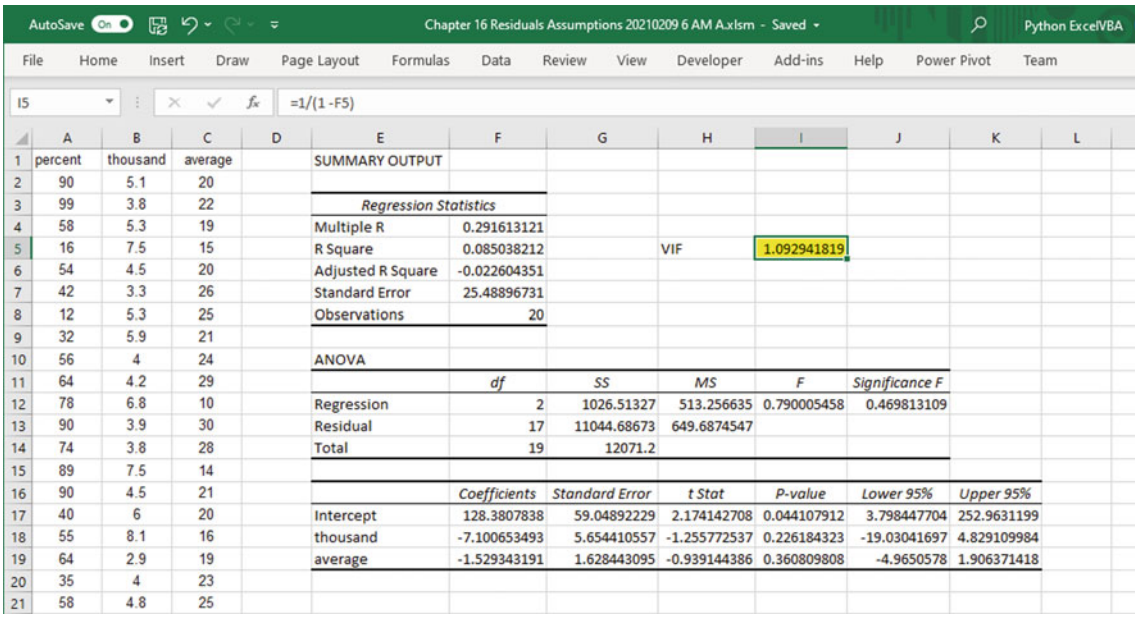
The steps to calculate the VIF for each independent variable would be to regress each of the independent variables against the other independent variables. In our example, we would do three independent variable regressions. The next step then would be to find the R^2 for each regression and then calculate the VIF.

The VIF for the "thousands" independent variable is 2.25. The calculation to get this number is shown below.



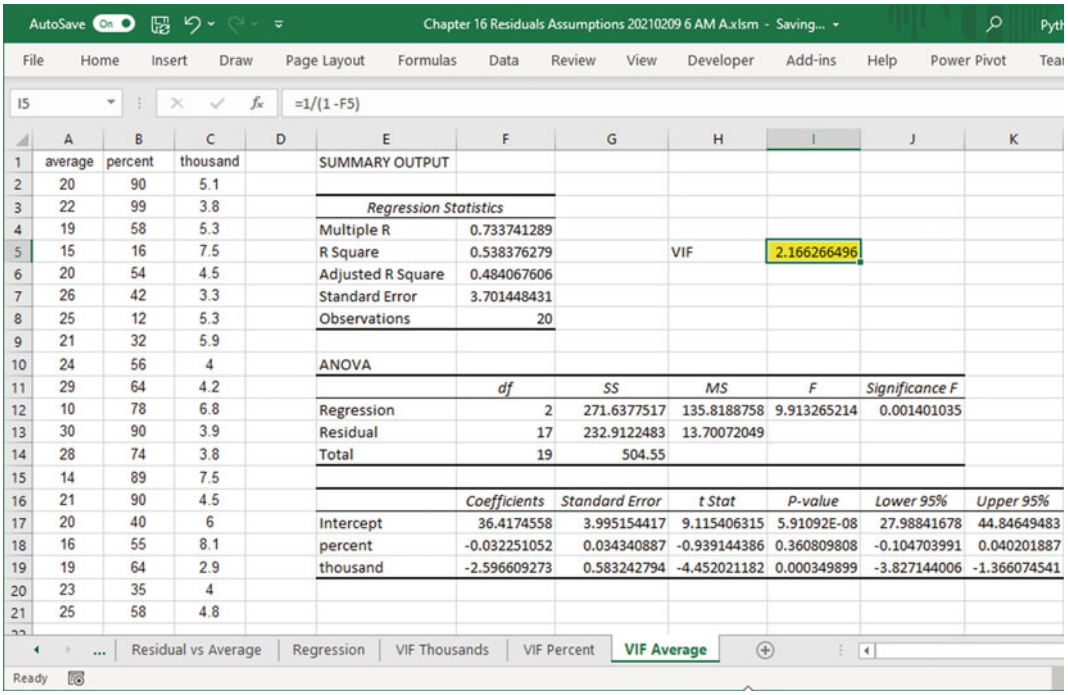
	A	B	C	D	E	F	G	H	I	J	K
1	thousand	percent	average		SUMMARY OUTPUT						
2	5.1	90	20								
3	3.8	99	22		*Regression Statistics*						
4	5.3	58	19		Multiple R	0.745416546					
5	7.5	16	15		R Square	0.555645827		VIF	2.250457094		
6	4.5	54	20		Adjusted R Square	0.503368866					
7	3.3	42	26		Standard Error	1.045869563					
8	5.3	12	25		Observations	20					
9	5.9	32	21								
10	4	56	24		ANOVA						
11	4.2	64	29			*df*	*SS*	*MS*	*F*	*Significance F*	
12	6.8	78	10		Regression	2	23.25266658	11.62633329	10.6288853	0.001013208	
13	3.9	90	30		Residual	17	18.59533342	1.093843142			
14	3.8	74	28		Total	19	41.848				
15	7.5	89	14								
16	4.5	90	21			*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
17	6	40	20		Intercept	10.20095592	1.175436738	8.678438901	1.18437E-07	7.721001175	12.68091066
18	8.1	55	16		percent	-0.011954981	0.009520021	-1.255772537	0.226184323	-0.03204047	0.008130508
19	2.9	64	19		average	-0.207309043	0.046565152	-4.452021182	0.000349899	-0.305552925	-0.10906516
20	4	35	23								
21	4.8	58	25								

The VIF for the "thousands" independent variable is 1.09. The calculation to get this number is shown below.



	A	B	C	D	E	F	G	H	I	J	K	L
1	percent	thousand	average		SUMMARY OUTPUT							
2	90	5.1	20									
3	99	3.8	22		*Regression Statistics*							
4	58	5.3	19		Multiple R	0.291613121						
5	16	7.5	15		R Square	0.085038212		VIF	1.092941819			
6	54	4.5	20		Adjusted R Square	-0.022604351						
7	42	3.3	26		Standard Error	25.48896731						
8	12	5.3	25		Observations	20						
9	32	5.9	21									
10	56	4	24		ANOVA							
11	64	4.2	29			*df*	*SS*	*MS*	*F*	*Significance F*		
12	78	6.8	10		Regression	2	1026.51327	513.256635	0.790005458	0.469813109		
13	90	3.9	30		Residual	17	11044.68673	649.6874547				
14	74	3.8	28		Total	19	12071.2					
15	89	7.5	14									
16	90	4.5	21			*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	
17	40	6	20		Intercept	128.3807838	59.04892229	2.174142708	0.044107912	3.798447704	252.9631199	
18	55	8.1	16		thousand	-7.100653493	5.654410557	-1.255772537	0.226184323	-19.03041697	4.829109984	
19	64	2.9	19		average	-1.529343191	1.628443095	-0.939144386	0.360809808	-4.9650578	1.906371418	
20	35	4	23									
21	58	4.8	25									

The VIF for the "thousands" independent variable is 2.166. The calculation to get this number is shown below.



AutoSave On — Chapter 16 Residuals Assumptions 20210209 6 AM A.xlsm - Saving...

File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins Help Power Pivot

I5 =1/(1-F5)

	A	B	C	D	E	F	G	H	I	J	K
1	average	percent	thousand		SUMMARY OUTPUT						
2	20	90	5.1								
3	22	99	3.8		*Regression Statistics*						
4	19	58	5.3		Multiple R	0.733741289					
5	15	16	7.5		R Square	0.538376279		VIF	2.166266496		
6	20	54	4.5		Adjusted R Square	0.484067606					
7	26	42	3.3		Standard Error	3.701448431					
8	25	12	5.3		Observations	20					
9	21	32	5.9								
10	24	56	4		ANOVA						
11	29	64	4.2			*df*	*SS*	*MS*	*F*	*Significance F*	
12	10	78	6.8		Regression	2	271.6377517	135.8188758	9.913265214	0.001401035	
13	30	90	3.9		Residual	17	232.9122483	13.70072049			
14	28	74	3.8		Total	19	504.55				
15	14	89	7.5								
16	21	90	4.5			*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
17	20	40	6		Intercept	36.4174558	3.995154417	9.115406315	5.91092E-08	27.98841678	44.84649483
18	16	55	8.1		percent	-0.032251052	0.034340887	-0.939144386	0.360809808	-0.104703991	0.040201887
19	19	64	2.9		thousand	-2.596609273	0.583242794	-4.452021182	0.000349899	-3.827144006	-1.366074541
20	23	35	4								
21	25	58	4.8								

... Residual vs Average | Regression | VIF Thousands | VIF Percent | VIF Average

Ready

Based on the VIF calculations for our example, we can conclude that there is no problem with each of our independent variables correlating with the other independent variables as a group.

16.12 Summary

In this chapter, we analyzed the six basic assumptions of linear regression. Analyzing them is important in every regression situation because when we make inferences about the regressed equation, we are assuming that the six basic assumptions are not materially violated. If they are violated, and we are not aware of that, then our inferences could be seriously flawed.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O'Reilly Media, 2017.

Nonparametric Statistics

17

17.1 Introduction

There are three types of statistical data. The three types of statistical data are numerical, categorical, and ordinal. Numerical data is used for measurement, for example, the height of a person and miles to New York City. You can make mathematical operations on numerical data, and the resulting number has meaning. Categorical data represents characteristics, for example, male or female and true or false. Ordinal data has a ranking. For example, an experience could be poor, fair, and excellent. In this chapter, we will do statistical tests on ordinal data.

In most of the previous chapters, we have been interested in estimating distribution parameters like the mean and variance and have assumed that we knew which distribution we were dealing with. In nonparametric statistics, these objectives and assumptions have been changed or cannot be achieved. Now we will be doing the test without making restrictive distribution assumptions about the parameters of the populations.

In general, relative rankings are used to test whether the probability distributions are equal or not. We will first look at a test called the Wilcoxon rank-sum test, also known as the Mann–Whitney U test.

This chapter is broken down into the following sections. In Sect. 17.2 we discuss the Mann–Whitney U Test, and in Sect. 17.3 we discuss the Kruskal–Wallis Test. In Sect. 17.4 we discuss Spearman's Rank Correlation Test. In Sect. 17.5 we use Python to test the randomness of the rate of return of JNJ, and in Sect. 17.6 we use Python to test the randomness of the rate of return of MSFT. Finally, in Sect. 17.7 we summarize the chapter.

17.2 Mann–Whitney *U* Test

A Mann–Whitney U test is a nonparametric test of independent samples from two populations when we cannot assume that the sample is from a population that has a normal distribution.

Example 17.1 The table below shows the research and development expenditures of 15 companies in each of the two major industries, A and B. Use the Mann–Whitney U test to see if the two industries' expenditures are equal. ◀

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_17

R&D Expenditures
Of
Two Major Industries
(in millions of dollars)

#	Industry A	Industry B
1	40	54
2	41	52
3	43	69
4	46	70
5	47	71
6	53	72
7	55	73
8	56	76
9	61	77
10	63	78
11	64	82
12	68	83
13	79	84
14	80	88
15	85	89

In previous chapters, we used the *t*-test for problems like this, but we had to assume that we were sampling from a normal distribution. If we assume that this keyboard problem does not follow a normal distribution, we cannot use a *t*-test. We will use the Mann–Whitney *U* nonparametric test instead because it does not assume that the sample will come from a normal distribution.

Our null and alternative hypotheses would be

H_0: The R&D expenditures of the two industries are equal

H_1: The R&D expenditures of the two industries are not equal.

The following is the Mann–Whitney *U* test statistic,

$$U = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$

where n_1 is the number of observations in sample 1 and n_2 is the number of observations in sample 2 and R_1 is the sum of the rankings in sample 1, and R_2 is the sum of the rankings in sample 2.

It can be shown that the sample distribution of *U* has the following mean and standard deviation.

$$\mu_U = \sqrt{\frac{(n_1 n_2)}{2}}$$

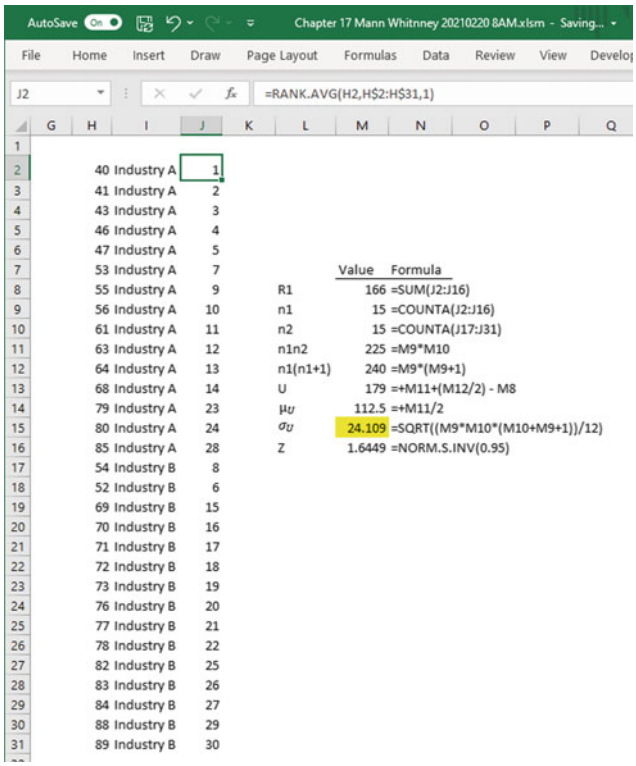
$$\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

Moreover, when the numbers of observations in both samples n_1 and n_2 are in excess of approximately 10 observations, the same distributions of the ranking dollar value of expenditure approach a normal distribution.

17.2.1 Calculation in Microsoft Excel

The following are the steps on how to calculate the Mann–Whitney *U* test statistics in Excel.

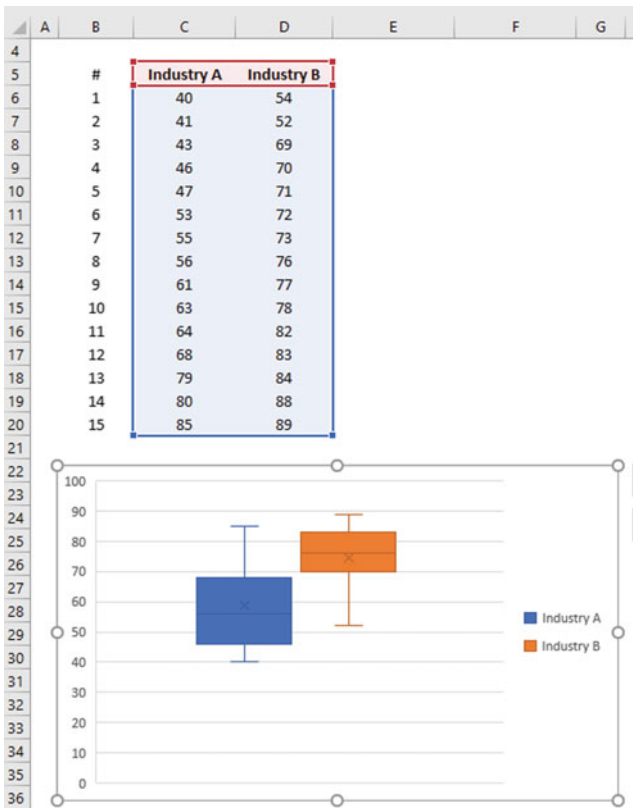
The first thing we will need to do is to combine all of the data into a single column and rank each number using the Excel function *RANK.AVG*.



Cell M15 is the calculated U statistics. The Z-value for a 5% level of significance for a two-tailed test is shown in cell M16. Because the calculated Z-value exceeds the critical value, we can reject the null hypothesis that the sample observations were drawn from the same population.

17.2.1.1 Boxplot

Below are the boxplots of industry A and industry B. The boxplots support our hypothesis of rejecting the null hypothesis that the sample observations were drawn from the same population.



17.2.2 Calculation in *R*

In this section, we will calculate the Mann–Whitney U test statistics in *R*. Below is the *R* code to calculate the Mann–Whitney U test statistics and the result from the code.

```
#Feb-20-2021 10AM
```{r}
IndustryA <- c(40,41,43,46,47,53,55,56,61,63,64,68,79,80,85)
IndustryB <- c(54,52,69,70,71,72,73,76,77,78,82,83,84,88,89)
wilcox.test(IndustryA, IndustryB)
```
```

```
        Wilcoxon rank sum exact test

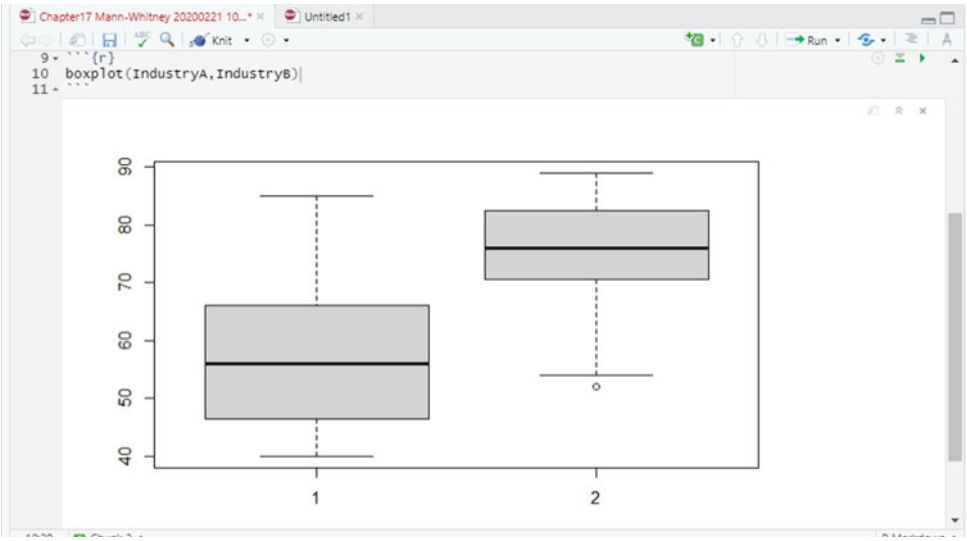
data:  IndustryA and IndustryB
W = 46, p-value = 0.00494
alternative hypothesis: true location shift is not equal to 0
```

R reports a Mann–Whitney statistic as a W. W is

$$W = R_1 - \frac{n_1(n_1+1)}{2}$$

The p-value is less than 0.05, which means we can reject the null hypothesis that the sample observations were drawn from the same population.

The following is the *R* code to generate the boxplot for Industry A and Industry B.



17.3 Kruskal–Wallis Test

The Kruskal–Wallis test is a one-factor analysis of variance by ranks. It is a nonparametric test that represents a generalization of the two-sample Mann–Whitney U test to situations where more than two populations are involved. Unlike one-factor analyses of variance, the Kruskal–Wallis test makes no assumptions about the population distributions.

Example 17.2 Assume that simple samples of executive vice presidents in a certain industry were drawn from firms classified into three size categories (large, medium, and small). After being assured of the confidentiality of their replies, the 20 executives were asked to rate the overall quality of their Board of Director's performance in setting general corporate policy during the past 3-year period on a scale from 0 to 100, with 0 denoting the lowest rating and 100 the highest. The scores, classified by the size of the firm, and the rankings of the pooled sample scores are shown below. ◀

Large	Medium	Small
79	69	83
96	78	66
86	85	51
88	62	94
76	63	71
91	73	61
81		74

The following is the Kruskal–Wallis test statistics, H.

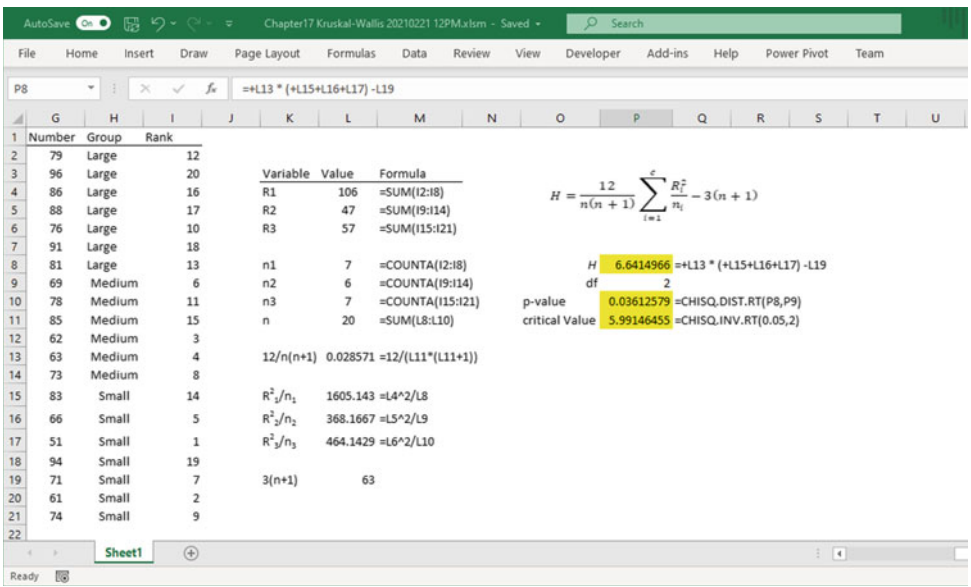
$$H = \frac{12}{n(n+1)} \sum_{i=1}^{c} \frac{R_i^2}{n_i} - 3(n+1)$$

It can be shown that the sample distribution of H statistic is approximately the same as the chi-square distribution with the number of samples minus 1 degree of freedom.

17.3.1 Calculation in Microsoft Excel

The following are the steps on how to calculate the Kruskal–Wallis H test statistics in Excel.

The first thing we will need to do is to combine all of the data into a single column and rank each number using the Excel function *RANK.AVG*.

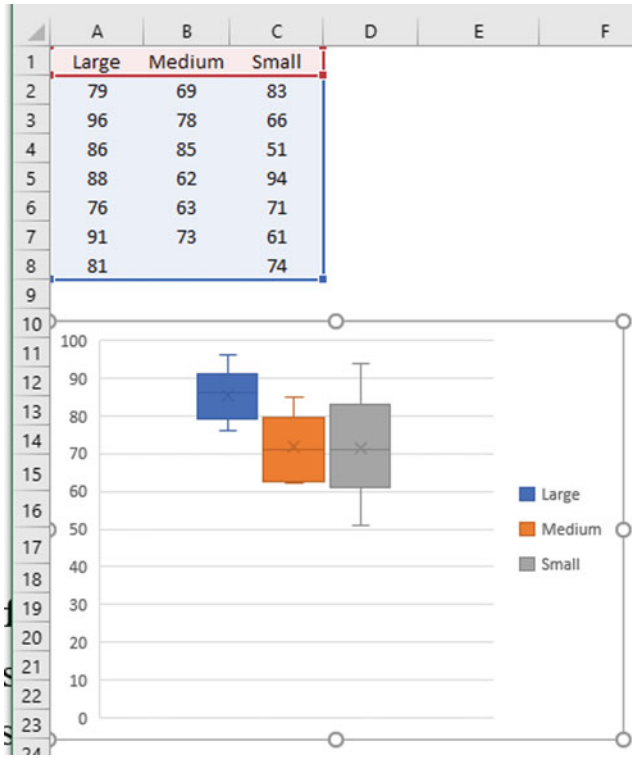


The H statistic is shown in cell O4. The calculations for the H statistic are shown in cells L4:L19. The formulas for column L are shown in column M. The critical value of 5.99 is calculated in cell P11. The p-value of 0.0361 is calculated in cell P10. The calculated H value is calculated in cell P8.

Because the H equals 6.64 is greater than the critical value of 5.991, we reject the null hypothesis of identically distributed populations. Therefore, we conclude that there are significant differences in the firm's size in the scores assigned by these 3 samples of executive vice presidents.

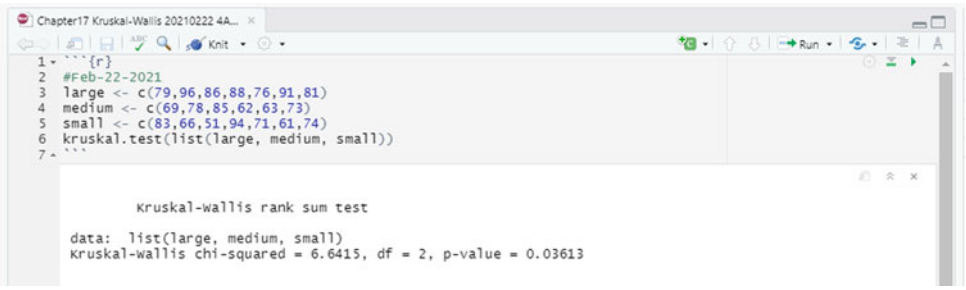
17.3.1.1 Boxplot

Below are the boxplots of industry A and industry B. The boxplots support our hypothesis of rejecting the null hypothesis that the sample observations were drawn from the same population.



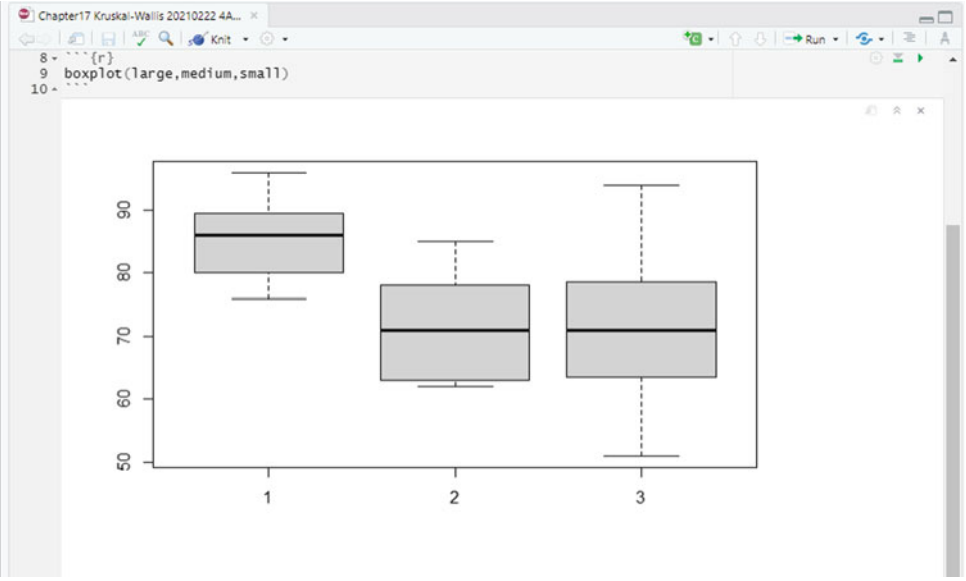
17.3.2 Calculation in *R*

In this section, we will calculate the Kruskal–Wallis *H* test statistics in *R*. Below is the *R* code to calculate the Kruskal–Wallis *H* test statistics and the result from the code.



R reported a chi-squared number of 6.6415. This number is the same *H* test statistics number we calculated above in the Excel section.

The following is the *R* code to generate the boxplot for the large, medium, and small industries.

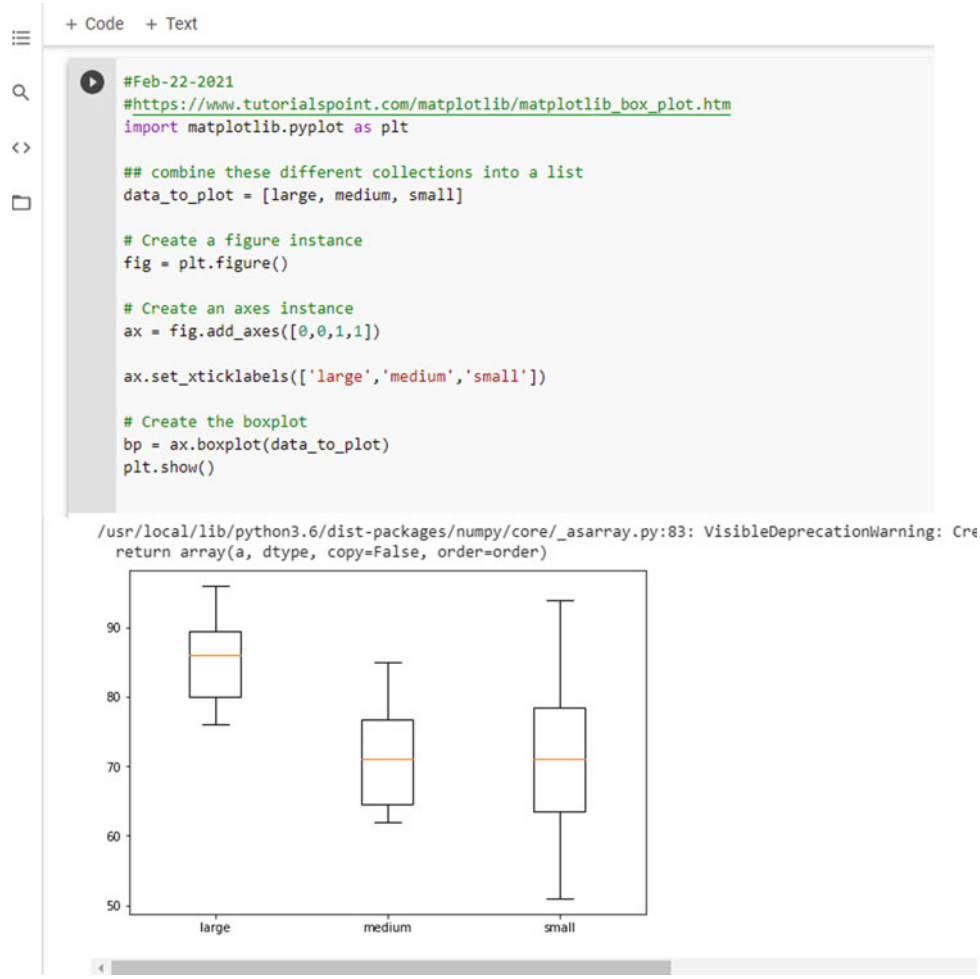


17.3.3 Calculation in Python

In this section, we will calculate the Kruskal–Wallis *H* test statistics in Python. Below is the Python code to calculate the Kruskal–Wallis *H* test statistics and the result from the code.

R reported a chi-squared number of 6.6415. This number is the same *H* test statistics number we calculated above in the Excel section.

The following is the *R* code to generate the boxplot for the large, medium, and small industries.



17.4 Spearman's Rank Correlation Test

Spearman's rank correlation test is a nonparametric method of correlation designed to measure the strength of association between two sets of *ranked* data. This is in contrast to the other more well-known correlation test, the Pearson correlation test, which does not use ranked data.

The following is the formula for *Spearman's Rank Correlation Coefficient*.

$$r_s = 1 - \frac{6\sum d^2}{n(n^2 - 1)}$$

where

d = difference between the ranks for the paired observations

n = number of paired observations.

The following is the formula for the *Pearson Correlation Coefficient*.

$$r = \frac{\text{sample covariance}}{(\text{sample standard deviation of } x)(\text{sample standard deviation of } y)}$$

$$= \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2]^{1/2}[\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y})^2]^{1/2}} = \frac{s_{xy}}{s_x s_y}$$

Comparing the two correlation formulas, we can see that the *Pearson Correlation Coefficient* is much more complicated and takes much more time and effort to calculate.

The calculations of the rank correlation coefficients for the two extreme cases are shown below. In the first table, there is a perfect direct correlation in the ranks; that is, the student who ranks highest in mathematical ability is also best in physics. In the second table, there is a perfect inverse correlation; that is, the student who ranks highest in mathematics is worst in physics.

Perfect Direct Correlation

Student	Rank in Mathematics Ability, X	Rank in Physics Ability,Y	Difference in Ranks, d = X - Y	$d^2 = (X-Y)^2$
A	1	1	0	0
B	2	2	0	0
C	3	3	0	0
D	4	4	0	0
E	5	5	0	0
F	6	6	0	0
G	7	7	0	0
H	8	8	0	0
I	9	9	0	0
J	10	10	0	0

Perfect Inverse Correlation

Student	Rank in Mathematics Ability, X	Rank in Physics Ability,Y	Difference in Ranks, d = X - Y	$d^2 = (X-Y)^2$
A	1	10	-9	81
B	2	9	-7	49
C	3	8	-5	25
D	4	7	-3	9
E	5	6	-1	1
F	6	5	1	1
G	7	4	3	9
H	8	3	5	25
I	9	2	7	49
J	10	1	9	81
				330

In the case of perfect correlation between the ranks, $r_s = 1$; in perfect inverse correlation, $r_s = -1$. An r_s. value of zero indicates no correlation between rankings.

$$r_s = 1 - \frac{6\Sigma d^2}{n(n^2-1)} = 1 - \frac{6(0)}{10(10^2-1)} = 1$$

$$r_s = 1 - \frac{6\Sigma d^2}{n(n^2-1)} = 1 - \frac{6(330)}{10(10^2-1)} = 1$$

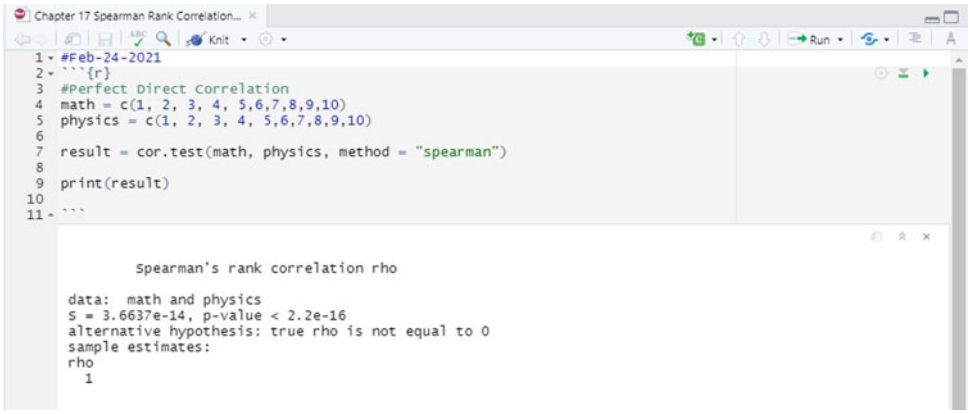
The following is the test statistics for Spearman's Rank Correlation.

$$t = \frac{r_s}{\sqrt{\frac{\left(1-r_s^2\right)}{n-2}}}$$

This test statistic has a t distribution with $(n - 2)$ degrees of freedom. If $r_s = 0.90$, then

$$t = \frac{0.90}{\sqrt{\frac{(1-0.81)}{(10-2)}}} = 5.84$$

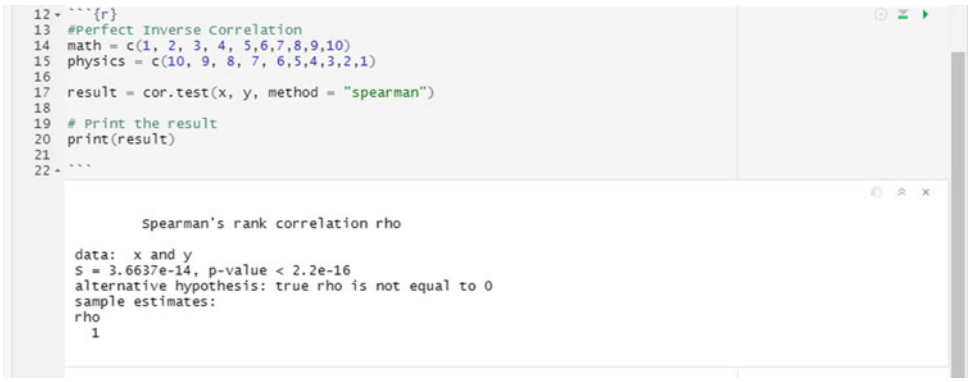
Assuming we are using a two-tailed test of the null hypothesis of zero correlation in the ranked data of the population, the critical t as a 5% level of significance with 8 degrees of freedom is equal to 2.306. You can use the Excel *T.INV* function to get this result.

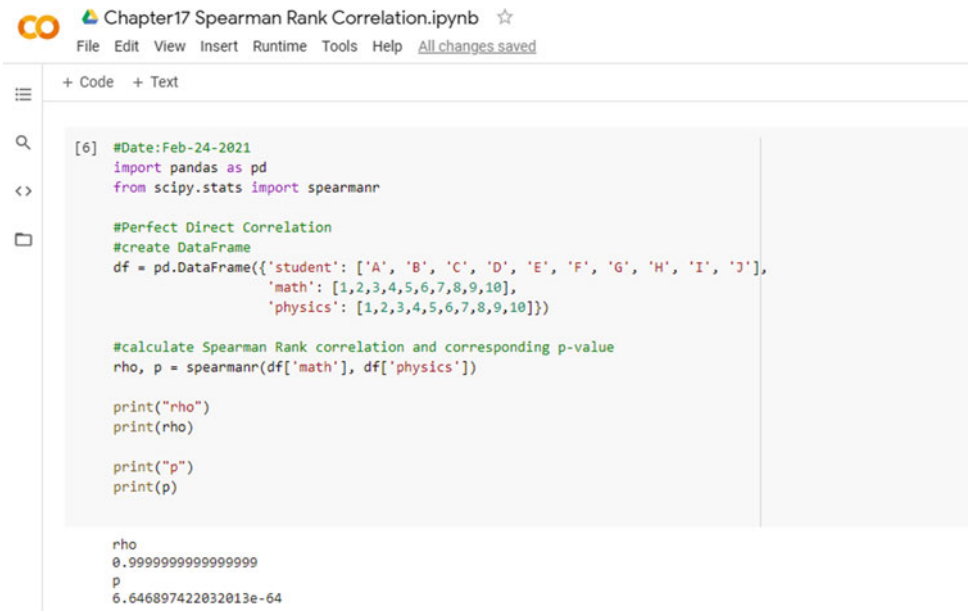


We reject the hypothesis of no rank correlation and conclude that a positive linear relationship exists between rank in mathematics learning ability and in physics learning ability.

17.4.1 Calculation in *R*

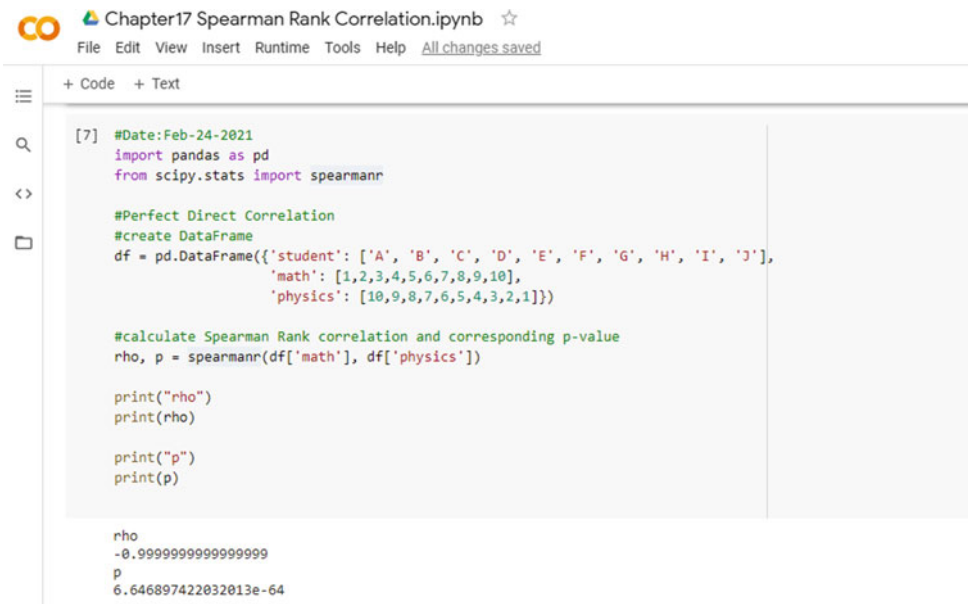
In this section, we will calculate Spearman's Rank Correlation test statistics in *R*. Below is the *R* code to calculate Spearman's Rank Correlation test statistics and the result from the code.





17.4.2 Calculation in Python

In this section, we will calculate Spearman's Rank Correlation test statistics in Python. Below is the Python code to calculate Spearman's Rank Correlation test statistics and the result from the code.



```
[59]  1 pip install requests_html
```

```
[60]  1 pip install yahoo_fin
```

```
[61]  1 pip install requests
```

```
[62]  1 pip install bs4
```

```
Requirement already satisfied: bs4 in /usr/local/lib/python3.7/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from bs4) (4.6.3)
```

```
[84]  1 #Date: Apr-11-2021
      2 import numpy as np
      3 import scipy.stats as st
      4 from yahoo_fin.stock_info import*
      5 from tabulate import tabulate
      6 import requests
      7 from bs4 import BeautifulSoup
      8 import datetime as dt#get current date
      9 from google.colab import files
```

17.5 Using Python to Test the Randomness of the Rate of Return of JNJ

In this section, we will perform a nonparametric run test to see if the 5-year daily rate of return for JNJ is random or not.

```
[85]  1 def run_test(data, v, alpha = 0.05):
      2   #get positive ret and neg ret
      3   mask = data>v
      4   p = mask == True
      5   n = mask == False
      6   change_sin = np.logical_xor(p[:-1],p[1:])
      7   sum_run = sum(change_sin)+1
      8   cum_sum = np.cumsum(change_sin)
      9   cum_sum = np.insert(cum_sum,0,0)
     10   #sum of positive and negative num
     11   n_p = sum(p)
     12   n_n = sum(n)
     13
     14   tmp = 2 * n_p * n_n
     15   tmps = n_p + n_n
     16
     17   # Expected value
     18   r_hat = np.float64(tmp) / tmps + 1
     19   # Variance
     20   s_r_squared = (tmp*(tmp - tmps)) / (tmps*tmps*(tmps-1))
     21   # Standard deviation
     22   s_r =  np.sqrt(s_r_squared)
     23   # Test score
     24   z = (sum_run - r_hat) / s_r
     25   # Get normal table
     26   z_alpha = st.norm.ppf(1-alpha)
     27   # Check hypothesis
     28   return z, z_alpha,cum_sum,n_p,n_n,sum_run
```

```
[86]  1 def html_beautifulsoup(url):
      2     response = requests.get(url)
      3     html = response.content
      4
      5     #beautifulsoup html will not look like soure
      6     #review beautifulsoup html to use find and findall
      7     soup = BeautifulSoup(html,"lxml")
      8     return soup
      9
     10 #soup_html: soup html is not same as source html
     11 #return description for a ticker
     12 def yahoo_description(soup_html):
     13     element = soup_html.find('h1',class_='D(ib) Fz(18px)')
     14     return element.text
```

```
[87]  1 def usr_input():
      2   previousday = dt.date.today()-dt.timedelta(days=1)# get the date of previous day
      3   Default_end = previousday.strftime("%Y-%m-%d") # change the format of the default datetime
      4   years_ago = dt.date.today()-dt.timedelta(days=1825)
      5   Default_start = years_ago.strftime("%Y-%m-%d")
      6   Ticker = input('Please enter ticker(Default = JNJ)-->') or 'JNJ'
      7   Start = input('Enter start date(' + 'Default = 5 Years ago, ' + Default_start + ')-->') or Default_start
      8   End = input('Enter end date(Default = Yesterday, ' + Default_end + ')-->') or Default_end
      9
     10   soup_html = html_beautifulsoup("https://finance.yahoo.com/quote/"  + Ticker)
     11   stock_name = yahoo_description(soup_html)
     12
     13   return Ticker,Start,End,stock_name
```

```
[88]  1 Ticker,Start,End,stock_name = usr_input()
      2 temp = get_data(Ticker,start_date = Start,end_date=End,index_as_date=True)['adjclose']
      3 Ret = temp.pct_change().dropna()
      4 z,z_alpha,cum,n_p,n_n,run = run_test(Ret.to_numpy(),0)
      5 Ret = Ret.to_frame()
      6 Ret.columns = ['Return']
      7 Ret.insert(1, "runs", cum, True)

    Please enter ticker(Default = JNJ)-->JNJ
    Enter start date(Default = 5 Years ago, 2016-04-12)-->2016-04-12
    Enter end date(Default = Yesterday, 2021-04-10)-->2021-04-10
```

```
[89]  1 print('Test result for '+stock_name)
      2 print('')
      3 result = {'n1':[n_p],'n2':[n_p],'run':[run],'statistic':[z],'Z-score':[z_alpha]}
      4 result = pd.DataFrame.from_dict(result)
      5 print(tabulate(result,result.columns))
      6 print(' ')
      7
      8 print ('Null Hyposesis: Returns are random')
      9 if z>z_alpha:
     10   print(f"Because our Z-value is greater than {z_alpha}, we are able to reject the null hypothesis at 0.05 level")
     11 else:
     12   print(f"Because our Z-value is less than {z_alpha}, we are not able to reject the null hypothesis at 0.05 level")

    Test result for Johnson & Johnson (JNJ)

          n1    n2    run    statistic    Z-score
    --  ----  ----  -----  -----------  ---------
     0   662   662    644     0.921671    1.64485

    Null Hyposesis: Returns are random
    Because our Z-value is less than 1.6448536269514722, we are not able to reject the null hypothesis at 0.05 level
```

```
    1 pd.set_option('display.max_rows', None)
    2 print(tabulate(Ret,Ret.columns))
```

	Return	runs
2016-04-13 00:00:00	0.00246372	0
2016-04-14 00:00:00	-0.000273259	1
2016-04-15 00:00:00	0.00309545	2
2016-04-18 00:00:00	0.00680703	2
2016-04-19 00:00:00	0.0157757	2
2016-04-20 00:00:00	0.0079873	2
2016-04-21 00:00:00	-8.80087e-05	3
2016-04-22 00:00:00	-0.00220149	3
2016-04-25 00:00:00	0.00167693	4
2016-04-26 00:00:00	-0.00528585	5
2016-04-27 00:00:00	-0.00124013	5
2016-04-28 00:00:00	-0.00257151	5

From the run test, we cannot reject the null hypothesis that John and John's daily 5-year returns are random.

```
[90] 2021-03-15 00:00:00   0.00513779      634
     2021-03-16 00:00:00   0.00592194      634
     2021-03-17 00:00:00  -0.00371811      635
     2021-03-18 00:00:00  -0.00186604      635
     2021-03-19 00:00:00  -0.00267968      635
     2021-03-22 00:00:00   0.00287432      636
     2021-03-23 00:00:00  -0.000934541     637
     2021-03-24 00:00:00   0.0097287       638
     2021-03-25 00:00:00   0.000370561     638
     2021-03-26 00:00:00   0.0182749       638
     2021-03-29 00:00:00   0.00673013      638
     2021-03-30 00:00:00  -0.00620332      639
     2021-03-31 00:00:00  -0.00399969      639
     2021-04-01 00:00:00  -0.00924858      639
     2021-04-05 00:00:00   0.00368477      640
     2021-04-06 00:00:00  -0.000244712     641
     2021-04-07 00:00:00   0.00134648      642
     2021-04-08 00:00:00  -0.00391174      643
     2021-04-09 00:00:00  -0.0105541       643

    1 Ret.to_csv('Return.csv')
    2 files.download('Return.csv')
    3 print ("On the pc the file was downloaded to your 'Downloads' folder.")

    On the pc the file was downloaded to your 'Downloads' folder.
```

```
[92]  1 Ticker,Start,End,stock_name = usr_input()
      2 temp = get_data(Ticker,start_date = Start,end_date=End,index_as_date=True)['adjclose']
      3 Ret = temp.pct_change().dropna()
      4 z,z_alpha,cum,n_p,n_n,run = run_test(Ret.to_numpy(),0)
      5 Ret = Ret.to_frame()
      6 Ret.columns = ['Return']
      7 Ret.insert(1, "runs", cum, True)

     Please enter ticker(Default = JNJ)-->MSFT
     Enter start date(Default = 5 Years ago, 2016-04-12)-->2016-04-12
     Enter end date(Default = Yesterday, 2021-04-10)-->2021-04-10
```

The following Python commands print out the 5-year daily returns and save the returns as a CSV file.

17.6 Using Python to Test the Randomness of the Rate of Return of MSFT

In this section, we will perform a nonparametric run test to see if the 5-year daily rate of return for MSFT is random or not.

```
[93]  1 print('Test result for '+stock_name)
      2 print('')
      3 result = {'n1':[n_p],'n2':[n_p],'run':[run],'statistic':[z],'Z-score':[z_alpha]}
      4 result = pd.DataFrame.from_dict(result)
      5 print(tabulate(result,result.columns))
      6 print(' ')
      7
      8 print ('Null Hyposesis: Returns are random')
      9 if z>z_alpha:
     10   print(f"Because our Z-value is greater than {z_alpha}, we are able to reject the null hypothesis at 0.05 level")
     11 else:
     12   print(f"Because our Z-value is less than {z_alpha}, we are not able to reject the null hypothesis at 0.05 level")

     Test result for Microsoft Corporation (MSFT)

          n1    n2    run    statistic    Z-score
     --  ----  ----  -----  -----------  ---------
      0   700   700    687      3.75255    1.64485

     Null Hyposesis: Returns are random
     Because our Z-value is greater than 1.6448536269514722, we are able to reject the null hypothesis at 0.05 level
```

```
 1 pd.set_option('display.max_rows', None)
 2 print(tabulate(Ret,Ret.columns))

                          Return    runs
-------------------  ------------  ------
2016-04-13 00:00:00   0.0128084         0
2016-04-14 00:00:00   0.00018076        0
2016-04-15 00:00:00   0.00523845        0
2016-04-18 00:00:00   0.0145552         0
2016-04-19 00:00:00  -0.00123978        1
2016-04-20 00:00:00  -0.0141869         1
2016-04-21 00:00:00   0.00341789        2
2016-04-22 00:00:00  -0.0717101         3
2016-04-25 00:00:00   0.00637288        4
2016-04-26 00:00:00  -0.0128573         5
2016-04-27 00:00:00  -0.0097202         5
2016-04-28 00:00:00  -0.0204161         5
2016-04-29 00:00:00  -0.000601256       5
```

From the run test, we can reject the null hypothesis that Microsoft Corporation's daily 5-year returns are random. JNJ's returns are random, while MSFT's returns are not random; This is probably due to the fact that they are from different industries.

The following Python commands print out the 5-year daily returns and save the returns as a CSV file.

```
2020-07-02 00:00:00   0.00762094      576
2020-07-06 00:00:00   0.0215262       576
2020-07-07 00:00:00  -0.0116279       577
2020-07-08 00:00:00   0.0219928       578
2020-07-09 00:00:00   0.00700093      578
2020-07-10 00:00:00  -0.0030329       579
2020-07-13 00:00:00  -0.0308888       579
2020-07-14 00:00:00   0.00618152      580
2020-07-15 00:00:00  -0.00148797      581
2020-07-16 00:00:00  -0.0198038       581
2020-07-17 00:00:00  -0.00510006      581
2020-07-20 00:00:00   0.0429812       582
2020-07-21 00:00:00  -0.0134688       583
2020-07-22 00:00:00   0.0143712       584
```

```
[91]  1 Ret.to_csv('Return.csv')
      2 files.download('Return.csv')
      3 print ("On the pc the file was downloaded to your 'Downloads' folder.")

     On the pc the file was downloaded to your 'Downloads' folder.
```

17.7 Summary

In this chapter, we discussed three statistical tests that do not require the assumption of normality in the distribution of the population. These tests are called nonparametric statistical tests. The three tests were the Mann–Whitney *U* test, the Kruskal–Wallis test, and Spearman's Rank Correlation test.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Time Series: Analysis, Model, and Forecasting 18

18.1 Introduction

In statistics, there are two kinds of data, cross-section data and time-series data. Time-series data are those recorded over time. Cross-section data pertain to a particular time. In this chapter, we will look at specific issues of time-series data.

Time-series data can be broken down into four components:

1. Secular Trend component, T
2. Cyclical component, C
3. Seasonal component, S
4. Irregular component, I.

These four components can be specified as an additive model

$$x = T + C + S + I$$

or as a multiplicative model

$$x = T * C * S * I$$

Time-series data can fluctuate wildly, mostly because of these four components. In the next section, we discuss how we can use the moving-average method to reduce the effects of fluctuations associated with seasonality and irregularity.

This chapter is broken down into the following sections. Section 18.2 discusses moving averages, Sect. 18.3 discusses the linear trend, and Sect. 18.4 discusses exponential smoothing. Finally, in Sect. 18.5, we summarize the chapter.

18.2 Moving Averages

We will use 2011–2020 quarterly earnings per share data of Johnson & Johnson (J&J) to show how moving averages can be obtained.

The formula to calculate a 4-point moving average is

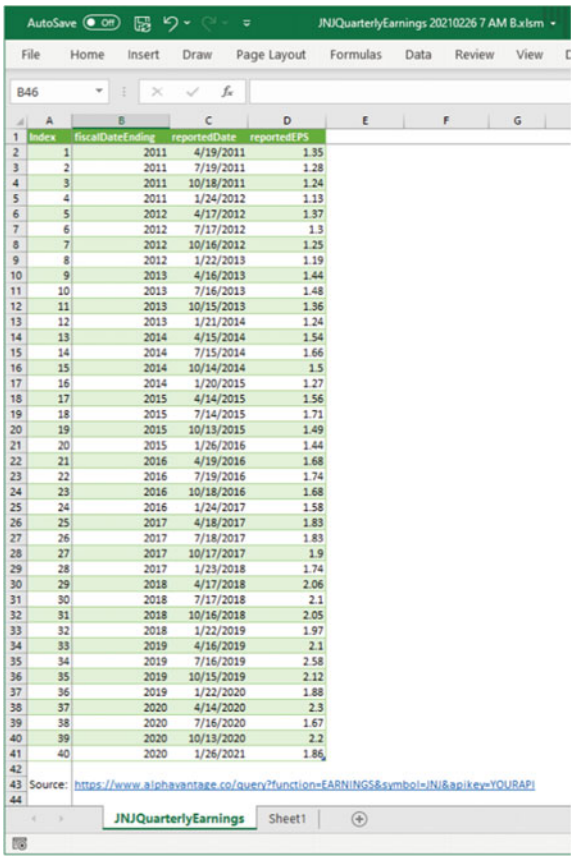
$$M_t = \frac{Y_{t-3} + Y_{t-2} + Y_{t-1} + Y}{4}$$

We start our calculation at period 4. The calculation for period 4 is

$$M_4 = \frac{Y_1 + Y_2 + Y_3 + Y_4}{4} = \frac{1.35 + 1.28 + 1.24 + 1.13}{4} = 1.25$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_18



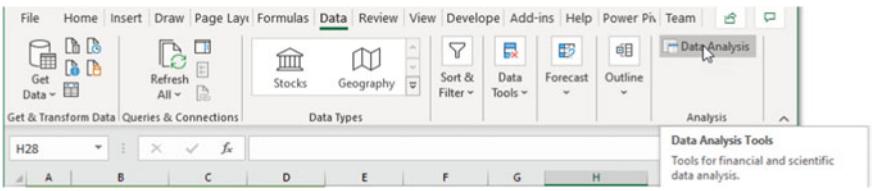
Index	fiscalDateEnding	reportedDate	reportedEPS
1	2011	4/19/2011	1.35
2	2011	7/19/2011	1.28
3	2011	10/18/2011	1.24
4	2011	1/24/2012	1.13
5	2012	4/17/2012	1.37
6	2012	7/17/2012	1.3
7	2012	10/16/2012	1.25
8	2012	1/22/2013	1.19
9	2013	4/16/2013	1.44
10	2013	7/16/2013	1.48
11	2013	10/15/2013	1.36
12	2013	1/21/2014	1.24
13	2014	4/15/2014	1.54
14	2014	7/15/2014	1.66
15	2014	10/14/2014	1.5
16	2014	1/20/2015	1.27
17	2015	4/14/2015	1.56
18	2015	7/14/2015	1.71
19	2015	10/13/2015	1.49
20	2015	1/26/2016	1.44
21	2016	4/19/2016	1.68
22	2016	7/19/2016	1.74
23	2016	10/18/2016	1.68
24	2016	1/24/2017	1.58
25	2017	4/18/2017	1.83
26	2017	7/18/2017	1.83
27	2017	10/17/2017	1.9
28	2017	1/23/2018	1.74
29	2018	4/17/2018	2.06
30	2018	7/17/2018	2.1
31	2018	10/16/2018	2.05
32	2018	1/22/2019	1.97
33	2019	4/16/2019	2.1
34	2019	7/16/2019	2.58
35	2019	10/15/2019	2.12
36	2019	1/22/2020	1.88
37	2020	4/14/2020	2.3
38	2020	7/16/2020	1.67
39	2020	10/13/2020	2.2
40	2020	1/26/2021	1.86

Source: https://www.alphavantage.co/query?function=EARNINGS&symbol=JNJ&apikey=YOURAPI

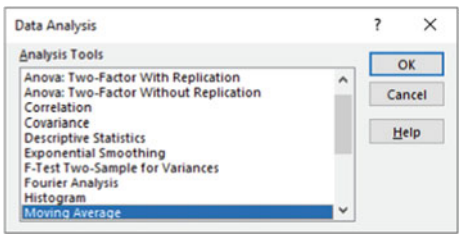
18.2.1 Moving Averages in Excel

18.2.1.1 *Data Analysis*

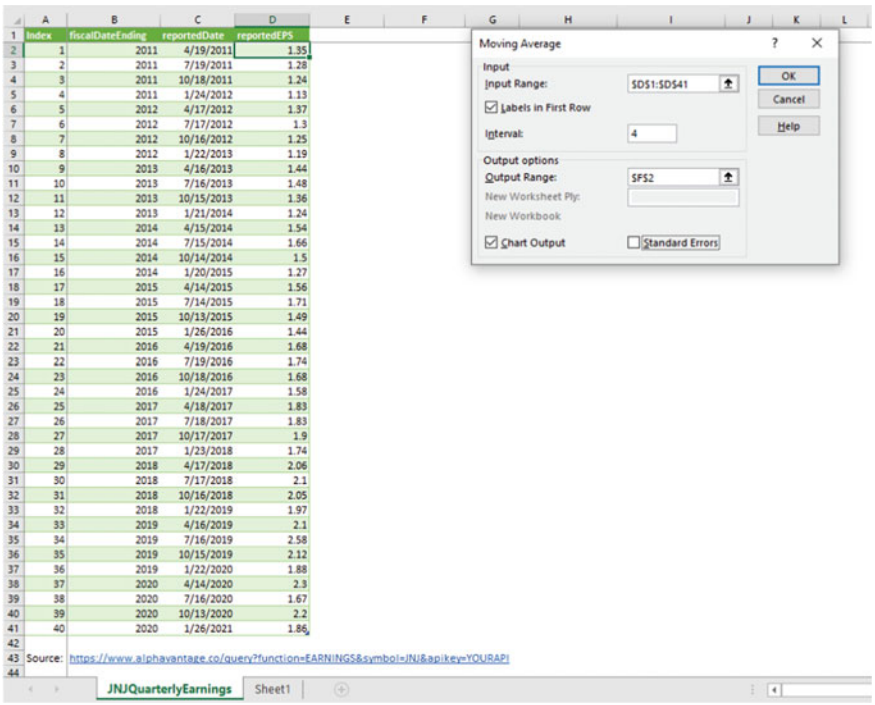
In this section, we will use the *Data Analysis* tool to calculate the moving average and produce a moving average chart.



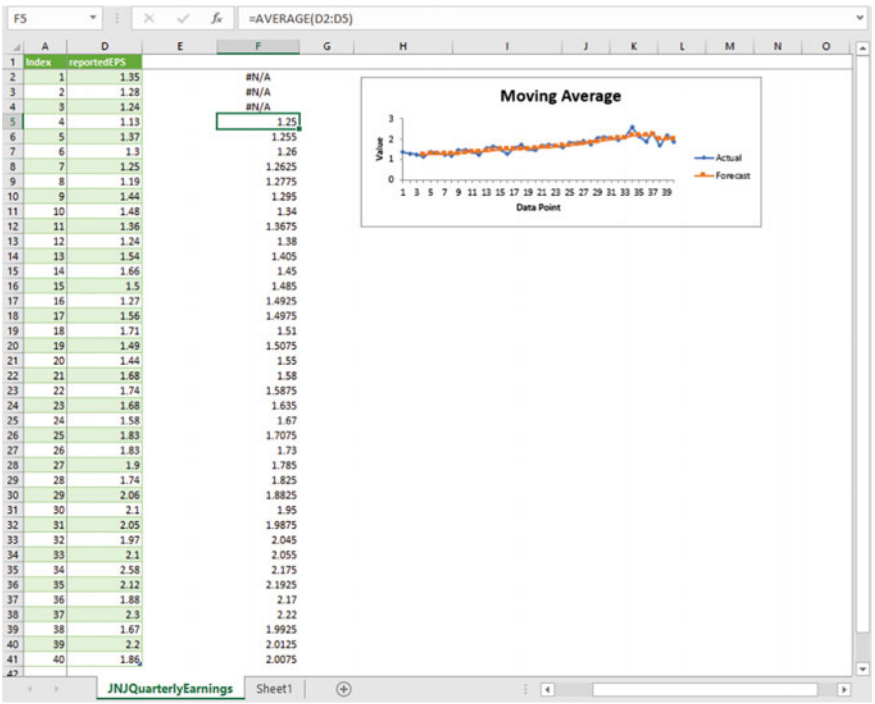
Choose the *Moving Average* option from the *Data Analysis* as shown below.



Fill out the *Moving Average* screen as shown below.



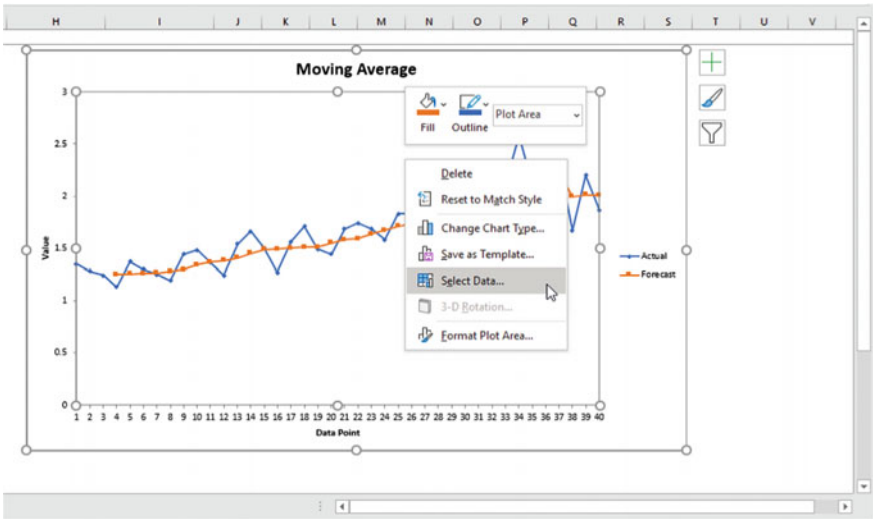
After pressing the *OK* button, the *Moving Average* dialog box is displayed, as shown below.



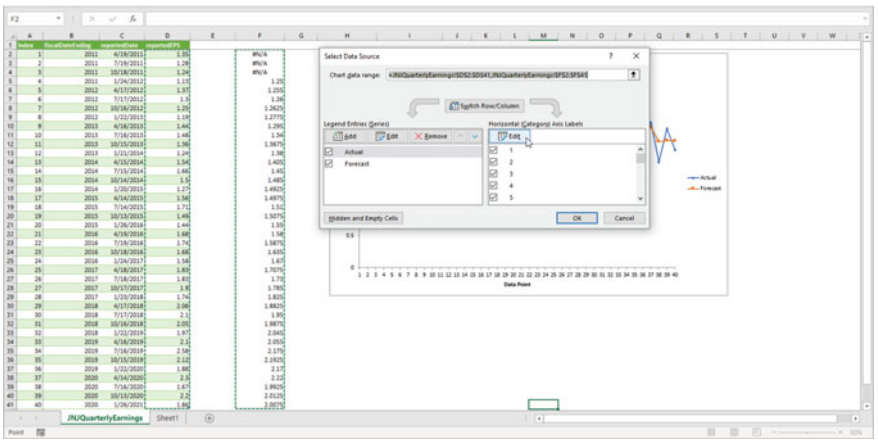
As shown above, the *Moving Average* dialog box puts the moving average calculation in column F and puts the moving average chart at cell H2. The moving average calculations start on period 4, and the *AVERAGE* Excel formula is used.

Do the following to change the x-axis in the chart to dates.

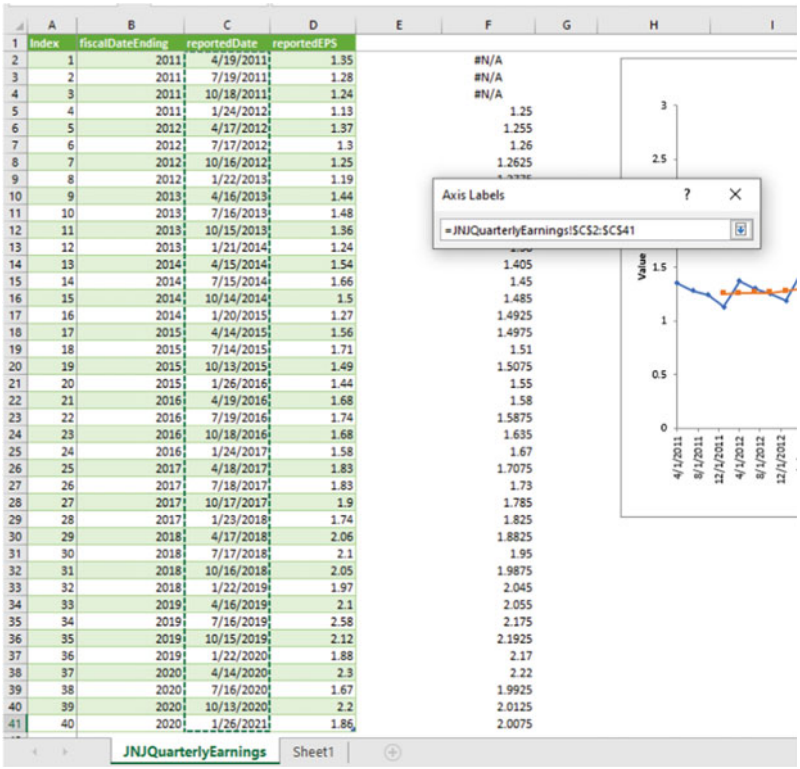
(1) Right-mouse click on the chart and click on *Select Data*



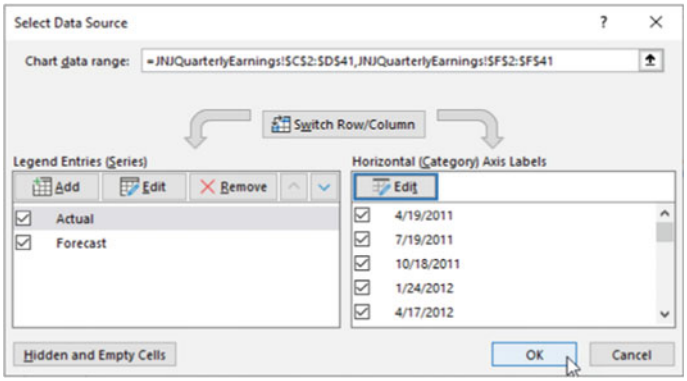
(2) Click on the *Edit* button under the *Horizontal* section.



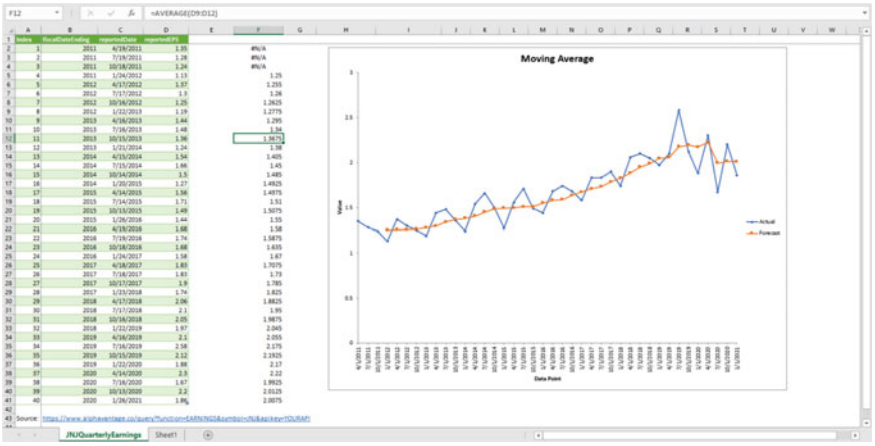
(3) Choose the date column



(4) Click on *OK* button



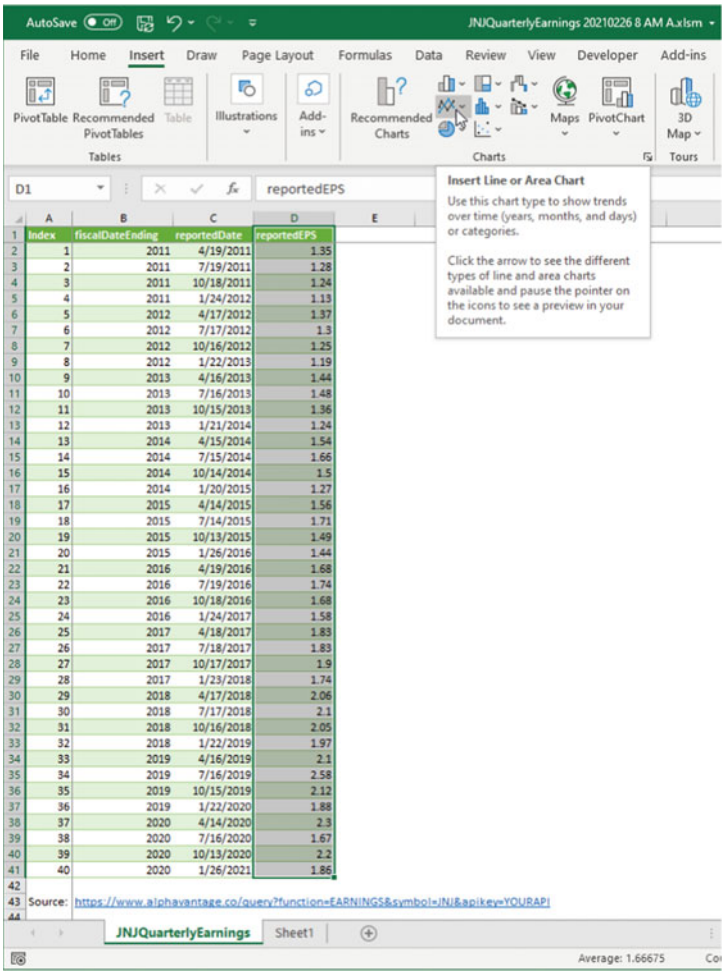
The following shows the resulting *moving average* chart with a date x-axis.



18.2.1.2 Moving Average Trend Chart

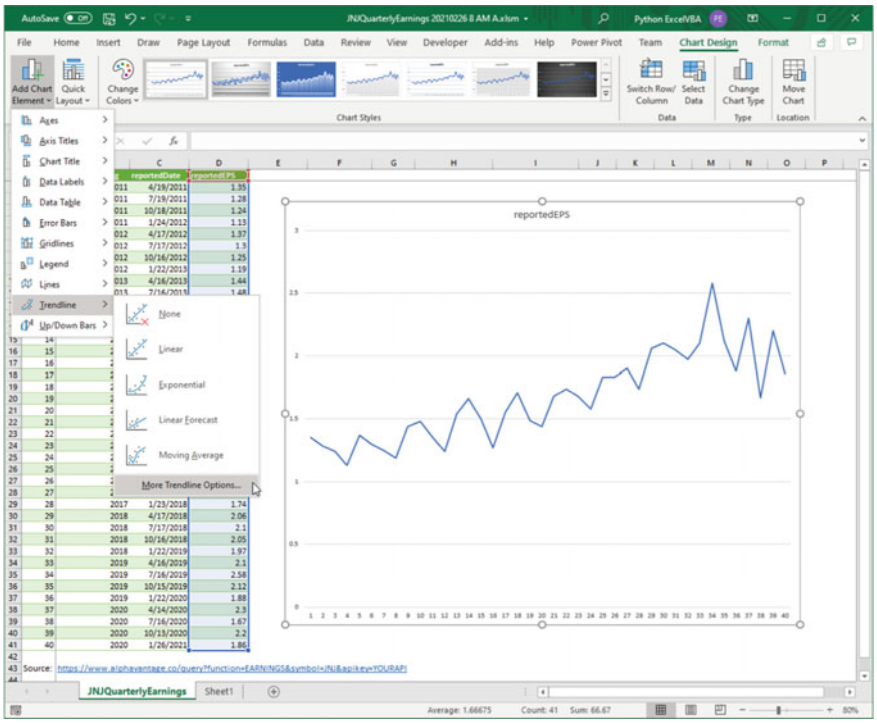
In this section, we will create a *moving average trend chart*.

(1) Select the EPS data and click on *Insert Line or Area Chart* in the *Insert* tab.

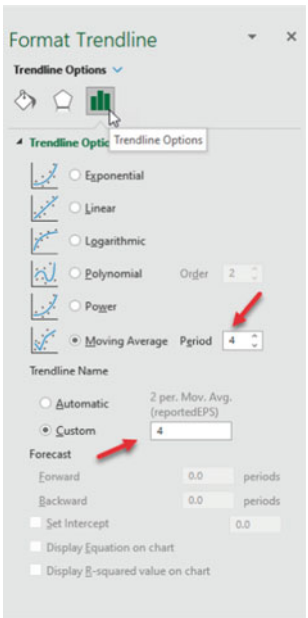


Index	fiscalDateEnding	reportedDate	reportedEPS
1	2011	4/19/2011	1.35
2	2011	7/19/2011	1.28
3	2011	10/18/2011	1.24
4	2011	1/24/2012	1.13
5	2012	4/17/2012	1.37
6	2012	7/17/2012	1.3
7	2012	10/16/2012	1.25
8	2012	1/22/2013	1.19
9	2013	4/16/2013	1.44
10	2013	7/16/2013	1.48
11	2013	10/15/2013	1.36
12	2013	1/21/2014	1.24
13	2014	4/15/2014	1.54
14	2014	7/15/2014	1.66
15	2014	10/14/2014	1.5
16	2014	1/20/2015	1.27
17	2015	4/14/2015	1.56
18	2015	7/14/2015	1.71
19	2015	10/13/2015	1.49
20	2015	1/26/2016	1.44
21	2016	4/19/2016	1.68
22	2016	7/19/2016	1.74
23	2016	10/18/2016	1.68
24	2016	1/24/2017	1.58
25	2017	4/18/2017	1.83
26	2017	7/18/2017	1.83
27	2017	10/17/2017	1.9
28	2017	1/23/2018	1.74
29	2018	4/17/2018	2.06
30	2018	7/17/2018	2.1
31	2018	10/16/2018	2.05
32	2018	1/22/2019	1.97
33	2019	4/16/2019	2.1
34	2019	7/16/2019	2.58
35	2019	10/15/2019	2.12
36	2019	1/22/2020	1.88
37	2020	4/14/2020	2.3
38	2020	7/16/2020	1.67
39	2020	10/13/2020	2.2
40	2020	1/26/2021	1.86

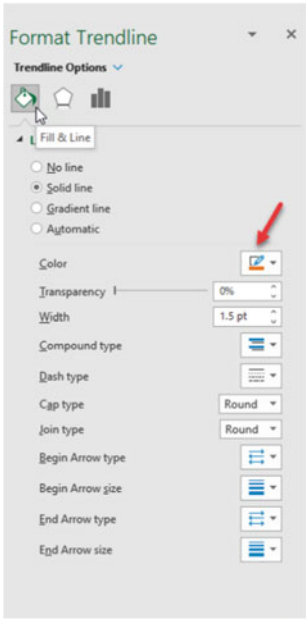
(2) Click on *More Trendline Options* in the *Chart Design* tab.



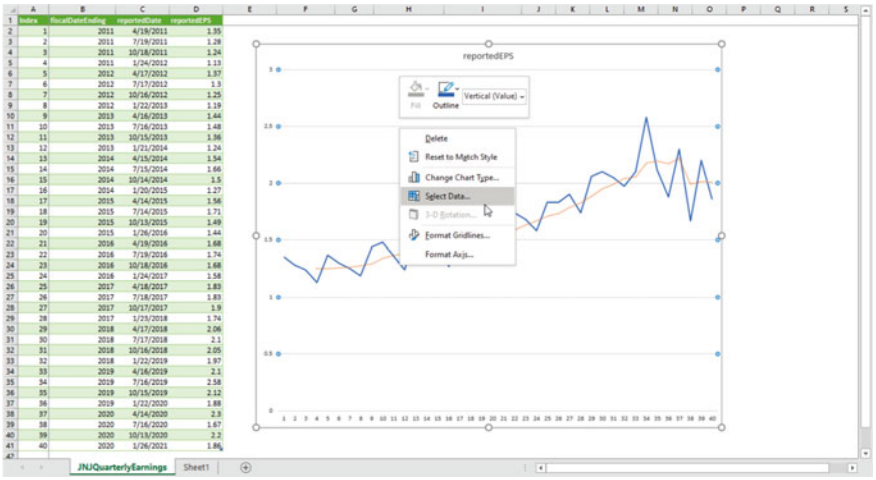
(3) In the *Trendline Options* screen, choose *Moving Average* and set period to 4, and set custom to 4.



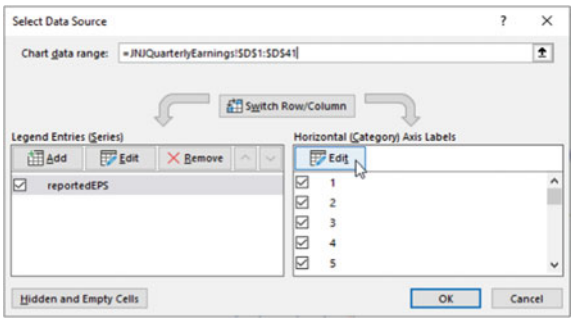
(4) In the *Fill & Line* section set *Color* to orange.



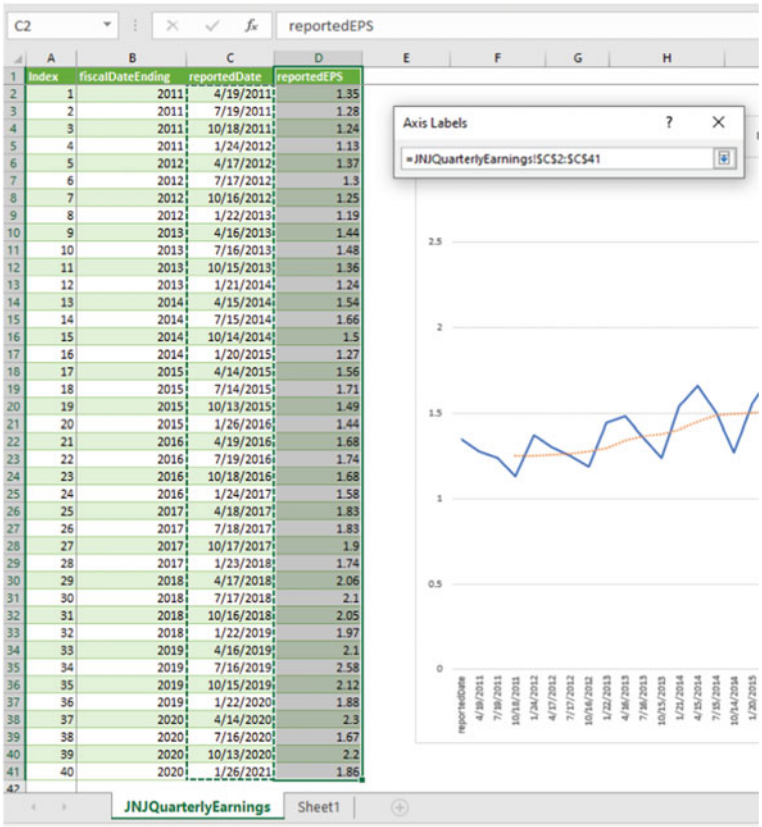
(5) We will now set the x-axis as dates. Right-mouse click on the chart and click on *Select Data*



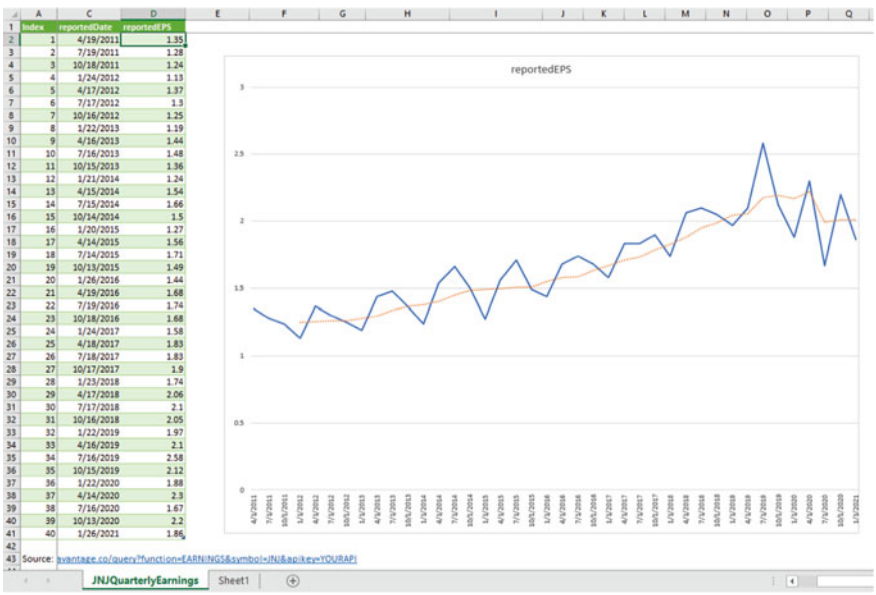
(6) Click on the *Edit* button under the horizontal section.



(7) Select date range

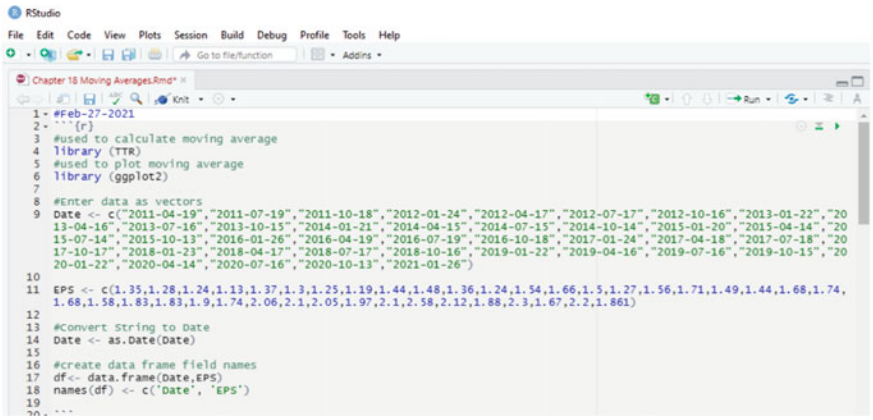


The following is J&J's moving average trend chart.

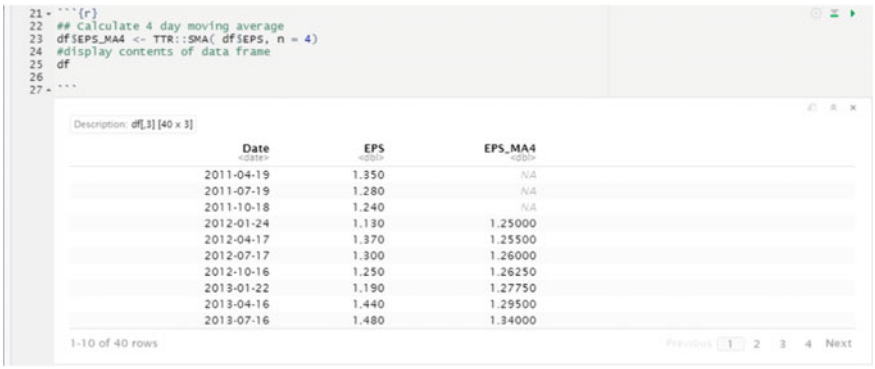


18.2.2 Moving Averages in *R*

In this section, we will calculate and plot Johnson & Johnson's moving average in *R*.



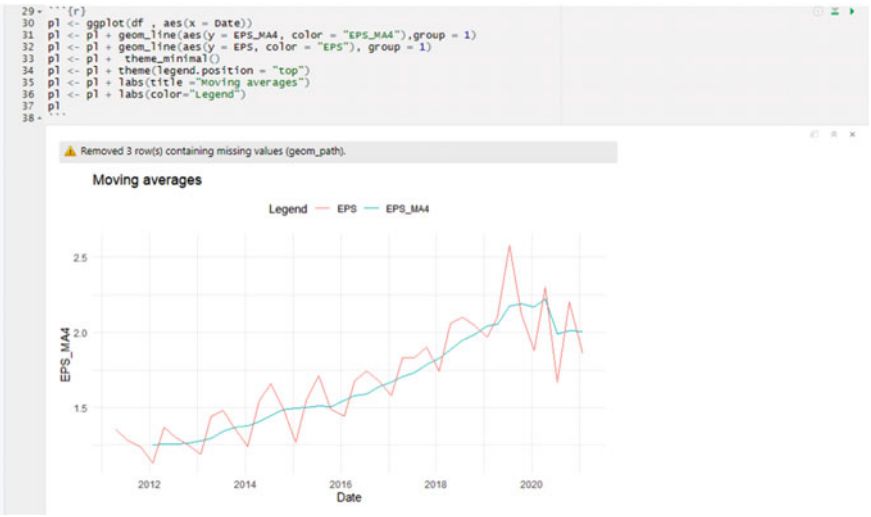
We will need R's TTR and ggplot2 library to calculate and plot the moving average; this is indicated in lines 4 and 6. The dates and EPS are entered as vectors in line 9 and line 11. The TTR library uses data frames; thus, the date and EPS are converted as a data frame in line 17. Data frames do not have field names; thus, the data frame names are created in line 18.



TTR's SMA function is used to calculate the moving average in line 23. The result is put in the data frame's *EPS_MA4* column. "df" in line 25 shows the content of the data frame. This line shows that *Date* is of type "date" and *EPS* and *EPS_MA4* are of type double.

The following *R* commands use the data frame to plot the EPS and moving average data.

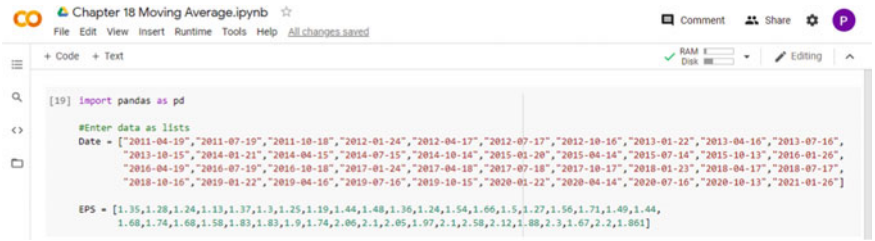
```r
pl <- ggplot(df , aes(x = Date))
pl <- pl + geom_line(aes(y = EPS_MA4, color = "EPS_MA4"),group = 1)
pl <- pl + geom_line(aes(y = EPS, color = "EPS"), group = 1)
pl <- pl +  theme_minimal()
pl <- pl + theme(legend.position = "top")
pl <- pl + labs(title ="Moving averages")
pl <- pl + labs(color="Legend")
pl
```



18.2.3 Moving Averages in Python

In this section, we will calculate and plot Johnson & Johnson's moving average in Python.

The EPS data are first entered as a list.

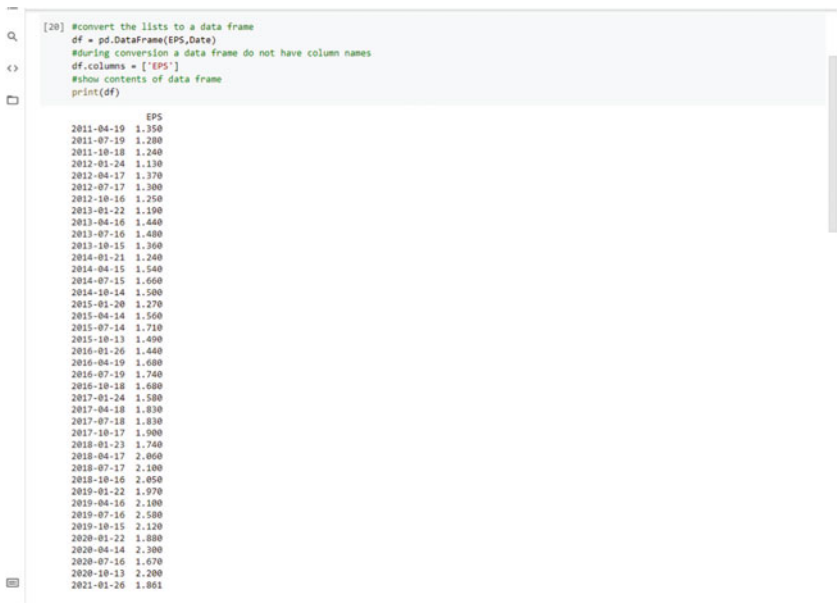



```
[19] import pandas as pd

     #Enter data as lists
     Date = ["2011-04-19","2011-07-19","2011-10-18","2012-01-24","2012-04-17","2012-07-17","2012-10-16","2013-01-22","2013-04-16","2013-07-16",
             "2013-10-15","2014-01-21","2014-04-15","2014-07-15","2014-10-14","2015-01-20","2015-04-14","2015-07-14","2015-10-13","2016-01-26",
             "2016-04-19","2016-07-19","2016-10-18","2017-01-24","2017-04-18","2017-07-18","2017-10-17","2018-01-23","2018-04-17","2018-07-17",
             "2018-10-16","2019-01-22","2019-04-16","2019-07-16","2019-10-15","2020-01-22","2020-04-14","2020-07-16","2020-10-13","2021-01-26"]

     EPS = [1.35,1.28,1.24,1.13,1.37,1.3,1.25,1.19,1.44,1.48,1.36,1.24,1.54,1.66,1.5,1.27,1.56,1.71,1.49,1.44,
            1.68,1.74,1.68,1.58,1.83,1.83,1.9,1.74,2.06,2.1,2.05,1.97,2.1,2.58,2.12,1.88,2.3,1.67,2.2,1.861]
```

The list is then converted to a data frame. Many Python features require the data to be in a data frame.



```
[20] #convert the lists to a data frame
     df = pd.DataFrame(EPS,Date)
     #during conversion a data frame do not have column names
     df.columns = ['EPS']
     #show contents of data frame
     print(df)

                 EPS
2011-04-19  1.350
2011-07-19  1.280
2011-10-18  1.240
2012-01-24  1.130
2012-04-17  1.370
2012-07-17  1.300
2012-10-16  1.250
2013-01-22  1.190
2013-04-16  1.440
2013-07-16  1.480
2013-10-15  1.360
2014-01-21  1.240
2014-04-15  1.540
2014-07-15  1.660
2014-10-14  1.500
2015-01-20  1.270
2015-04-14  1.560
2015-07-14  1.710
2015-10-13  1.490
2016-01-26  1.440
2016-04-19  1.680
2016-07-19  1.740
2016-10-18  1.680
2017-01-24  1.580
2017-04-18  1.830
2017-07-18  1.830
2017-10-17  1.900
2018-01-23  1.740
2018-04-17  2.060
2018-07-17  2.100
2018-10-16  2.050
2019-01-22  1.970
2019-04-16  2.100
2019-07-16  2.580
2019-10-15  2.120
2020-01-22  1.880
2020-04-14  2.300
2020-07-16  1.670
2020-10-13  2.200
2021-01-26  1.861
```

The *rolling* function calculates the moving average.

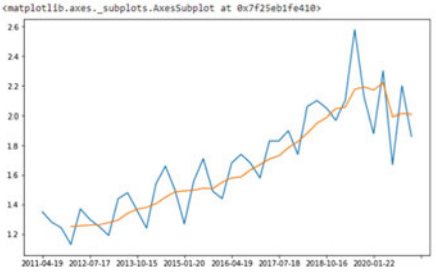
```
[21] #calculate moving average
     df['EPS'].rolling(window=4).mean()

2011-04-19       NaN
2011-07-19       NaN
2011-10-18       NaN
2012-01-24   1.25000
2012-04-17   1.25500
2012-07-17   1.26000
2012-10-16   1.26250
2013-01-22   1.27750
2013-04-16   1.29500
2013-07-16   1.34000
2013-10-15   1.36750
2014-01-21   1.38000
2014-04-15   1.40500
2014-07-15   1.45000
2014-10-14   1.48500
2015-01-20   1.49250
2015-04-14   1.49750
2015-07-14   1.51000
2015-10-13   1.50750
2016-01-26   1.55000
2016-04-19   1.58000
2016-07-19   1.58750
2016-10-18   1.63500
2017-01-24   1.67000
2017-04-18   1.70750
2017-07-18   1.73000
2017-10-17   1.78500
2018-01-23   1.82500
2018-04-17   1.88250
2018-07-17   1.95000
2018-10-16   1.98750
2019-01-22   2.04500
2019-04-16   2.05500
2019-07-16   2.17500
2019-10-15   2.19250
2020-01-22   2.17000
2020-04-14   2.22000
2020-07-16   1.99250
2020-10-13   2.01250
2021-01-26   2.00775
Name: EPS, dtype: float64
```

The Python code below plots the EPS and moving EPS together.

```
[23] #plot EPS and moving averages together
     df['EPS'].plot(figsize=(10, 6))
     df['EPS'].rolling(window=4).mean().plot()

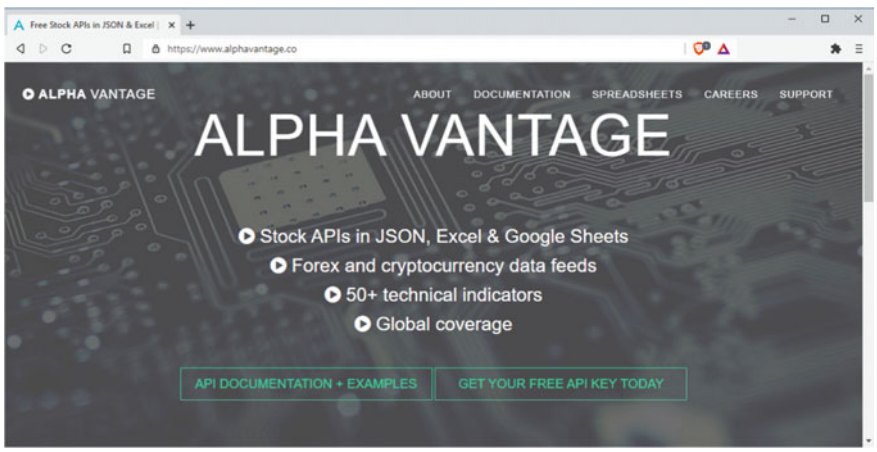
<matplotlib.axes._subplots.AxesSubplot at 0x7f25eb1fe410>
```



18.2.4 Data Source

18.2.4.1 https://www.alphavantage.co/

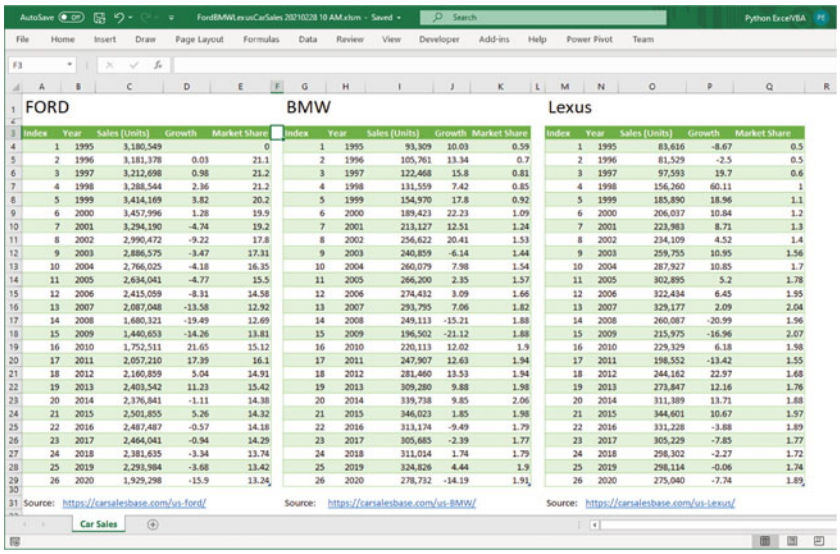
The data source for Johnson & Johnson's EPS data is https://www.alphavantage.co/.



It is free to use but you are limited to 5 API calls per minute and 500 calls per day. But this limitation is good enough for you to get an idea if you need to get the premium version or not.

18.3 Linear Trend

In this section, we analyze time-series data that are expected to change linearly over time. We will analyze car sales by units of Ford, BMW, and Lexus.



FORD

Index	Year	Sales (Units)	Growth	Market Share
1	1995	3,180,549		0
2	1996	3,181,378	0.03	21.1
3	1997	3,212,698	0.98	21.2
4	1998	3,288,544	2.36	21.2
5	1999	3,414,169	3.82	20.2
6	2000	3,457,996	1.28	19.9
7	2001	3,294,190	-4.74	19.2
8	2002	2,990,472	-9.22	17.8
9	2003	2,886,575	-3.47	17.31
10	2004	2,766,025	-4.18	16.35
11	2005	2,634,041	-4.77	15.5
12	2006	2,415,059	-8.31	14.58
13	2007	2,087,048	-13.58	12.92
14	2008	1,680,321	-19.49	12.69
15	2009	1,440,653	-14.26	13.81
16	2010	1,752,511	21.65	15.12
17	2011	2,057,210	17.39	16.1
18	2012	2,160,859	5.04	14.91
19	2013	2,403,542	11.23	15.42
20	2014	2,376,841	-1.11	14.38
21	2015	2,501,855	5.26	14.32
22	2016	2,487,487	-0.57	14.18
23	2017	2,464,041	-0.94	14.29
24	2018	2,381,635	-3.34	13.74
25	2019	2,293,984	-3.68	13.42
26	2020	1,929,298	-15.9	13.24

Source: https://carsalesbase.com/us-ford/

BMW

Index	Year	Sales (Units)	Growth	Market Share
1	1995	93,309	10.03	0.59
2	1996	105,761	13.34	0.7
3	1997	122,468	15.8	0.81
4	1998	131,559	7.42	0.85
5	1999	154,970	17.8	0.92
6	2000	189,423	22.23	1.09
7	2001	213,127	12.51	1.24
8	2002	256,622	20.41	1.53
9	2003	240,859	-6.14	1.44
10	2004	260,079	7.98	1.54
11	2005	266,200	2.35	1.57
12	2006	274,432	3.09	1.66
13	2007	293,795	7.06	1.82
14	2008	249,113	-15.21	1.88
15	2009	196,502	-21.12	1.88
16	2010	220,113	12.02	1.9
17	2011	247,907	12.63	1.94
18	2012	281,460	13.53	1.94
19	2013	309,280	9.88	1.98
20	2014	339,738	9.85	2.06
21	2015	346,023	1.85	1.98
22	2016	313,174	-9.49	1.79
23	2017	305,685	-2.39	1.77
24	2018	311,014	1.74	1.79
25	2019	324,826	4.44	1.9
26	2020	278,732	-14.19	1.91

Source: https://carsalesbase.com/us-BMW/

Lexus

Index	Year	Sales (Units)	Growth	Market Share
1	1995	83,616	-8.67	0.5
2	1996	81,529	-2.5	0.5
3	1997	97,593	19.7	0.6
4	1998	156,260	60.11	1
5	1999	185,890	18.96	1.1
6	2000	206,037	10.84	1.2
7	2001	223,983	8.71	1.3
8	2002	234,109	4.52	1.4
9	2003	259,755	10.95	1.56
10	2004	287,927	10.85	1.7
11	2005	302,895	5.2	1.78
12	2006	322,434	6.45	1.95
13	2007	329,177	2.09	2.04
14	2008	260,087	-20.99	1.96
15	2009	215,975	-16.96	2.07
16	2010	229,329	6.18	1.98
17	2011	198,552	-13.42	1.55
18	2012	244,162	22.97	1.68
19	2013	273,847	12.16	1.76
20	2014	311,389	13.71	1.88
21	2015	344,601	10.67	1.97
22	2016	331,228	-3.88	1.89
23	2017	305,229	-7.85	1.77
24	2018	298,302	-2.27	1.72
25	2019	298,114	-0.06	1.74
26	2020	275,040	-7.74	1.89

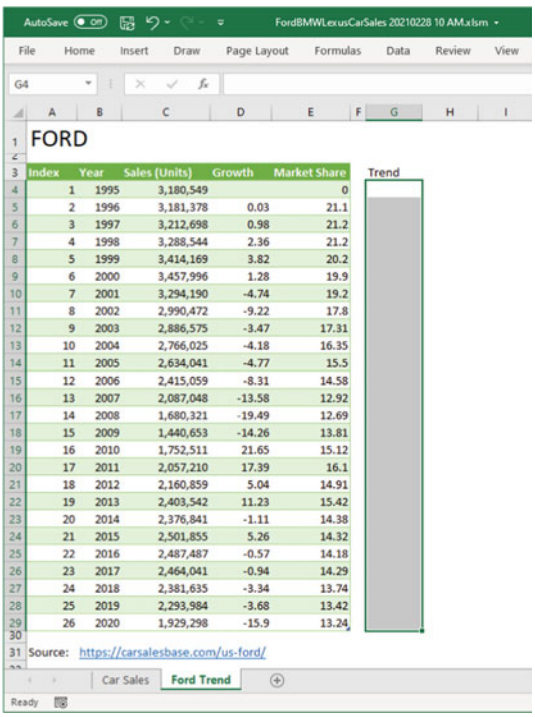
Source: https://carsalesbase.com/us-Lexus/

18.3.1 Linear Trend Analysis in Excel

18.3.1.1 Excel TREND Function

We will use Excel's *TREND* function to calculate the data of a trend line.

To use the *TREND* function, you need to select the range of the trend data.

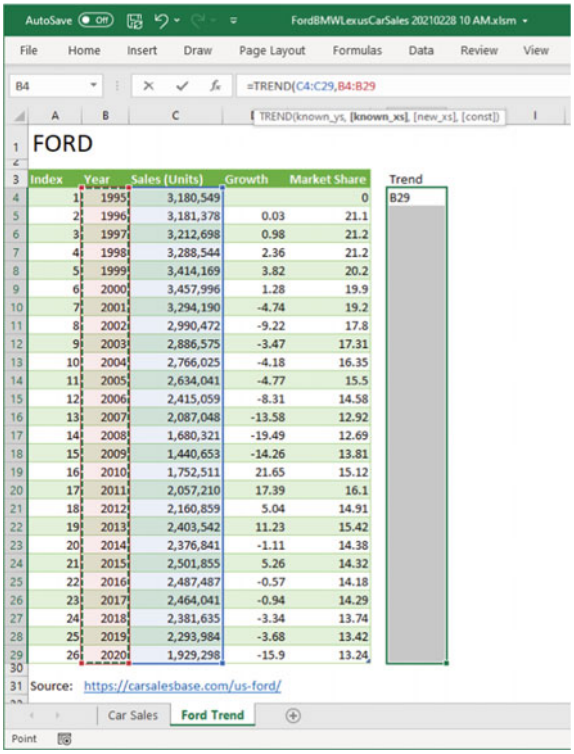


FORD

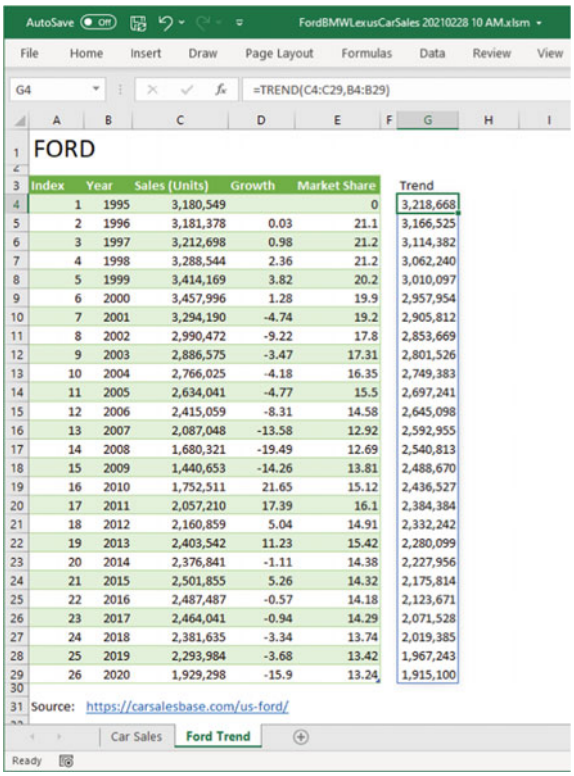
Index	Year	Sales (Units)	Growth	Market Share	Trend
1	1995	3,180,549		0	
2	1996	3,181,378	0.03	21.1	
3	1997	3,212,698	0.98	21.2	
4	1998	3,288,544	2.36	21.2	
5	1999	3,414,169	3.82	20.2	
6	2000	3,457,996	1.28	19.9	
7	2001	3,294,190	-4.74	19.2	
8	2002	2,990,472	-9.22	17.8	
9	2003	2,886,575	-3.47	17.31	
10	2004	2,766,025	-4.18	16.35	
11	2005	2,634,041	-4.77	15.5	
12	2006	2,415,059	-8.31	14.58	
13	2007	2,087,048	-13.58	12.92	
14	2008	1,680,321	-19.49	12.69	
15	2009	1,440,653	-14.26	13.81	
16	2010	1,752,511	21.65	15.12	
17	2011	2,057,210	17.39	16.1	
18	2012	2,160,859	5.04	14.91	
19	2013	2,403,542	11.23	15.42	
20	2014	2,376,841	-1.11	14.38	
21	2015	2,501,855	5.26	14.32	
22	2016	2,487,487	-0.57	14.18	
23	2017	2,464,041	-0.94	14.29	
24	2018	2,381,635	-3.34	13.74	
25	2019	2,293,984	-3.68	13.42	
26	2020	1,929,298	-15.9	13.24	

Source: https://carsalesbase.com/us-ford/

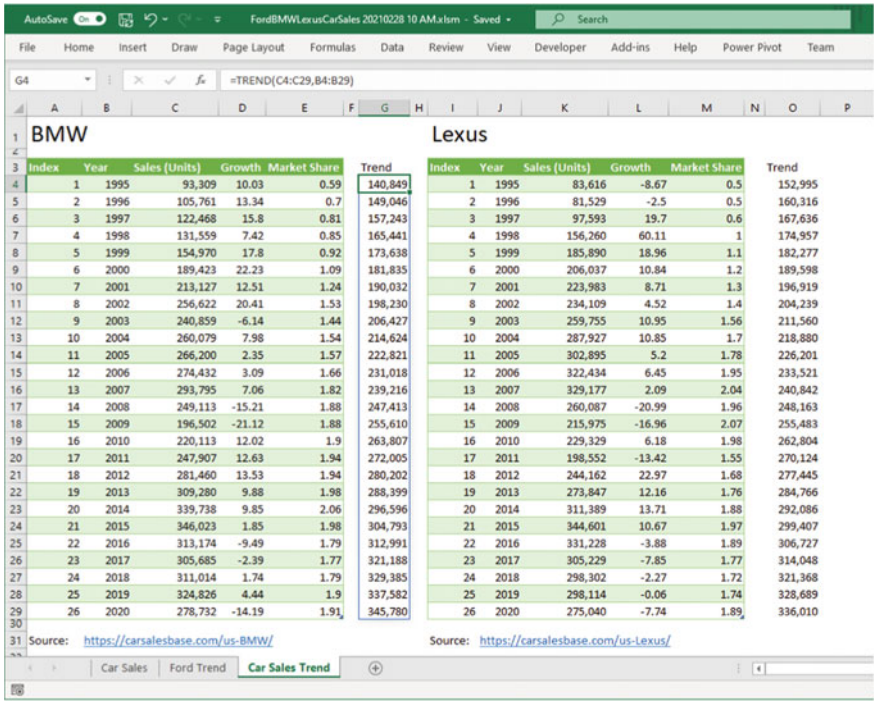
The next thing to do is first enter the range of the sales unit as the first parameter and the year as the second parameter.



The *TREND* function returns a dynamic array with the trend data.



The following is BMW's and Lexus's sales trend data.



AutoSave On FordBMWLexusCarSales 20210228 10 AM.xlsm - Saved Search

File Home Insert Draw Page Layout Formulas Data Review View Developer Add-ins Help Power Pivot Team

G4 =TREND(C4:C29,B4:B29)

BMW

Index	Year	Sales (Units)	Growth	Market Share	Trend
1	1995	93,309	10.03	0.59	140,849
2	1996	105,761	13.34	0.7	149,046
3	1997	122,468	15.8	0.81	157,243
4	1998	131,559	7.42	0.85	165,441
5	1999	154,970	17.8	0.92	173,638
6	2000	189,423	22.23	1.09	181,835
7	2001	213,127	12.51	1.24	190,032
8	2002	256,622	20.41	1.53	198,230
9	2003	240,859	-6.14	1.44	206,427
10	2004	260,079	7.98	1.54	214,624
11	2005	266,200	2.35	1.57	222,821
12	2006	274,432	3.09	1.66	231,018
13	2007	293,795	7.06	1.82	239,216
14	2008	249,113	-15.21	1.88	247,413
15	2009	196,502	-21.12	1.88	255,610
16	2010	220,113	12.02	1.9	263,807
17	2011	247,907	12.63	1.94	272,005
18	2012	281,460	13.53	1.94	280,202
19	2013	309,280	9.88	1.98	288,399
20	2014	339,738	9.85	2.06	296,596
21	2015	346,023	1.85	1.98	304,793
22	2016	313,174	-9.49	1.79	312,991
23	2017	305,685	-2.39	1.77	321,188
24	2018	311,014	1.74	1.79	329,385
25	2019	324,826	4.44	1.9	337,582
26	2020	278,732	-14.19	1.91	345,780

Source: https://carsalesbase.com/us-BMW/

Lexus

Index	Year	Sales (Units)	Growth	Market Share	Trend
1	1995	83,616	-8.67	0.5	152,995
2	1996	81,529	-2.5	0.5	160,316
3	1997	97,593	19.7	0.6	167,636
4	1998	156,260	60.11	1	174,957
5	1999	185,890	18.96	1.1	182,277
6	2000	206,037	10.84	1.2	189,598
7	2001	223,983	8.71	1.3	196,919
8	2002	234,109	4.52	1.4	204,239
9	2003	259,755	10.95	1.56	211,560
10	2004	287,927	10.85	1.7	218,880
11	2005	302,895	5.2	1.78	226,201
12	2006	322,434	6.45	1.95	233,521
13	2007	329,177	2.09	2.04	240,842
14	2008	260,087	-20.99	1.96	248,163
15	2009	215,975	-16.96	2.07	255,483
16	2010	229,329	6.18	1.98	262,804
17	2011	198,552	-13.42	1.55	270,124
18	2012	244,162	22.97	1.68	277,445
19	2013	273,847	12.16	1.76	284,766
20	2014	311,389	13.71	1.88	292,086
21	2015	344,601	10.67	1.97	299,407
22	2016	331,228	-3.88	1.89	306,727
23	2017	305,229	-7.85	1.77	314,048
24	2018	298,302	-2.27	1.72	321,368
25	2019	298,114	-0.06	1.74	328,689
26	2020	275,040	-7.74	1.89	336,010

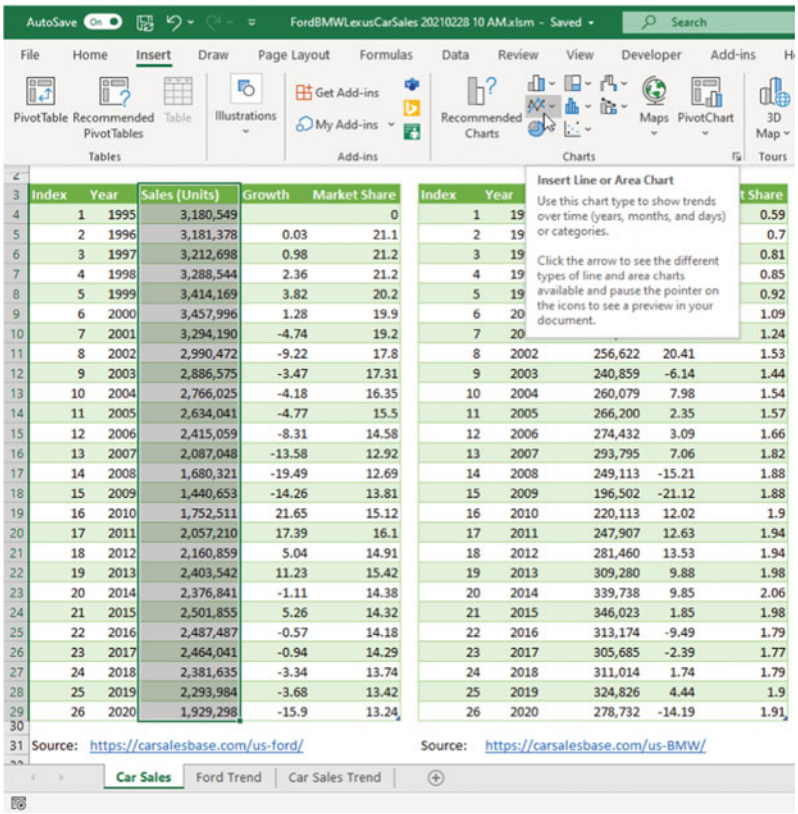
Source: https://carsalesbase.com/us-Lexus/

Car Sales | Ford Trend | Car Sales Trend

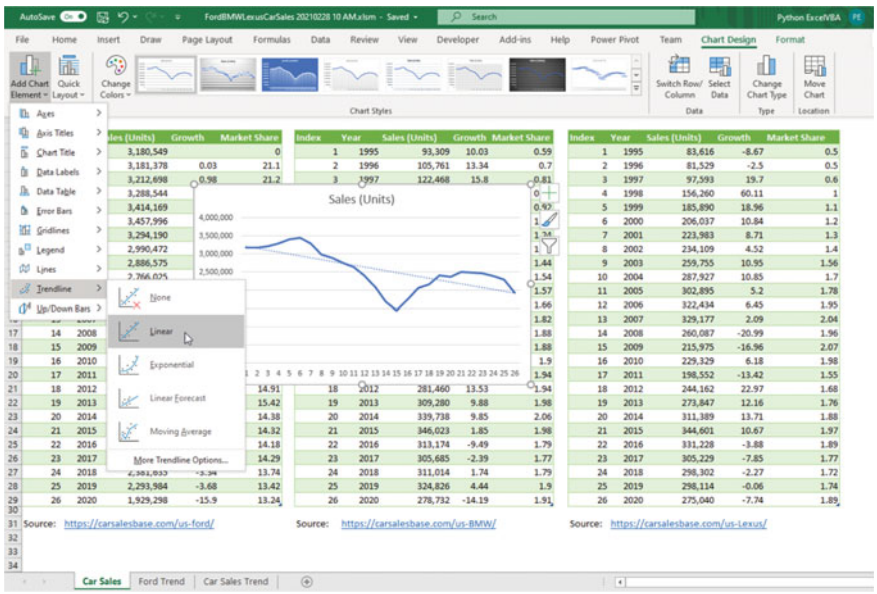
18.3.1.2 Excel Linear Trend Graph

Excel has trend graphs. One type of trends graphs available is the linear trend. The following are steps to create a *Linear Trend Graph* for Ford's sales data.

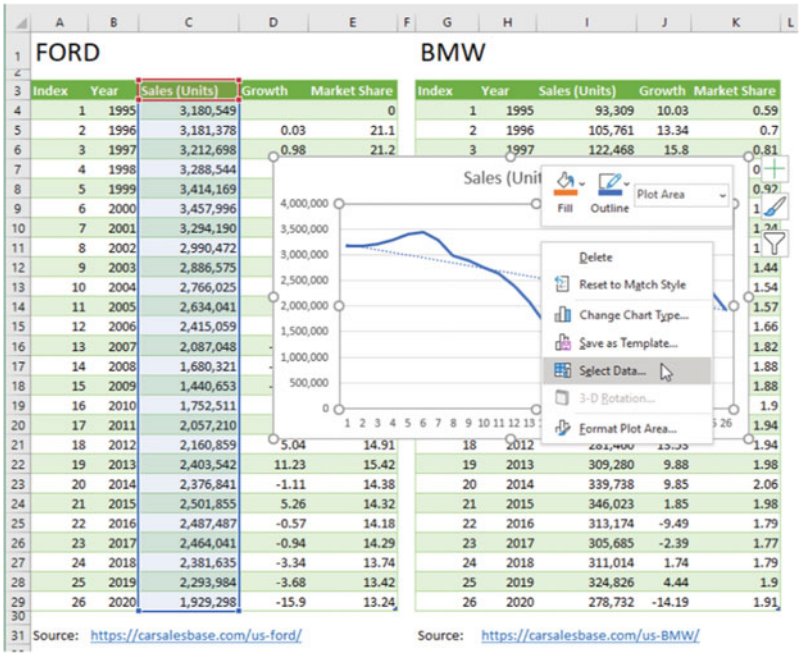
First, select the sales data and then click on *Insert Line or Area Chart* in the *Insert* tab.



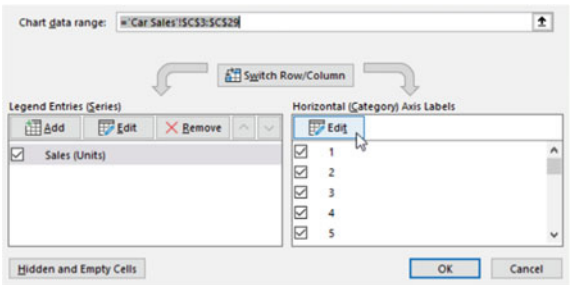
Next, with the chart selected, click on the *Linear Trend* in the *Chart Design* tab.



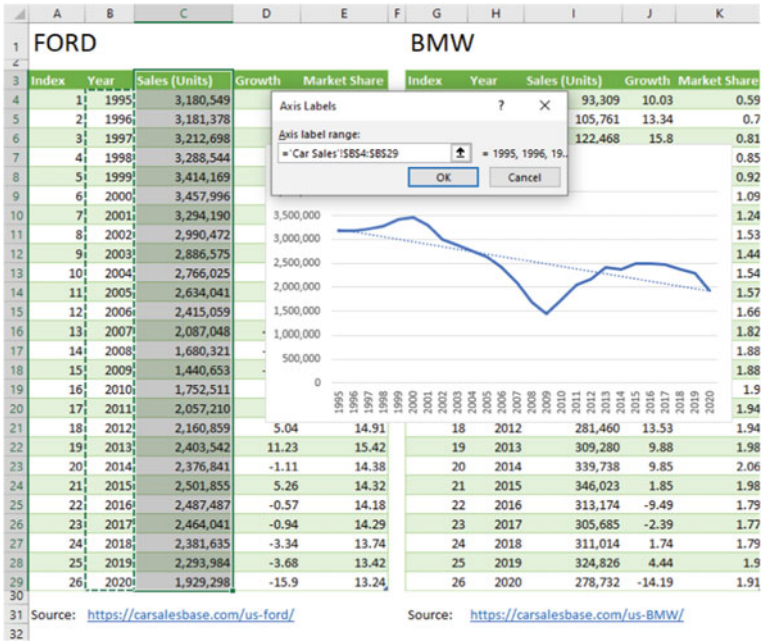
Our next task is to change the x-axis to year. To do this, select the chart and right-mouse click and select *Select Data*.



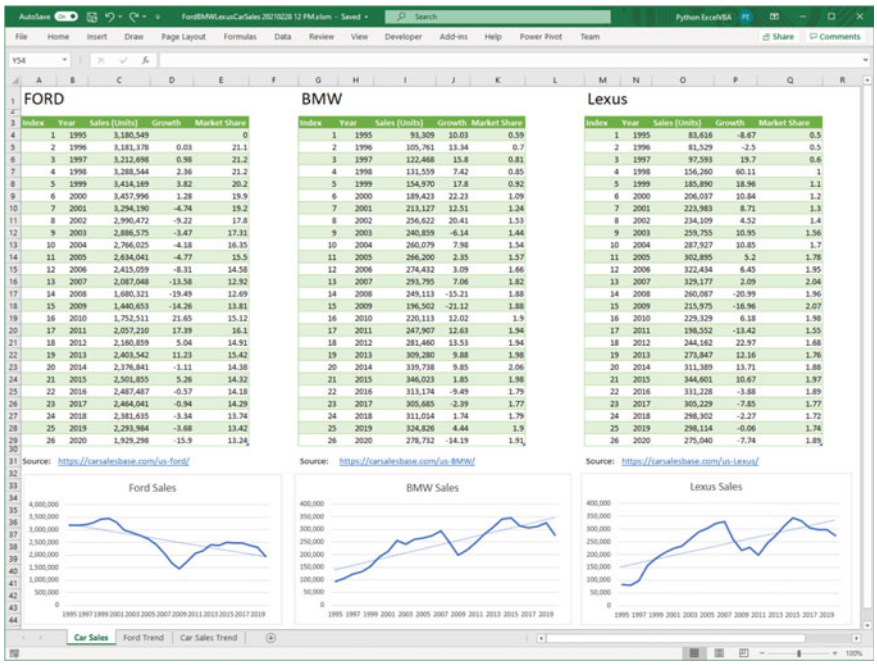
Next, click on the *Edit* button.



Next select the year data.



Next, do the same steps for BMW's and Lexus' sales data.
The following are the *Linear Trend* graphs for Ford's, BMW's, and Lexus' sales data.



From the three trend graphs, we can see that Ford's sales have a downtrend from the years 1995 to 2019 while BMW and Lexus had an upward sales trend from 1995 to 2019. The downward trend for both BMW and Lexus from 2008 and 2009 was because of the 2008 great recession.

18.3.1.3 Data Source

https://carsalesbase.com/

The data source for sales car data was https://carsalesbase.com/. Power query was used to retrieve the sales data.

The following is the M code to retrieve Ford's sales data.

The following is the M code to retrieve BMW's sales data.

The following is the M code to retrieve Lexus' sales data.

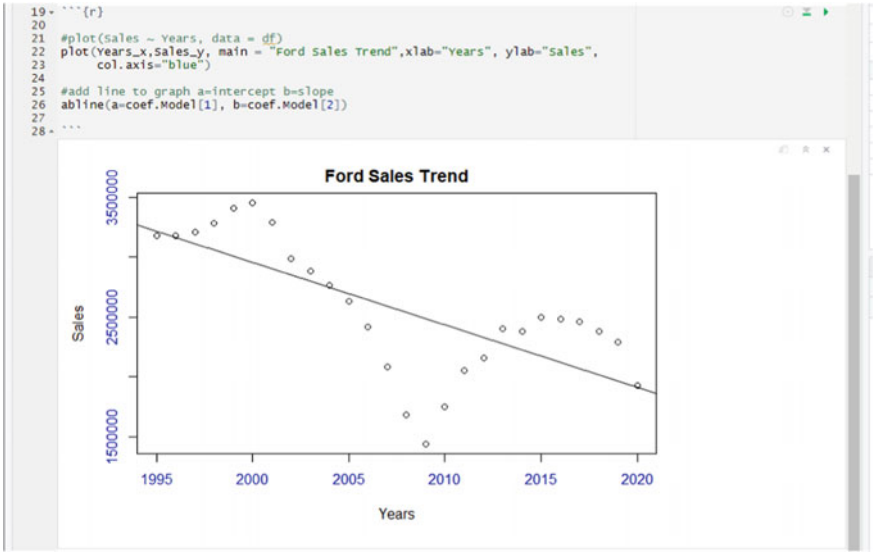
18.3.2 Linear Trend Analysis in R

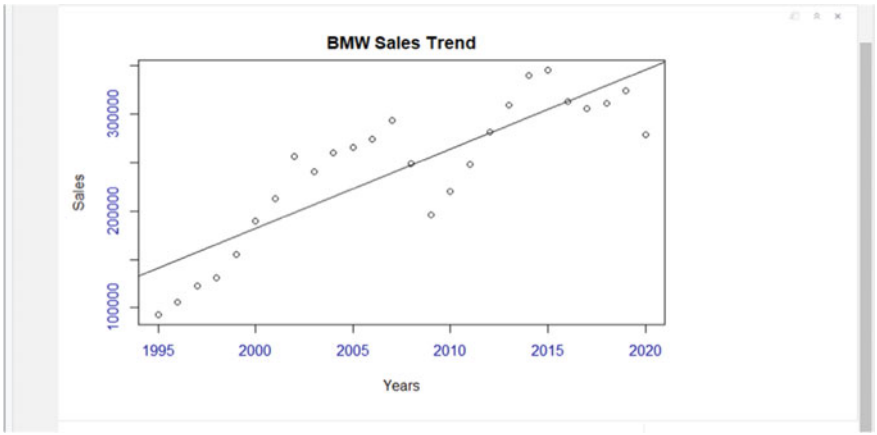
The following is the *R* code to do *linear trend analysis* on Ford's, BMW's, and Lexus's sales trend data.

18.3.2.1 Ford's Sales Linear Trend Analysis in R

The following *R* code enters Ford's sales data as vectors. It also creates a data frame variable from the years and sales variables.

```
#Date:Mar-01-2021
```{r}
#Enter data as vectors
Years_x <-c(1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020)

Sales_y <-c(3180549,3181378,3212698,3288544,3414169,3457996,3294190,2990472,2886575,2766025,2634041,2415059,2087048,1680321,1440653,1752511,2057210,2160859,2403542,2376841,2501855,2487487,2464041,2381635,2293984,1929298)

#create data frame field names
df<- data.frame(Years_x,Sales_y)
names(df) <- c('Years', 'Sales')
```
```

The following *R* code does a regression on Ford's sales data. The *model* function does a regression on the sales data. The *model* function requires a data frame data set.

```
```{r}
#do regression to get slop and intercept of line
Model <- lm(Sales ~ Years, data=df)
coef.Model <- coef(Model)
coef.Model

```

(Intercept)   Years
107243381.24   -52142.71
```

From the regression output, the trend line for Ford's sales number is

$$y = 107{,}243{,}381 - 52{,}142\,\text{Years}$$

The following *R* code plots Ford's sales data and the trend line. The trend line is from the above regression line. The *R* function *abline* uses regression intercept and regression slope to plot the trend line.

```
```{r}

#plot(Sales ~ Years, data = df)
plot(Years_x,Sales_y, main = "Ford Sales Trend",xlab="Years", ylab="Sales",
     col.axis="blue")

#add line to graph a=intercept b=slope
abline(a=coef.Model[1], b=coef.Model[2])

```
```



18.3.2.2 BMW's Sales Linear Trend Analysis in R

The following *R* code enters BMW's sales data as vectors. It also creates a data frame variable from the years and sales variables.

```
#Date:Mar-01-2021
```{r}
#Enter data as vectors
Years_x <-c(1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020)

Sales_y <-c(93309,105761,122468,131559,154970,189423,213127,256622,240859,260079,266200,274432,293795,249113,196502,220113,247907,281460,309280,339738,346023,313174,305685,311014,324826,278732)

#create data frame field names
df<- data.frame(Years_x,Sales_y)
names(df) <- c('Years', 'Sales')
```
```

The following *R* code does a regression on BMW's sales data. The *model* function does a regression on the sales data. The *model* function requires a data frame data set.

```
```{r}
#do regression to get slop and intercept of trend line
Model <- lm(Sales_y ~ Years_x, data=df)
coef.Model <- coef(Model)
coef.Model

```

  (Intercept)   Years_x
-16212610.715   8197.223
```

From the regression output, the trend line for BMW's sales number is

$$y = -16{,}212{,}610 + 8{,}197 \text{ Years}$$

The following *R* code plots BMW's sales data and the trend line. The trend line is from the above regression line. The *R* function *abline* uses regression intercept and regression slope to plot the trend line.



18.3.2.3 Lexus' Sales Linear Trend Analysis in R

The following *R* code enters Lexus's sales data as vectors. It also creates a data frame variable from the years and sales variables.

```
#Date:Mar-01-2021
```{r}
#Enter data as vectors
Years_x <-c(1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014
,2015,2016,2017,2018,2019,2020)

Sales_y <-c(83616,81529,97593,156260,185890,206037,223983,234109,259755,287927,302895,322434,329177,260087,2159
75,229329,198552,244162,273847,311389,344601,331228,305229,298302,298114,275040)

#create data frame field names
df<- data.frame(Years_x,Sales_y)
names(df) <- c('Years', 'Sales')
```
```

The following *R* code does a regression on Lexus' sales data. The *model* function does a regression on the sales data. The *model* function requires a data frame data set.

```
```{r}
#do regression to get slop and intercept of trend line
Model <- lm(Sales_y ~ Years_x, data=df)
coef.Model <- coef(Model)
coef.Model

```

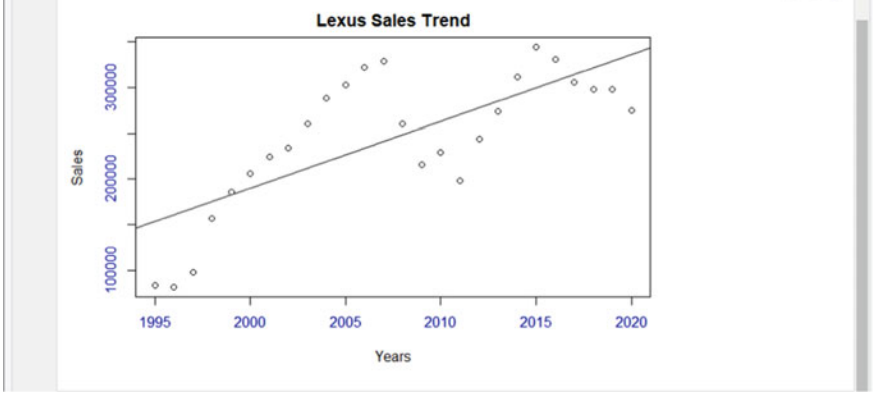
  (Intercept)   Years_x
-14451563.072   7320.581
```

From the regression output, the trend line for Lexus's sales number is

$$y = 16{,}212{,}610 + 7{,}320\,\text{Years}$$

The following *R* code plots Lexus sales data and the trend line. The trend line is from the above regression line. The *R* function *abline* uses regression intercept and regression slope to plot the trend line.

```r
#plot(Sales ~ Years, data = df)
plot(Years_x,Sales_y, main = "Lexus Sales Trend",xlab="Years", ylab="Sales",
     col.axis="blue")

#add line to graph a=intercept b=slope
abline(a=coef.Model[1], b=coef.Model[2])
```

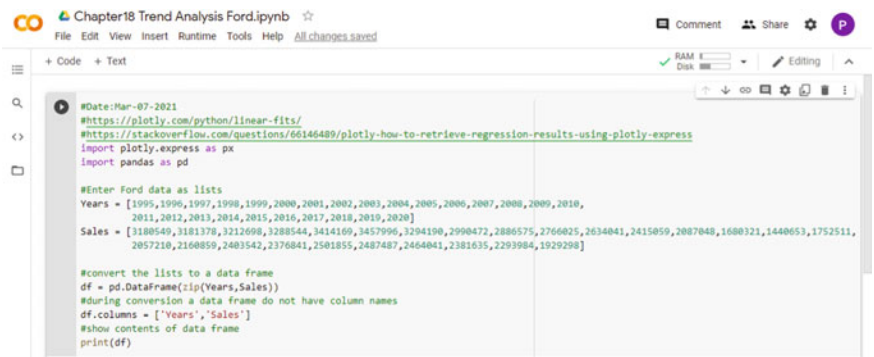


18.3.3 Linear Trend Analysis in Python

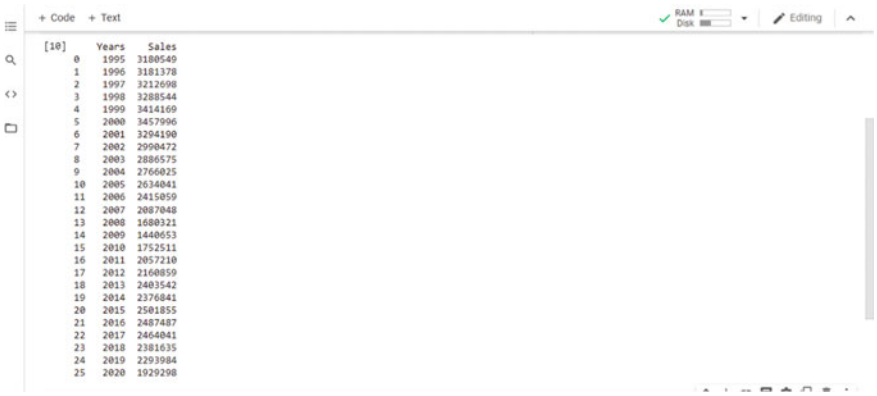
The following is the *Python* code to do *linear trend analysis* on Ford's, BMW's, and Lexus's sales trend data.

18.3.3.1 Ford's Sales Linear Trend Analysis in Python

The following *Python* code enters Lexus' sales data as lists. It also creates a data frame variable from the year and sale variables.




```
#Date:Mar-07-2021
#https://plotly.com/python/linear-fits/
#https://stackoverflow.com/questions/66146489/plotly-how-to-retrieve-regression-results-using-plotly-express
import plotly.express as px
import pandas as pd

#Enter Ford data as lists
Years = [1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,
         2011,2012,2013,2014,2015,2016,2017,2018,2019,2020]
Sales = [3180549,3181378,3212698,3288544,3414169,3457996,3294190,2990472,2886575,2766025,2634041,2415059,2087048,1680321,1440653,1752511,
         2057210,2160859,2403542,2376841,2501855,2487487,2464041,2381635,2293984,1929298]

#convert the lists to a data frame
df = pd.DataFrame(zip(Years,Sales))
#during conversion a data frame do not have column names
df.columns = ['Years','Sales']
#show contents of data frame
print(df)
```



```
[10]    Years   Sales
     0   1995  3180549
     1   1996  3181378
     2   1997  3212698
     3   1998  3288544
     4   1999  3414169
     5   2000  3457996
     6   2001  3294190
     7   2002  2990472
     8   2003  2886575
     9   2004  2766025
    10   2005  2634041
    11   2006  2415059
    12   2007  2087048
    13   2008  1680321
    14   2009  1440653
    15   2010  1752511
    16   2011  2057210
    17   2012  2160859
    18   2013  2403542
    19   2014  2376841
    20   2015  2501855
    21   2016  2487487
    22   2017  2464041
    23   2018  2381635
    24   2019  2293984
    25   2020  1929298
```

The following *Python* code does a regression on Ford's sales data. The *get_trendline* function does a regression on the sales data.

```
# retrieve model estimates
fig = px.scatter(df, x="Years", y="Sales", trendline="ols",title="Ford Sales")
model = px.get_trendline_results(fig)
results = model.iloc[0]["px_fit_results"]
alpha = results.params[0]
beta = results.params[1]
p_beta = results.pvalues[1]
r_squared = results.rsquared

line1 = 'y = ' + str(round(alpha, 4)) + ' + ' + str(round(beta, 4))+'x'
line2 = 'p-value = ' + '{:.5f}'.format(p_beta)
line3 = 'R^2 = ' + str(round(r_squared, 3))

print (line1)
print (line2)
print (line3)
```

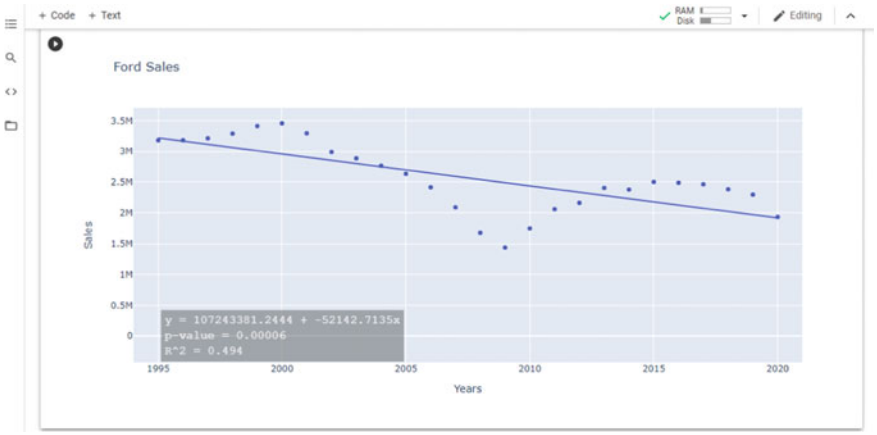
```
y = 107243381.2444 + -52142.7135x
p-value = 0.00006
R^2 = 0.494
```

From the regression output, the trend line for Ford's sales number is

$$\text{y} = 107{,}243{,}381 - 52{,}142\,\text{Years}$$

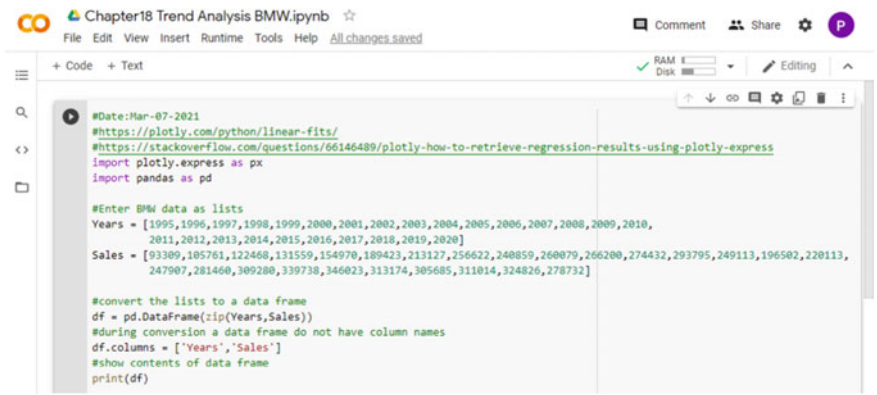
The following *Python* code plots Ford's sales data and the trend line. The trend line is from the above regression line.

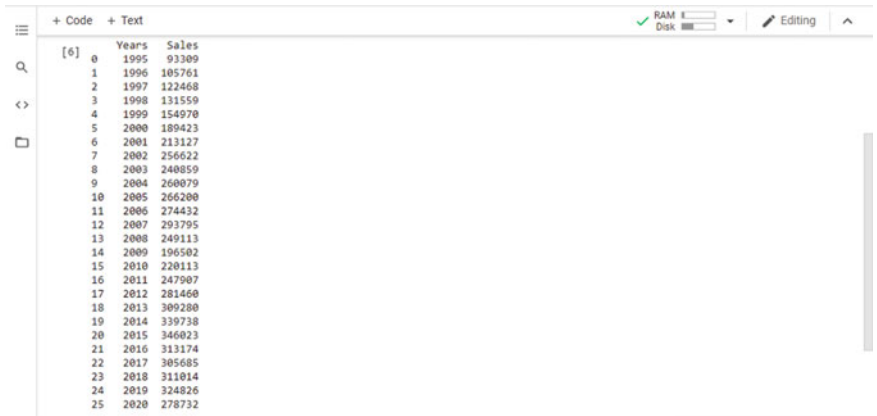
```
summary = line1 + '<br>' + line2 + '<br>' + line3
fig.add_annotation(
        x=2000,
        y=140,
        xref="x",
        yref="y",
        text=summary,
        showarrow=False,
        font=dict(
            family="Courier New, monospace",
            size=16,
            color="#ffffff"
            ),
        align="left",
        arrowhead=2,
        arrowsize=1,
        arrowwidth=2,
        arrowcolor="#636363",
        ax=20,
        ay=-30,
        borderwidth=2,
        borderpad=4,
        bgcolor="rgba(100,100,100, 0.6)",
        opacity=0.8
        )

fig.update_layout(xaxis_range=[1994,2021])
fig.show()
```



18.3.3.2 BMW's Sales Linear Trend Analysis in Python

The following *Python* code enters BMW's sales data as lists. It also creates a data frame variable from the year and sale variables.





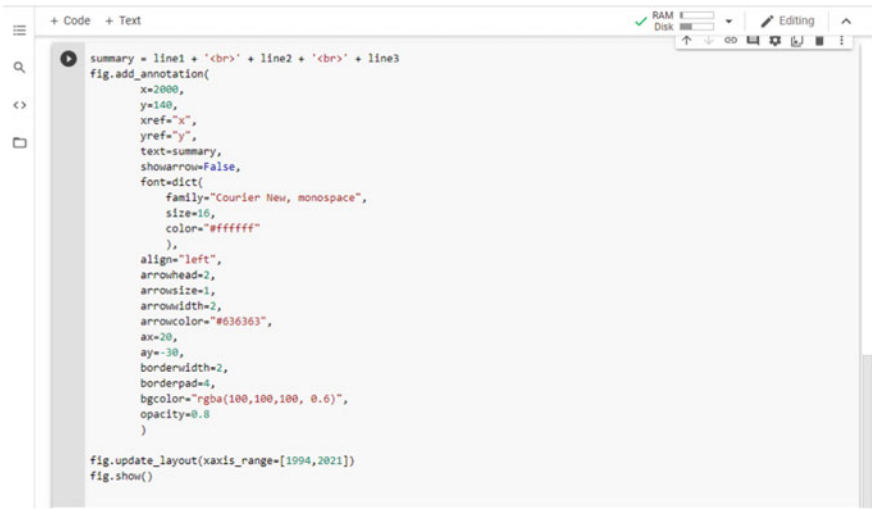
The following *Python* code does a regression on BMW's sales data. The *get_trendline* function does a regression on the sales data.

```
[9] # retrieve model estimates
    fig = px.scatter(df, x="Years", y="Sales", trendline="ols",title="BMW Sales")
    model = px.get_trendline_results(fig)
    results = model.iloc[0]["px_fit_results"]
    alpha = results.params[0]
    beta = results.params[1]
    p_beta = results.pvalues[1]
    r_squared = results.rsquared

    line1 = 'y = ' + str(round(alpha, 4)) + ' + ' + str(round(beta, 4))+'x'
    line2 = 'p-value = ' + '{:.5f}'.format(p_beta)
    line3 = 'R^2 = ' + str(round(r_squared, 3))

    print (line1)
    print (line2)
    print (line3)

    y = -16212610.7145 + 8197.2229x
    p-value = 0.00000
    R^2 = 0.732
```

From the regression output, the trend line for BMW's sales number is

$$\text{y} \ = \ -16{,}212{,}610 + 8{,}197 \text{ Years}$$

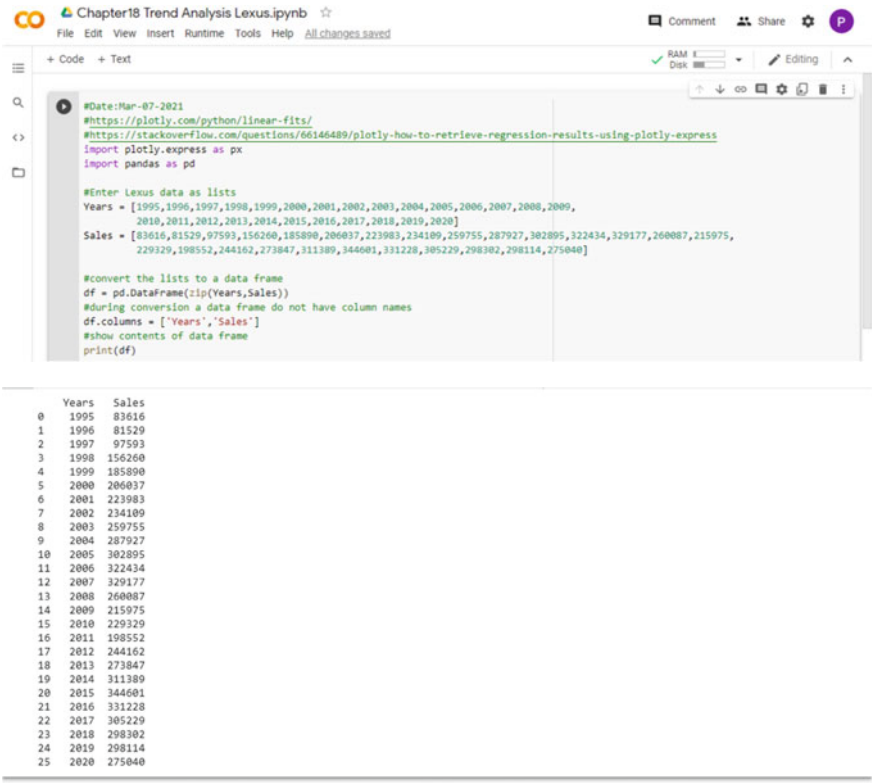
The following *Python* code plots BMW's sales data and the trend line. The trend line is from the above regression line.



```
summary = line1 + '<br>' + line2 + '<br>' + line3
fig.add_annotation(
        x=2000,
        y=140,
        xref="x",
        yref="y",
        text=summary,
        showarrow=False,
        font=dict(
            family="Courier New, monospace",
            size=16,
            color="#ffffff"
            ),
        align="left",
        arrowhead=2,
        arrowsize=1,
        arrowwidth=2,
        arrowcolor="#636363",
        ax=20,
        ay=-30,
        borderwidth=2,
        borderpad=4,
        bgcolor="rgba(100,100,100, 0.6)",
        opacity=0.8
        )

fig.update_layout(xaxis_range=[1994,2021])
fig.show()
```

18.3.3.3 Lexus' Sales Linear Trend Analysis in Python

The following *Python* code enters Lexus' sales data as lists. It also creates a data frame variable from the year and sale variables.



```
#Date:Mar-07-2021
#https://plotly.com/python/linear-fits/
#https://stackoverflow.com/questions/66146489/plotly-how-to-retrieve-regression-results-using-plotly-express
import plotly.express as px
import pandas as pd

#Enter Lexus data as lists
Years = [1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,
         2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020]
Sales = [83616,81529,97593,156260,185890,206037,223983,234109,259755,287927,302895,322434,329177,260087,215975,
         229329,198552,244162,273847,311389,344601,331228,305229,298302,298114,275040]

#convert the lists to a data frame
df = pd.DataFrame(zip(Years,Sales))
#during conversion a data frame do not have column names
df.columns = ['Years','Sales']
#show contents of data frame
print(df)
```

```
    Years   Sales
0    1995   83616
1    1996   81529
2    1997   97593
3    1998  156260
4    1999  185890
5    2000  206037
6    2001  223983
7    2002  234109
8    2003  259755
9    2004  287927
10   2005  302895
11   2006  322434
12   2007  329177
13   2008  260087
14   2009  215975
15   2010  229329
16   2011  198552
17   2012  244162
18   2013  273847
19   2014  311389
20   2015  344601
21   2016  331228
22   2017  305229
23   2018  298302
24   2019  298114
25   2020  275040
```

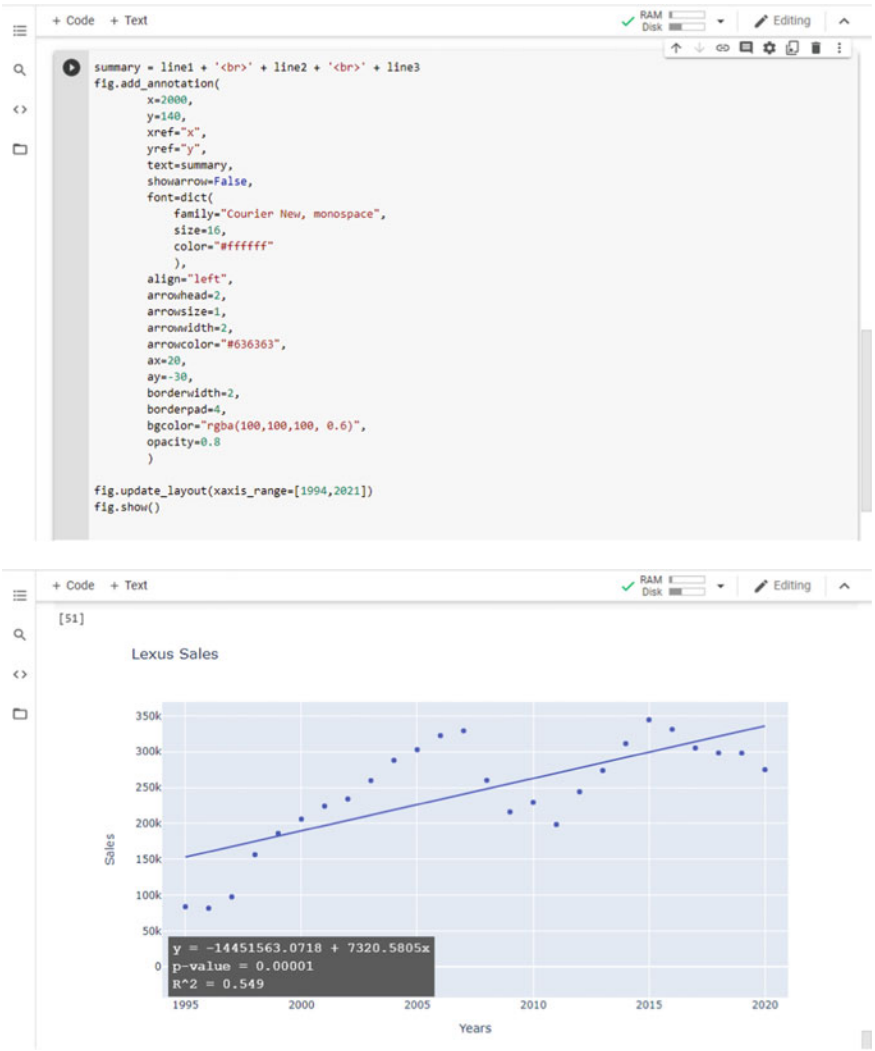
The following *Python* code does a regression on Lexus' sales data. The *get_trendline* function does a regression on the sales data.

```
[57] # retrieve model estimates
     fig = px.scatter(df, x="Years", y="Sales", trendline="ols",title="Lexus Sales")
     model = px.get_trendline_results(fig)
     results = model.iloc[0]["px_fit_results"]
     alpha = results.params[0]
     beta = results.params[1]
     p_beta = results.pvalues[1]
     r_squared = results.rsquared

     line1 = 'y = ' + str(round(alpha, 4)) + ' + ' + str(round(beta, 4))+'x'
     line2 = 'p-value = ' + '{:.5f}'.format(p_beta)
     line3 = 'R^2 = ' + str(round(r_squared, 3))

     print (line1)
     print (line2)
     print (line3)

     y = -14451563.0718 + 7320.5805x
     p-value = 0.00001
     R^2 = 0.549
```

From the regression output, the trend line for Lexus's sales number is

$$y = 16{,}212{,}610 + 7{,}320 \text{ Years}$$

The following *Python* code plots Lexus' sales data and the trend line. The trend line is from the above regression line.



18.4 Exponential Smoothing

Smoothing techniques are often used to forecast future values of a time series. One problem that arises in using a moving average to forecast time series is that values at the ends of the series are lost. Therefore, we must subjectively extend the graph of the moving average into the future. No exact calculation of a forecast is available because generating the moving average at a future time period t requires that we know one or more future time series values. A technique that leads to forecasts that can be explicitly calculated is called *exponential smoothing*. To use the exponential smoothing technique in forecasting, we need only past and current values of the time series.

To obtain an exponentially smoothed series, we first need to choose a weight α between 0 and 1 called the *exponential smoothing constant*. The exponentially smoothed series, denoted s_t is then calculated as follows:

$$
\begin{aligned}
s_1 &= x_1 \\
s_2 &= \alpha x_2 + (1-\alpha)s_1 \\
s_3 &= \alpha x_3 + (1-\alpha)s_2 \\
&\dots \\
&\dots \\
&\dots \\
s_i &= \alpha x_t + (1-\alpha)s_{t-1}
\end{aligned}
$$

We can see that the exponentially smoothed value at time t is simply a weighted average of the current time-series value x_t and the exponentially smoothed value at the previous time period, s_{t-1}. Then we can use s_t to do forecasting as follows:

$$\widehat{x}_{t+1} = s_t = \alpha x_t + (1-\alpha)s_{t-1}$$

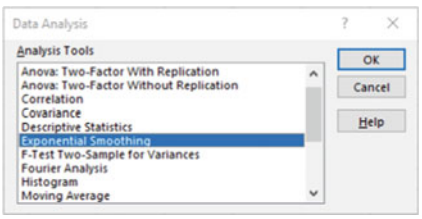
where $\widehat{x}_{t+1}$ is the next period's forecast value. In other words, $\widehat{x}_{t+1}$ is expressed in terms of the smoothing constant times x_t plus $(1 - \alpha)$ times s_{t-1}.

18.4.1 Exponential Smoothing in Excel

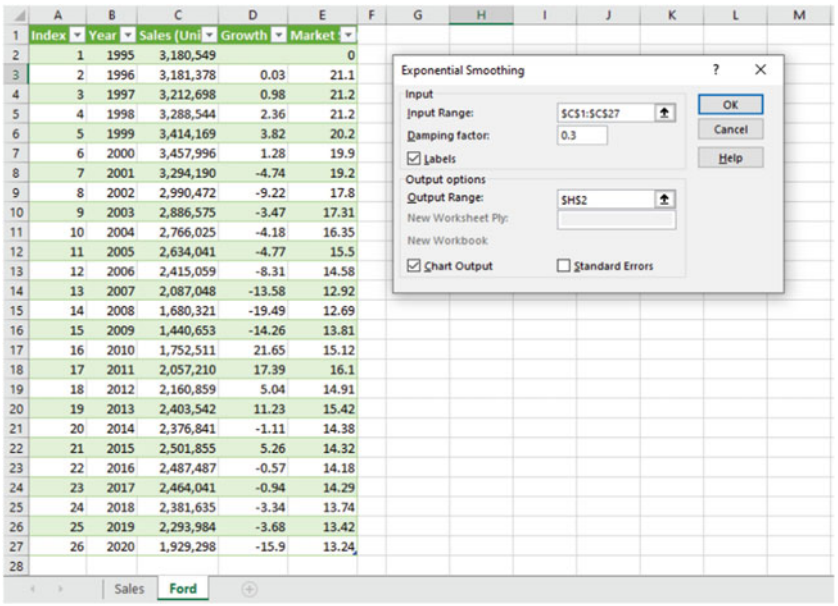
18.4.1.1 Exponential Smoothing Ford's Sales

The following are the steps in how to do *exponential smoothing* on Ford's sales.

(1) Choose *Exponential Smoothing* in the *Data Analysis* window.

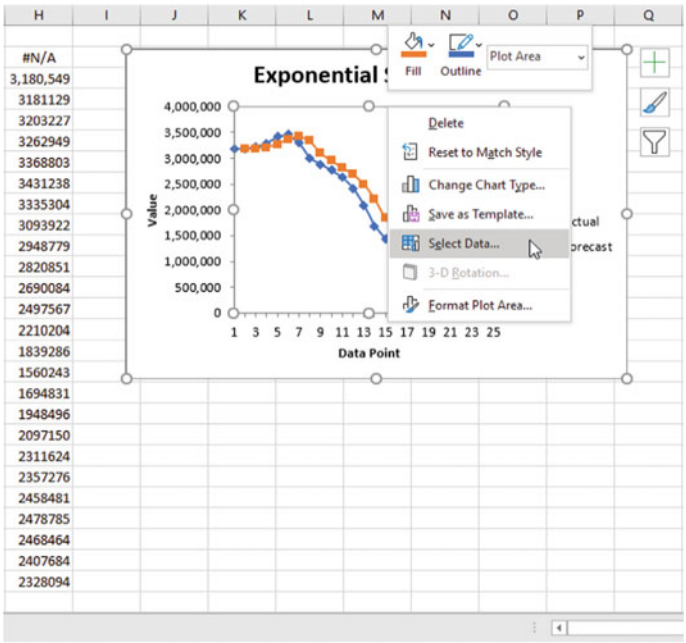


(2) Fill out the *Exponential Smoothing* screen as shown below.

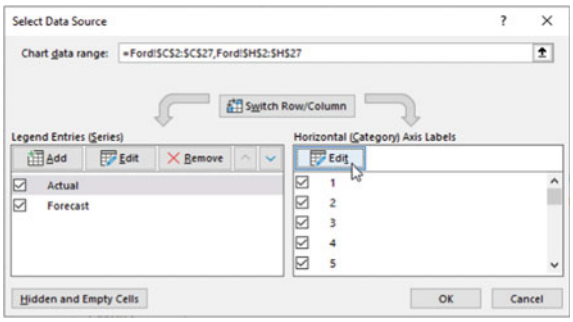


The *Damping factor* in the *Exponential Smoothing* is 1-alpha.

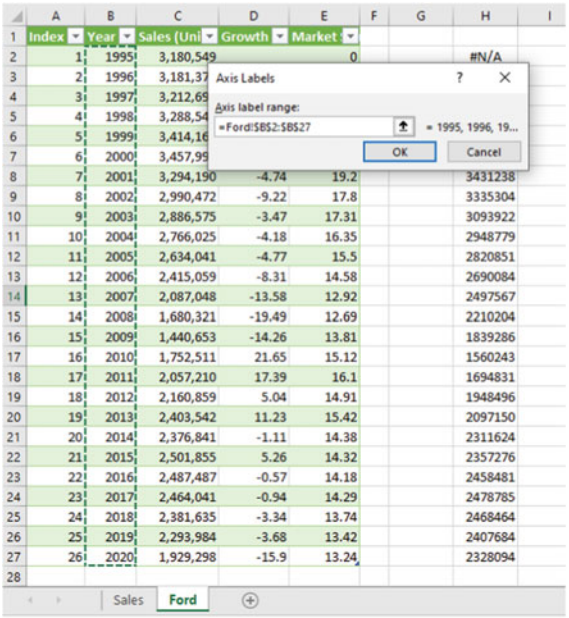
(3) Right-mouse click on the chart and choose *Select Data*



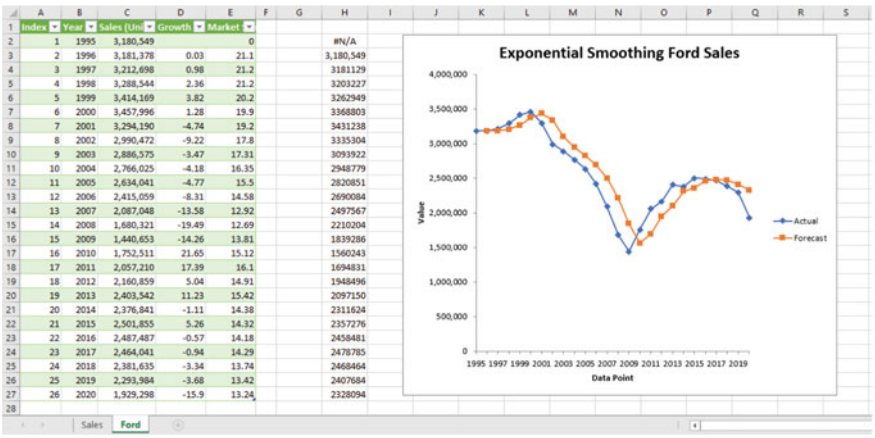
(4) Click on *Edit* under the *Horizontal* section



(5) Choose b2 to b27 as the a-axis label.



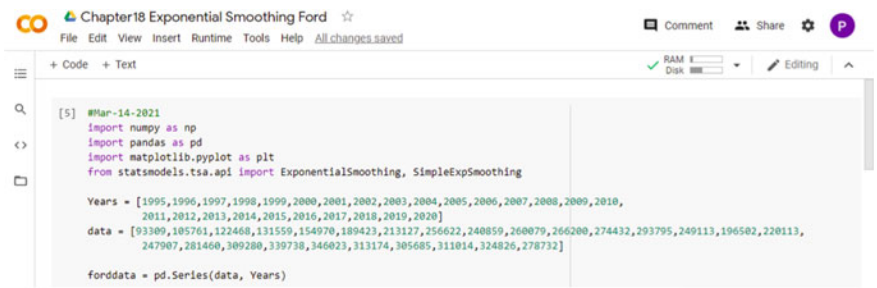
Below shows Ford's exponential smoothing chart and calculation. Column H is Ford's exponential smoothing calculation.

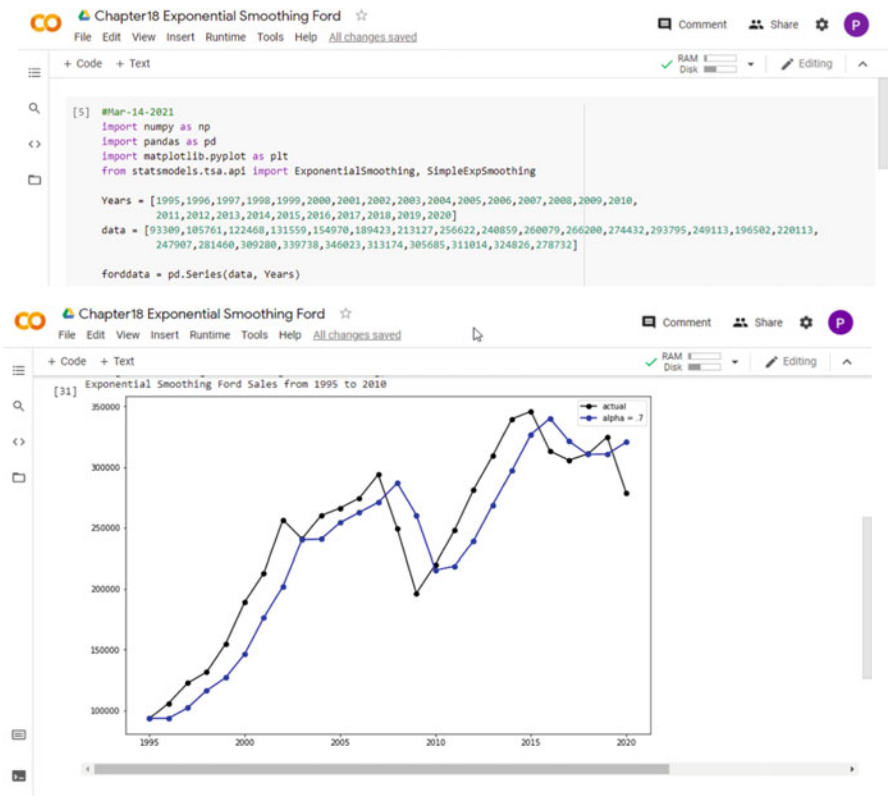


18.4.2 Exponential Smoothing in Python

18.4.2.1 Exponential Smoothing Ford's Sales

The following show the Python code to do *exponential smoothing* on Ford's 1995–2020 sales data.





18.5 Summary

In this chapter, we examined time-series component analysis. The major components of a time series are the trend, cyclical, seasonal, and irregular components. To analyze these time-series components, we used the moving-average method to obtain seasonally adjusted time series. We discussed two forecasting models. The forecasting models are linear trend and exponential smoothing.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O’Reilly Media, 2017.

Check for updates

Index Numbers and Stock Market Indexes

19

19.1 Introduction

In this chapter, we will look at indexes. Index numbers are numbers that compare an activity in one time or place to a similar activity in a specific base period or place. The first index that we will look at is the simple price index of a single item. When used this way, we can measure the relative price change over time.

This chapter is broken down into the following sections. Section 19.2 discusses the Simple Price Index, Sect. 19.3 discusses the Laspeyres Price Index, Sect. 19.4 discusses the Paasche Price Index, and Sect. 19.5 discusses Fisher's Ideal Price Index. In Sect. 19.6, we look at the S&P 500 Index and NASDAQ Composite Index. In Sect. 19.7, we take a look at the Dow Jones Industrial Average, and in Sect. 19.8, we discuss components of the Dow Jones Industrial Average. In Sect. 19.9, we discuss components of the S&P 500, and in Sect. 19.10 we discuss components of the NASDAQ Composite Index. In Sect. 19.11, we use Python to calculate the four statistical moments of the rate of returns of every component in the S&P 500. Finally, in Sect. 19.12, we summarize the chapter.

19.2 Simple Price Index

The formula for the simple price index is

$$I_t = \frac{P_t}{P_0} * 100$$

where P_t is the price in period t, P_0 is the base period price, and I_t is the simple price index at time t.

19.2.1 Example 19.1

Suppose we would like to create a simple price index for eggs. We know the price of eggs for the following years.

Price of Eggs	
Year	Price
2015	$1.00
2016	$1.20
2017	$1.50

In creating an index, we have to decide on a base year. Let us choose 2015 as the base year in this case.

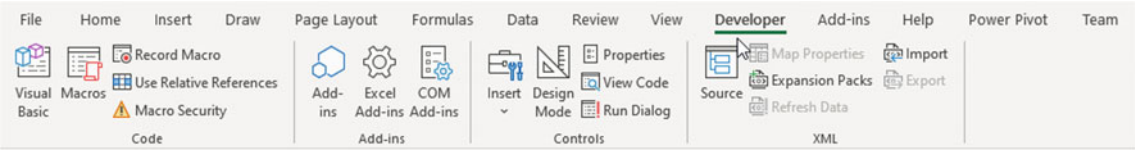
We will create custom functions in Excel to calculate the indexes in this chapter. The Visual Basic for Application or VBA function to calculate the simple price index is shown below.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_19

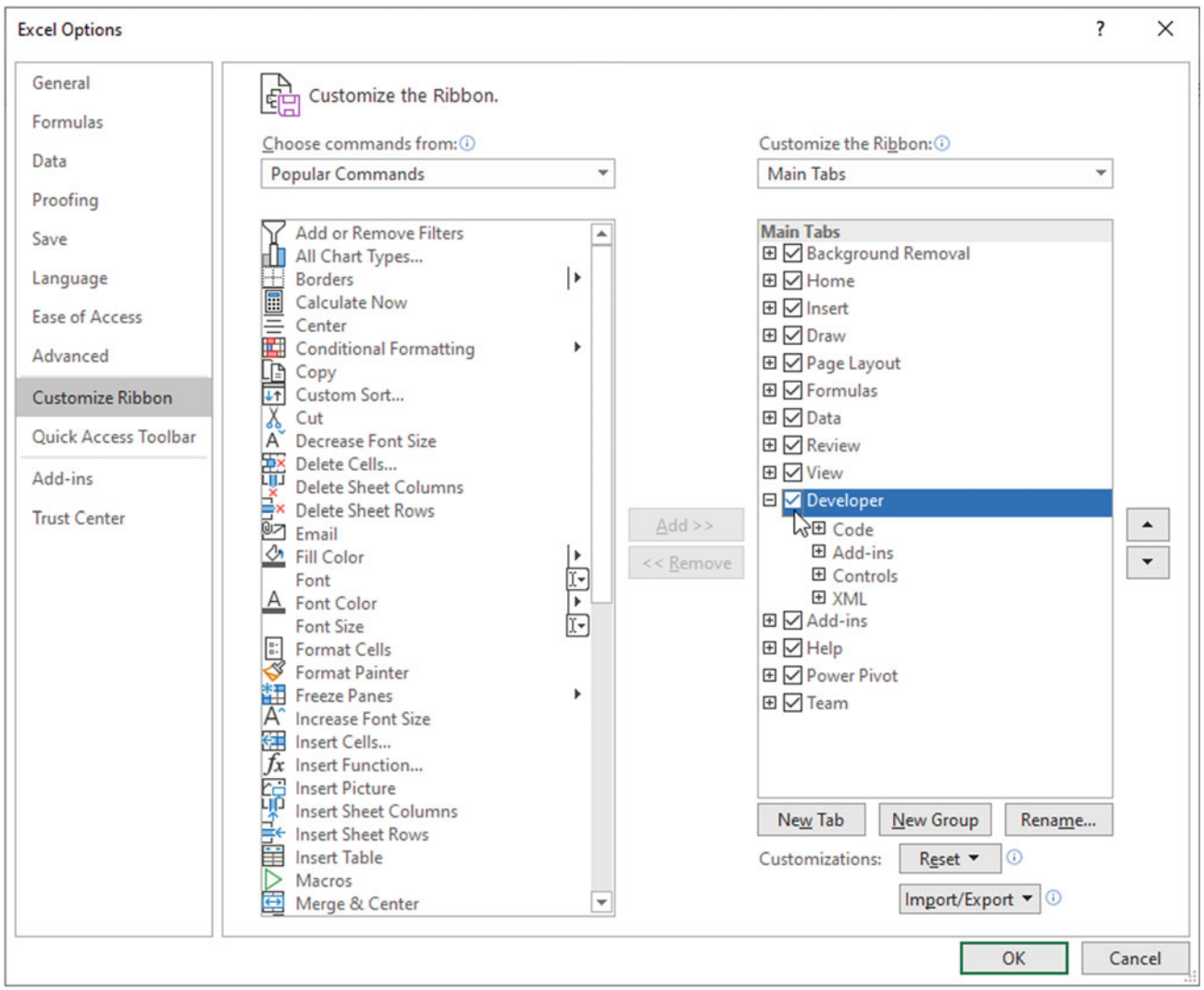
```
'/**************************************************************************************
'/Purpose: Calculates the price index given the current year and base year
'/Parameter: CurrentYear: Range that contains the current year.
'/           BaseYear: Range that contains the base year.
'/**************************************************************************************
Function PriceIndex(CurrentYear As Range, BaseYear As Range) As Double
            PriceIndex = (WorksheetFunction.Sum(CurrentYear) / _
                        WorksheetFunction.Sum(BaseYear)) * 100
End Function
```

When working with VBA code, the *Developer* tab should be visible. The *Developer* tab, by default, is not visible.



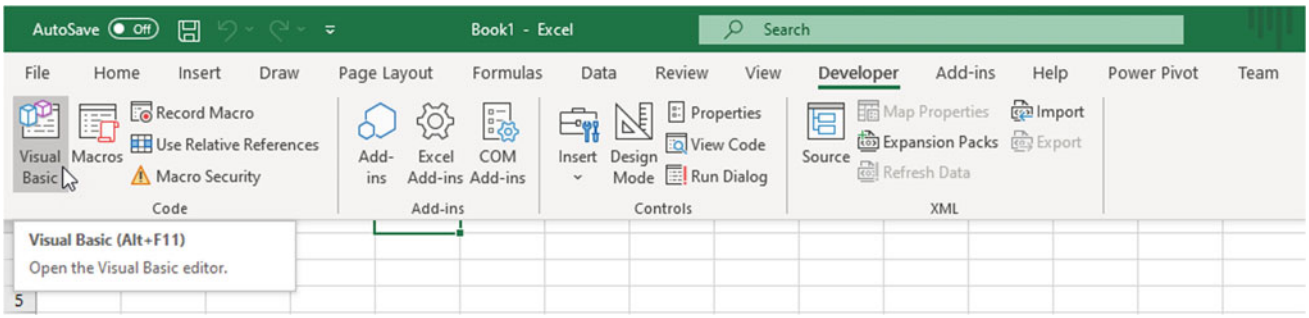
Do the following to show the *Developer* tab.

(1) Choose Files → Options to get to the *Excel Options* screen.

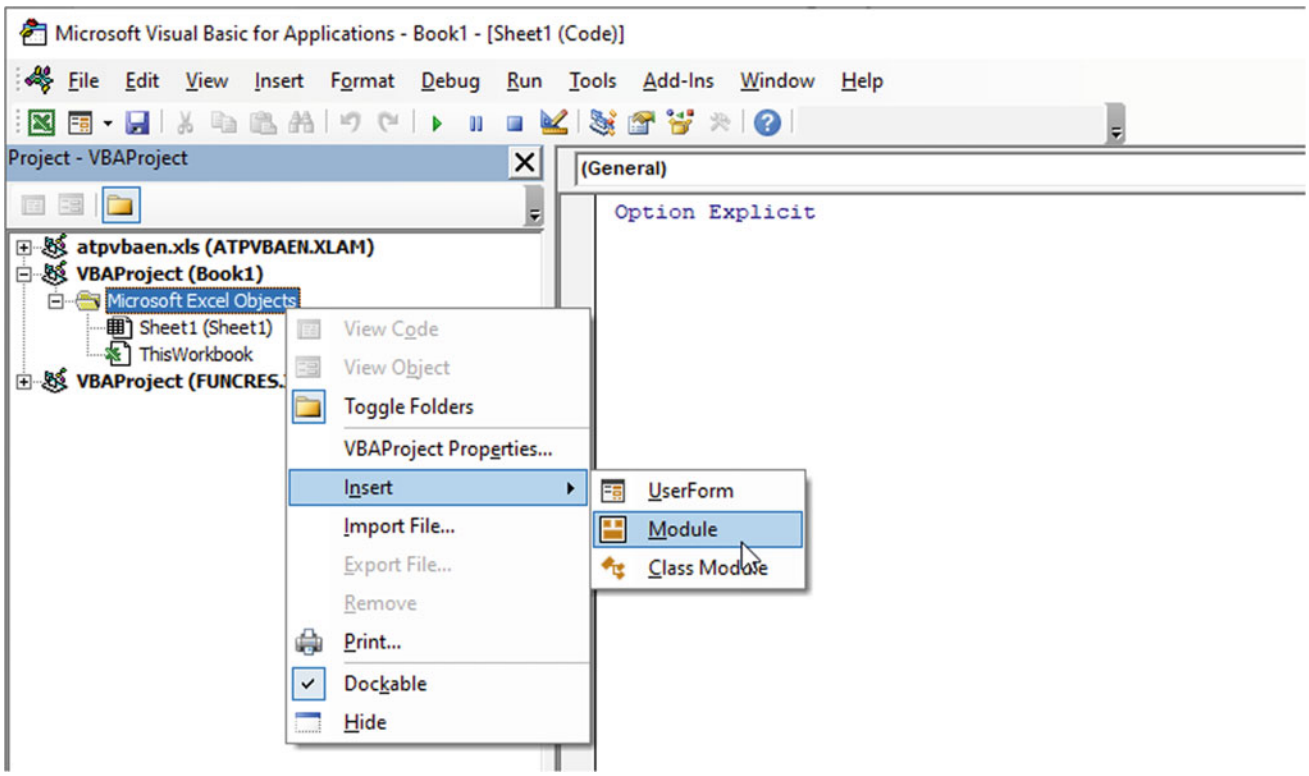


(2) Click on the *Developer* checkbox.

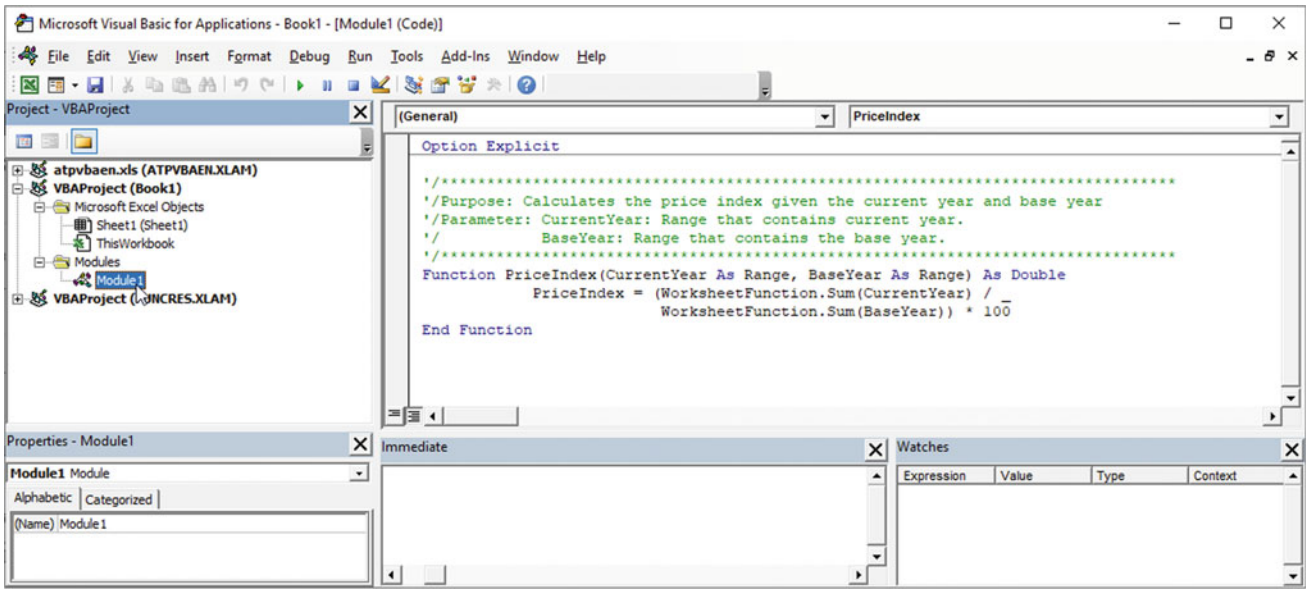
Click on the *Visual Basic* button on the *Developer's* tab to insert the VBA code into the workbook.



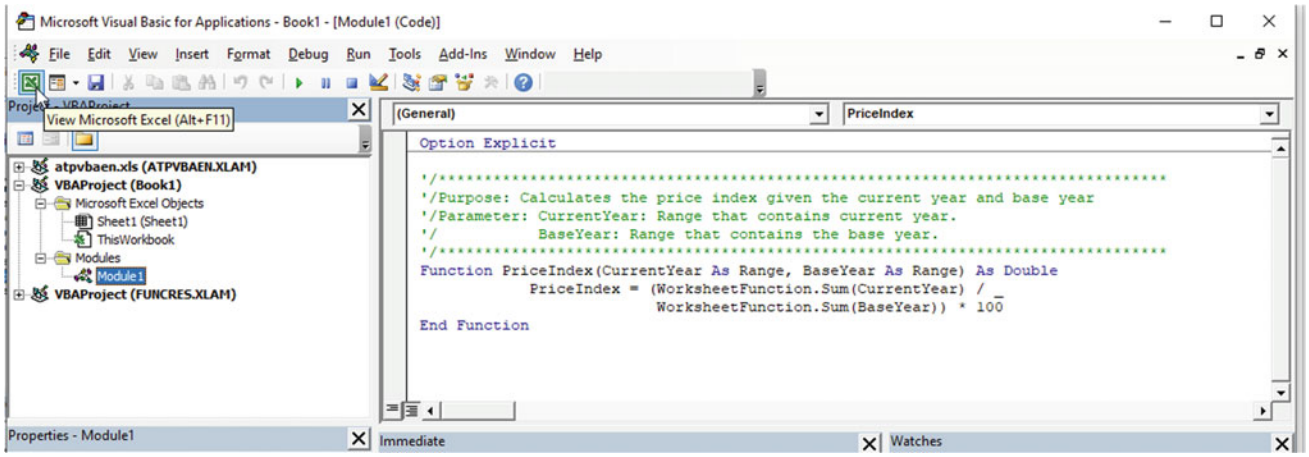
Select *Microsoft Excel Objects* and then click on *Module* as shown below.



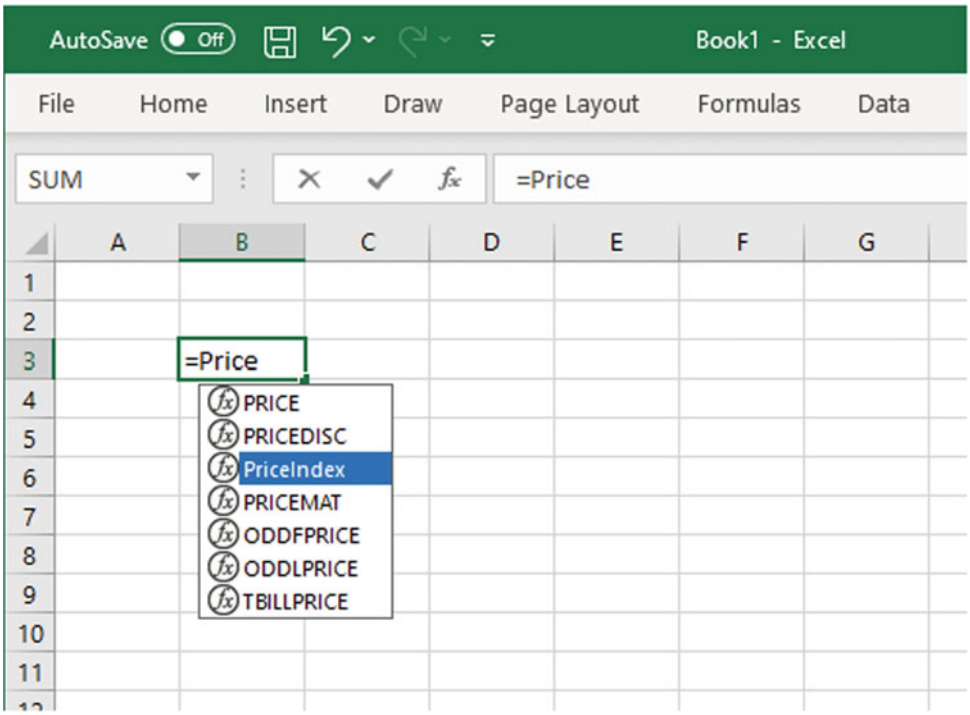
Enter the VBA code in *Module1*.



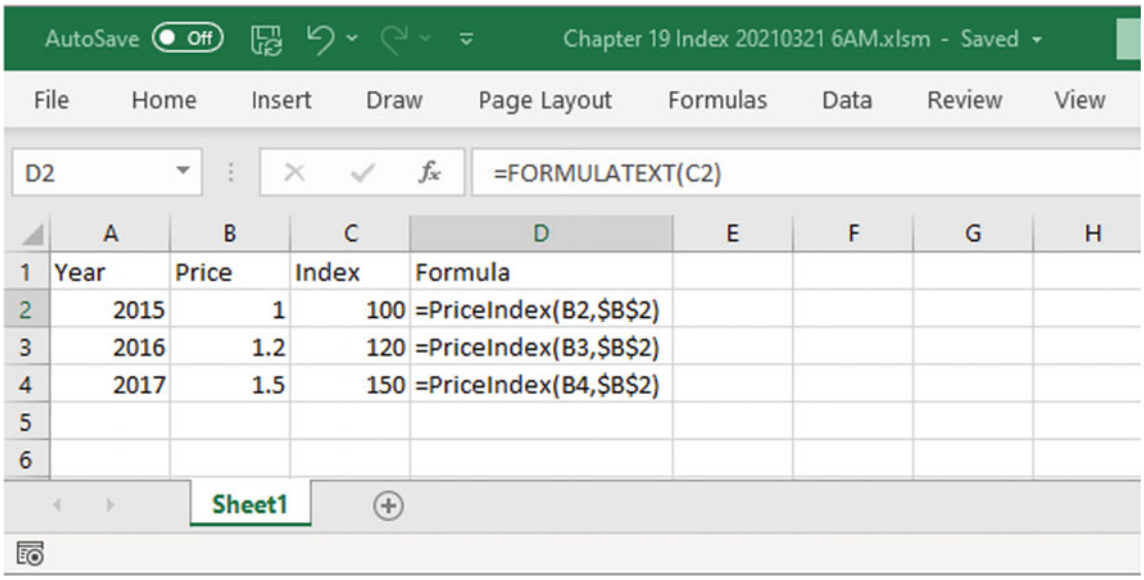
Click on the *Excel* icon to view Excel worksheet.



The *PriceIndex* function will show up in the function drop-down list. Notice that Excel functions are shown in capital letters while the *PriceIndex* is not.



In Excel, input the price of eggs data and enter the *PriceIndex* function as shown below.



The index numbers tell us that the price of eggs in 2016 was 20% greater than it was in 2015, or 120% of the price of eggs in 2015. The index numbers also tell us that the price of eggs in 2017 was 50% greater than it was in 2015, or 150% of the price of eggs in 2015.

19.3 Laspeyres Price Index

Previously, we have only looked at price in our index. The Laspeyres price index takes into consideration both price and quantity. The base period quantity is in both the numerator and the denominator of the Laspeyres index. The formula for the Laspeyres index is

$$I_t = \frac{\sum_{i=1}^{n} P_{ii} Q_{0i}}{\sum_{i=1}^{n} P_{0i} Q_{0i}} * 100$$

where P_{0i} is the base price of item i, Q_{0i} is the base quantity of item i, P_{ti} is the price at time t of item i, and I_t is the Laspeyres price index.

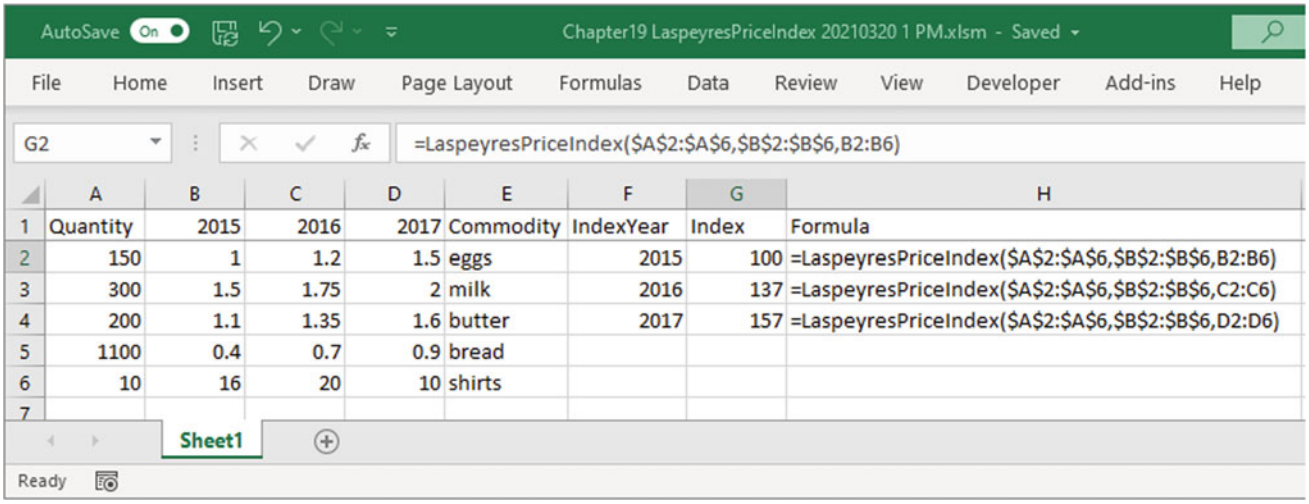
We will use the following items to calculate the Laspeyres price index.

	Eggs		Milk		Butter		Bread	
Year	Price	Quantity	Price	Quantity	Price	Quantity	Price	Quantity
2015	1.00	150	1.50	300	1.1	200	0.4	1100
2016	1.20	160	1.75	250	1.35	180	0.7	1000
2017	1.50	180	2.00	300	1.6	250	0.9	1050

We will now create a custom Laspeyres price index function in Excel. The Excel function code to calculate the Laspeyres price index is shown below.

```
'/**************************************************************************************
'/Purpose: Calculates the Laspeyres price index  given the current year and base year
'/          and the quantity of the base year.
'/Parameter: CurrentYear: Range that contains current year.
'/            BaseYear: Range that contains the base year.
'/            BaseQuantity: Range that contains the base year quantity of the base year.
'/**************************************************************************************
Function LaspeyresPriceIndex(BaseQuantity As Range, BaseYear As Range, _
                             CurrentYear As Range) As Double
         'the sumproduct is a worksheet function
         LaspeyresPriceIndex = (WorksheetFunction.SumProduct(BaseQuantity, CurrentYear)/ _
                               WorksheetFunction.SumProduct(BaseQuantity, BaseYear)) _
                               * 100
End Function
```

Below shows the calculation of the Laspeyres price index in Excel.



	A	B	C	D	E	F	G	H
1	Quantity	2015	2016	2017	Commodity	IndexYear	Index	Formula
2	150	1	1.2	1.5	eggs	2015	100	=LaspeyresPriceIndex($A$2:$A$6,$B$2:$B$6,B2:B6)
3	300	1.5	1.75	2	milk	2016	137	=LaspeyresPriceIndex($A$2:$A$6,$B$2:$B$6,C2:C6)
4	200	1.1	1.35	1.6	butter	2017	157	=LaspeyresPriceIndex($A$2:$A$6,$B$2:$B$6,D2:D6)
5	1100	0.4	0.7	0.9	bread			
6	10	16	20	10	shirts			
7								

From the Excel output, we can see that the 2016 basket of goods is 37% more expensive than the 2015 basket of goods and that the 2017 basket of goods is 57% more expensive than the 2015 basket of goods.

19.4 Paasche Price Index

The Paasche price index is like the Laspeyres index except that it uses the current-year quantities instead of the base-year quantity, this is indicated with Q_{ti} in the denominator of the Paasche price index is

$$I_t = \frac{\sum_{i=1}^{n} P_{ti} Q_{ti}}{\sum_{i=1}^{n} P_{0i} Q_{ti}} * 100$$

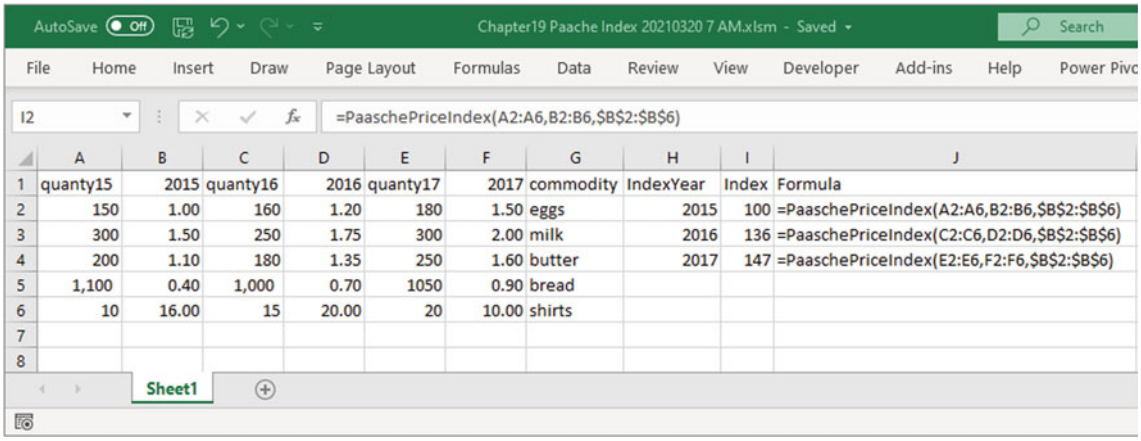
We will use the same data as the Laspeyres price index. It is reproduced below for convenience.

	Eggs		Milk		Butter		Bread	
Year	Price	Quantity	Price	Quantity	Price	Quantity	Price	Quantity
2015	1.00	150	1.50	300	1.1	200	0.4	1100
2016	1.20	160	1.75	250	1.35	180	0.7	1000
2017	1.50	180	2.00	300	1.6	250	0.9	1050

We will now create a custom Paasche price index in Excel. The Excel function code to calculate the Paasche price index is shown below.

```
'/**************************************************************************
'/Purpose: Calculates the Paasche price index given the current year and base year
'/          and the current quantity
'/Parameter: CurrentYear: Range that contains the current year.
'/           BaseYear: Range that contains the base year.
'/           CurrentQuantity: Range that contains the current year quantity.
'/**************************************************************************
Function PaaschePriceIndex(CurrentQuantity As Range, CurrentYear As Range, _
                           BaseYear As Range) As Double
        PaaschePriceIndex = (WorksheetFunction.SumProduct(CurrentQuantity, CurrentYear)/ _
                             WorksheetFunction.SumProduct(CurrentQuantity, BaseYear)) _
                             * 100
End Function
```

Below shows the calculation of the Paasche price index in Excel.



	A	B	C	D	E	F	G	H	I	J
1	quanty15	2015	quanty16	2016	quanty17	2017	commodity	IndexYear	Index	Formula
2	150	1.00	160	1.20	180	1.50	eggs	2015	100	=PaaschePriceIndex(A2:A6,B2:B6,$B$2:$B$6)
3	300	1.50	250	1.75	300	2.00	milk	2016	136	=PaaschePriceIndex(C2:C6,D2:D6,$B$2:$B$6)
4	200	1.10	180	1.35	250	1.60	butter	2017	147	=PaaschePriceIndex(E2:E6,F2:F6,$B$2:$B$6)
5	1,100	0.40	1,000	0.70	1050	0.90	bread			
6	10	16.00	15	20.00	20	10.00	shirts			
7										
8										

From the computer output, we can see that the 2016 price of goods is 36% greater than the price of goods in 2015. We can see that the 2017 price of goods is 47% greater than the price of goods in 2015.

19.5 Fisher's Ideal Price Index

Fisher's ideal price index, which contains both the Laspeyres price index and the Paasche price index, has a value between those two indexes. The mathematical formula for the Fisher's ideal price index is

$$FI = \sqrt{\frac{\sum_{i=1}^{n} P_{ti}Q_{0i}}{\sum_{i=1}^{n} P_{0t}Q_{0i}} * \frac{\sum_{i=1}^{n} P_{ti}Q_{ti}}{\sum_{i=1}^{n} P_{0t}Q_{ti}}}$$

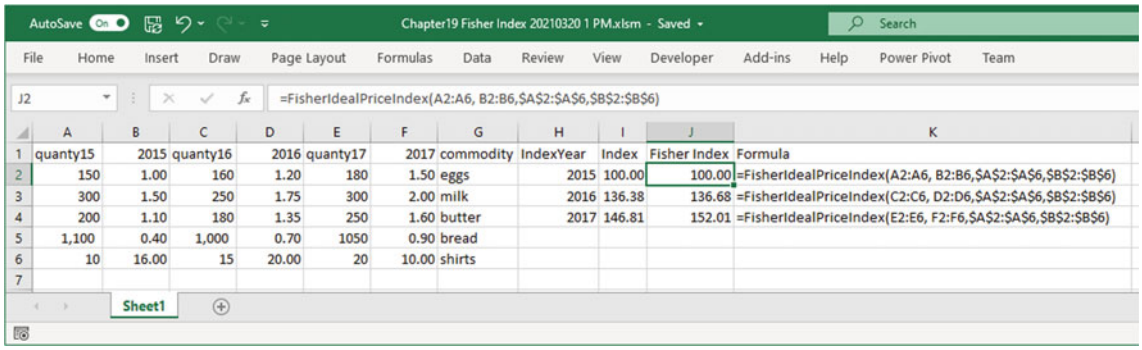
Laspeyres Price Index Paasche Price Index

We will use the same data as the Laspeyres price index. It is reproduced below for convenience.

	Eggs		Milk		Butter		Bread	
Year	Price	Quantity	Price	Quantity	Price	Quantity	Price	Quantity
2015	1.00	150	1.50	300	1.1	200	0.4	1100
2016	1.20	160	1.75	250	1.35	180	0.7	1000
2017	1.50	180	2.00	300	1.6	250	0.9	1050

```
'/**************************************************************************
'/Purpose: Calculates the Fisher Ideal Price Index. Calculates the index by
'         calling the Laspeyres Price Index and Paasche Price Index.
'/Parameter: CurrentYear: Range that contains the current year.
'/           BaseYear: Range that contains the base year.
'/           BaseQuantity: Range that contains the base year quantity of the base year.
'/           CurrentQuantity: Range that contains the current year quantity
'/**************************************************************************
Function FisherIdealPriceIndex(CurrentQuantity As Range, CurrentYear As Range, _
                               BaseQuantity As Range, BaseYear As Range) As Double
 FisherIdealPriceIndex = Sqr(LaspeyresPriceIndex(BaseQuantity, BaseYear, CurrentYear) _
                         * PaaschePriceIndex(CurrentQuantity, CurrentYear, BaseYear))
End Function
```

To illustrate the Fisher Ideal price index, we will use the Paasche price index data that we entered in Excel. We will put the Fisher Index in column J, as shown below.



	A	B	C	D	E	F	G	H	I	J	K
1	quanty15	2015	quanty16	2016	quanty17	2017	commodity	IndexYear	Index	Fisher Index	Formula
2	150	1.00	160	1.20	180	1.50	eggs	2015	100.00	100.00	=FisherIdealPriceIndex(A2:A6, B2:B6,$A$2:$A$6,$B$2:$B$6)
3	300	1.50	250	1.75	300	2.00	milk	2016	136.38	136.68	=FisherIdealPriceIndex(C2:C6, D2:D6,$A$2:$A$6,$B$2:$B$6)
4	200	1.10	180	1.35	250	1.60	butter	2017	146.81	152.01	=FisherIdealPriceIndex(E2:E6, F2:F6,$A$2:$A$6,$B$2:$B$6)
5	1,100	0.40	1,000	0.70	1050	0.90	bread				
6	10	16.00	15	20.00	20	10.00	shirts				
7											

From the Excel output, we can see that the price of goods in 2016 is 36% greater than the price of goods in 2015. We can also see that the price of goods in 2017 is 52% greater than the price of goods in 2018.

19.6 Stock Indexes: S&P500 Index and NASDAQ Composite Index

We will now look at three "headliner" stock indexes. The three "headliner" stock indexes are the Dow Jones Industrial Average, S&P 500 Index, and the NASDAQ Composite Index. These three indexes are usually reported together.

A stock market index is a statistical measure that shows how the price of a group of stocks changes over time. A stock market index encompasses either all or only a portion of stocks in its market. Stock market indexes employ different weighting schemes, so we can use this basis to categorize the indexes by type. The three most common types of stock market indexes are market-value-weighted indexes, price-weighted indexes, and equally weighted indexes. Price per share in the current period (P_0), price per share in the next period (P_1), number of shares outstanding in the current period (Q_0), and number of shares outstanding in the next period (Q_1) are listed below. These data are used to illustrate the various weighting schemes and to provide information about the weights applied to the market stock indexes.

Price Per Share and Outstanding Shares for Stocks, A,B,C

Stock	P_0	P_1	Q_0	Q_1
A	100	100	60	55
B	50	60	90	100
C	20	40	250	300

The first stock market index we will look at is the *market-value-weighted index,* which is shown below.

$$I = \frac{\sum_{i=1}^{n} Q_{ti} P_{ti}}{\sum_{i=}^{n} Q_{0i} P_{0i}}$$

The following is the market-value-weighted index of the table above,

$$I = \frac{55(100) + 100(60) + 300(40)}{60(100) + 90(50) + 250(20)}(100)$$
$$= \frac{23{,}500}{15{,}500}(100) = 152$$

The market-value-weighted index implies that there was a 52% increase from the base period to the current period.

Standard and Poor's 500 Composite Index is an example of a market-value-weighted index. The S&P 500 index comprises industrial firms, utilities, transportation firms, and financial firms. S&P 500 tracks the 500 largest publicly traded companies by market cap. Changes in the index are based on changes in the firm's total market value with respect to a base year. Currently, the base period (1941–1943 = 10) for the S&P 500 index is stated formally as follows:

$$S\&P\,500\,index = \frac{\sum_{i=1}^{n} Q_{ti}P_{ti}}{\sum_{i=}^{n} Q_{0i}P_{0i}}(10)$$

The NASDAQ Composite Index is also another example of a market-value-weighted index. One thing that the NASDAQ is known for is it has a high concentration of technology stocks.

19.7 Stock Indexes: Dow Jones Industrial Average (DJIA)

The next stock market index that we will look at is the *price-weighted index*. The price-weighted index shows the change in the average price of the stocks that are included in the index. Using the table above, we can compute the price-weighted index as follows:

$$\begin{aligned} I &= \frac{(100+60+40)/3}{(100+50+20)/3}(100) \\ &= \frac{200/3}{170/3}(100) = 117.65 \end{aligned}$$

The closest thing to a true price-weighted stock market index is the *Dow Jones Industrial Average (DJIA).* Simply stated, the DJIA is an arithmetic average of the stock prices that make up the index. The DJIA originally assumed a single share of each stock in the index. The DJIA originally assumed a single share of each stock in the index, and the total of stock prices was divided by the number of stocks that made up of the index:

$$DJIA_t = \frac{\frac{\sum_{i=1}^{30} P_{ti}}{30}}{\frac{\sum_{i=1}^{30} P_{0i}}{30}}$$

Today the index is adjusted for stock splits and the issuance of stock dividends:

$$DJIA_t = \frac{\frac{\sum_{i=1}^{30} \frac{P_{ti}}{AD_t}}{30}}{\frac{\sum_{i=1}^{30} P_{0i}}{30}}$$

where P_{ti} = the closing price of stock i on day t, and AD_t = the adjusted divisor on the day t. The adjustment process is shown below.

Adjustment of DJIA Divisor to Allow for a Stock Split

	Before Split	After 2-for-1 Stock Split by Stock A
Stock	*Price*	*Price*
A	60	30
B	30	30
C	20	20
D	10	10
	120	90

Average before split = 120/4 = 30

Adjust divisor = (sum of prices after the split)/(Average before split) = 90/30 = 3

Average after split = 90/30 = 30

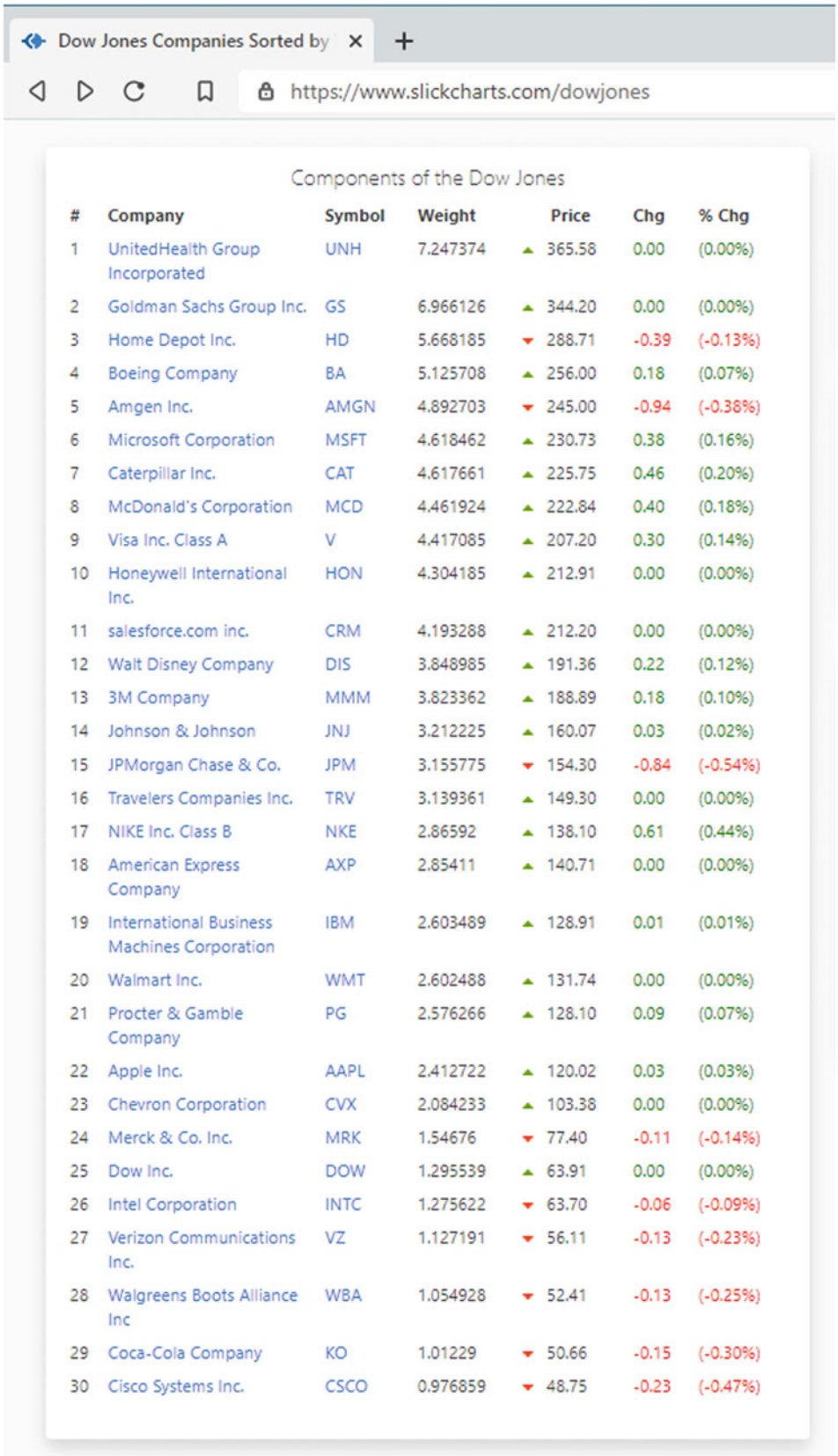
Before split divisor = 4 After split divisor = 30

As shown above, the adjustment process is designed to keep the index value the same it would have been if the split had not occurred. Similar adjustments have been made when it has been found necessary to replace one of the component stocks with the stock of another company, thus preserving the consistency and the comparability of index values at different points in time.

Nevertheless, the adjustment process used for the DJIA has its share of critics. Because price weighting itself causes high-priced stocks to dominate the series, the same effect can cause a shift in this balance when rapidly growing firms split their stock. For example, a 20% increase in the price of stock A in the table above would in itself have caused a 10% increase in the value of the sample index before the split, whereas a 20% increase in the price of stock B would have caused only a 5% increase in the index value. After the 2-for-1 split of stock A, a 20% increase in either stock A or stock B would have the same effect on the index value (a 6.7% increase); a downward shift in the importance of stock A relative to that of the other stocks in the sample has occurred. This effect would relegate the stock of the fastest-growing companies to a position of *least* importance in determining index values.

19.8 Components of the Dow Jones Industrial Average (DJIA)

On March 19, 2021, the following URL list all the 30 stock components of the DJIA, https://www.slickcharts.com/dowjones.

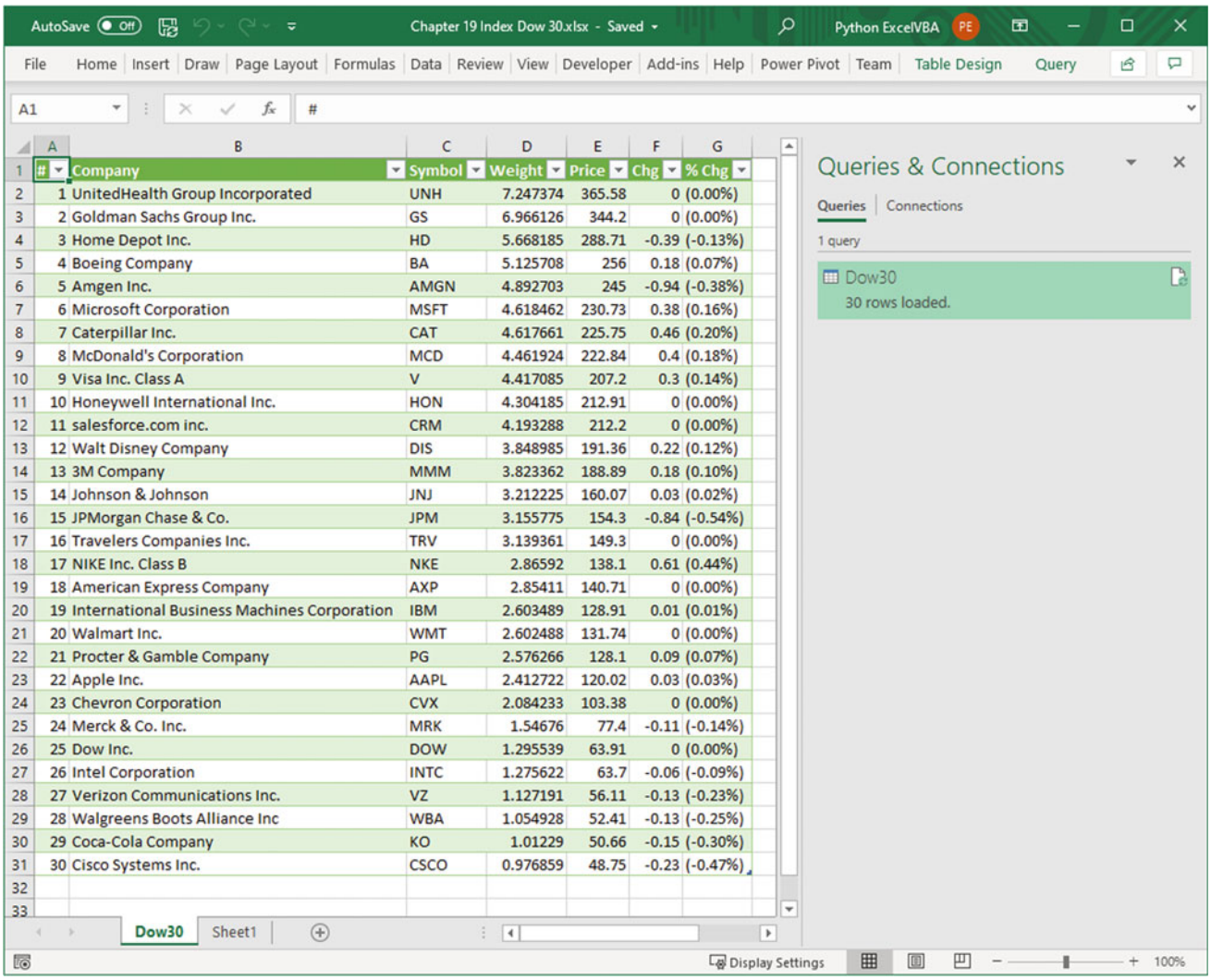


Components of the Dow Jones

#	Company	Symbol	Weight	Price	Chg	% Chg
1	UnitedHealth Group Incorporated	UNH	7.247374	365.58	0.00	(0.00%)
2	Goldman Sachs Group Inc.	GS	6.966126	344.20	0.00	(0.00%)
3	Home Depot Inc.	HD	5.668185	288.71	-0.39	(-0.13%)
4	Boeing Company	BA	5.125708	256.00	0.18	(0.07%)
5	Amgen Inc.	AMGN	4.892703	245.00	-0.94	(-0.38%)
6	Microsoft Corporation	MSFT	4.618462	230.73	0.38	(0.16%)
7	Caterpillar Inc.	CAT	4.617661	225.75	0.46	(0.20%)
8	McDonald's Corporation	MCD	4.461924	222.84	0.40	(0.18%)
9	Visa Inc. Class A	V	4.417085	207.20	0.30	(0.14%)
10	Honeywell International Inc.	HON	4.304185	212.91	0.00	(0.00%)
11	salesforce.com inc.	CRM	4.193288	212.20	0.00	(0.00%)
12	Walt Disney Company	DIS	3.848985	191.36	0.22	(0.12%)
13	3M Company	MMM	3.823362	188.89	0.18	(0.10%)
14	Johnson & Johnson	JNJ	3.212225	160.07	0.03	(0.02%)
15	JPMorgan Chase & Co.	JPM	3.155775	154.30	-0.84	(-0.54%)
16	Travelers Companies Inc.	TRV	3.139361	149.30	0.00	(0.00%)
17	NIKE Inc. Class B	NKE	2.86592	138.10	0.61	(0.44%)
18	American Express Company	AXP	2.85411	140.71	0.00	(0.00%)
19	International Business Machines Corporation	IBM	2.603489	128.91	0.01	(0.01%)
20	Walmart Inc.	WMT	2.602488	131.74	0.00	(0.00%)
21	Procter & Gamble Company	PG	2.576266	128.10	0.09	(0.07%)
22	Apple Inc.	AAPL	2.412722	120.02	0.03	(0.03%)
23	Chevron Corporation	CVX	2.084233	103.38	0.00	(0.00%)
24	Merck & Co. Inc.	MRK	1.54676	77.40	-0.11	(-0.14%)
25	Dow Inc.	DOW	1.295539	63.91	0.00	(0.00%)
26	Intel Corporation	INTC	1.275622	63.70	-0.06	(-0.09%)
27	Verizon Communications Inc.	VZ	1.127191	56.11	-0.13	(-0.23%)
28	Walgreens Boots Alliance Inc	WBA	1.054928	52.41	-0.13	(-0.25%)
29	Coca-Cola Company	KO	1.01229	50.66	-0.15	(-0.30%)
30	Cisco Systems Inc.	CSCO	0.976859	48.75	-0.23	(-0.47%)

19.8.1 Using Power Query to Retrieve the Dow 30 Components

The following is Excel's Power Query M Code to retrieve the Dow 30 components.



#	Company	Symbol	Weight	Price	Chg	% Chg
1	UnitedHealth Group Incorporated	UNH	7.247374	365.58	0	(0.00%)
2	Goldman Sachs Group Inc.	GS	6.966126	344.2	0	(0.00%)
3	Home Depot Inc.	HD	5.668185	288.71	-0.39	(-0.13%)
4	Boeing Company	BA	5.125708	256	0.18	(0.07%)
5	Amgen Inc.	AMGN	4.892703	245	-0.94	(-0.38%)
6	Microsoft Corporation	MSFT	4.618462	230.73	0.38	(0.16%)
7	Caterpillar Inc.	CAT	4.617661	225.75	0.46	(0.20%)
8	McDonald's Corporation	MCD	4.461924	222.84	0.4	(0.18%)
9	Visa Inc. Class A	V	4.417085	207.2	0.3	(0.14%)
10	Honeywell International Inc.	HON	4.304185	212.91	0	(0.00%)
11	salesforce.com inc.	CRM	4.193288	212.2	0	(0.00%)
12	Walt Disney Company	DIS	3.848985	191.36	0.22	(0.12%)
13	3M Company	MMM	3.823362	188.89	0.18	(0.10%)
14	Johnson & Johnson	JNJ	3.212225	160.07	0.03	(0.02%)
15	JPMorgan Chase & Co.	JPM	3.155775	154.3	-0.84	(-0.54%)
16	Travelers Companies Inc.	TRV	3.139361	149.3	0	(0.00%)
17	NIKE Inc. Class B	NKE	2.86592	138.1	0.61	(0.44%)
18	American Express Company	AXP	2.85411	140.71	0	(0.00%)
19	International Business Machines Corporation	IBM	2.603489	128.91	0.01	(0.01%)
20	Walmart Inc.	WMT	2.602488	131.74	0	(0.00%)
21	Procter & Gamble Company	PG	2.576266	128.1	0.09	(0.07%)
22	Apple Inc.	AAPL	2.412722	120.02	0.03	(0.03%)
23	Chevron Corporation	CVX	2.084233	103.38	0	(0.00%)
24	Merck & Co. Inc.	MRK	1.54676	77.4	-0.11	(-0.14%)
25	Dow Inc.	DOW	1.295539	63.91	0	(0.00%)
26	Intel Corporation	INTC	1.275622	63.7	-0.06	(-0.09%)
27	Verizon Communications Inc.	VZ	1.127191	56.11	-0.13	(-0.23%)
28	Walgreens Boots Alliance Inc	WBA	1.054928	52.41	-0.13	(-0.25%)
29	Coca-Cola Company	KO	1.01229	50.66	-0.15	(-0.30%)
30	Cisco Systems Inc.	CSCO	0.976859	48.75	-0.23	(-0.47%)

19.9 Components of the S&P 500 Index

The CNN Business website lists the 500 plus stocks that are in the S&P 500 index. Instead of listing all the stocks in one big table, the list is in 34 tables. Every table shows 15 stocks. Table 34 shows nine stocks. The list across 34 tables is sorted by description in ascending order. The first URL table for March 19, 2021, is https://money.cnn.com/quote/quote.html?symb=spx&page=1.

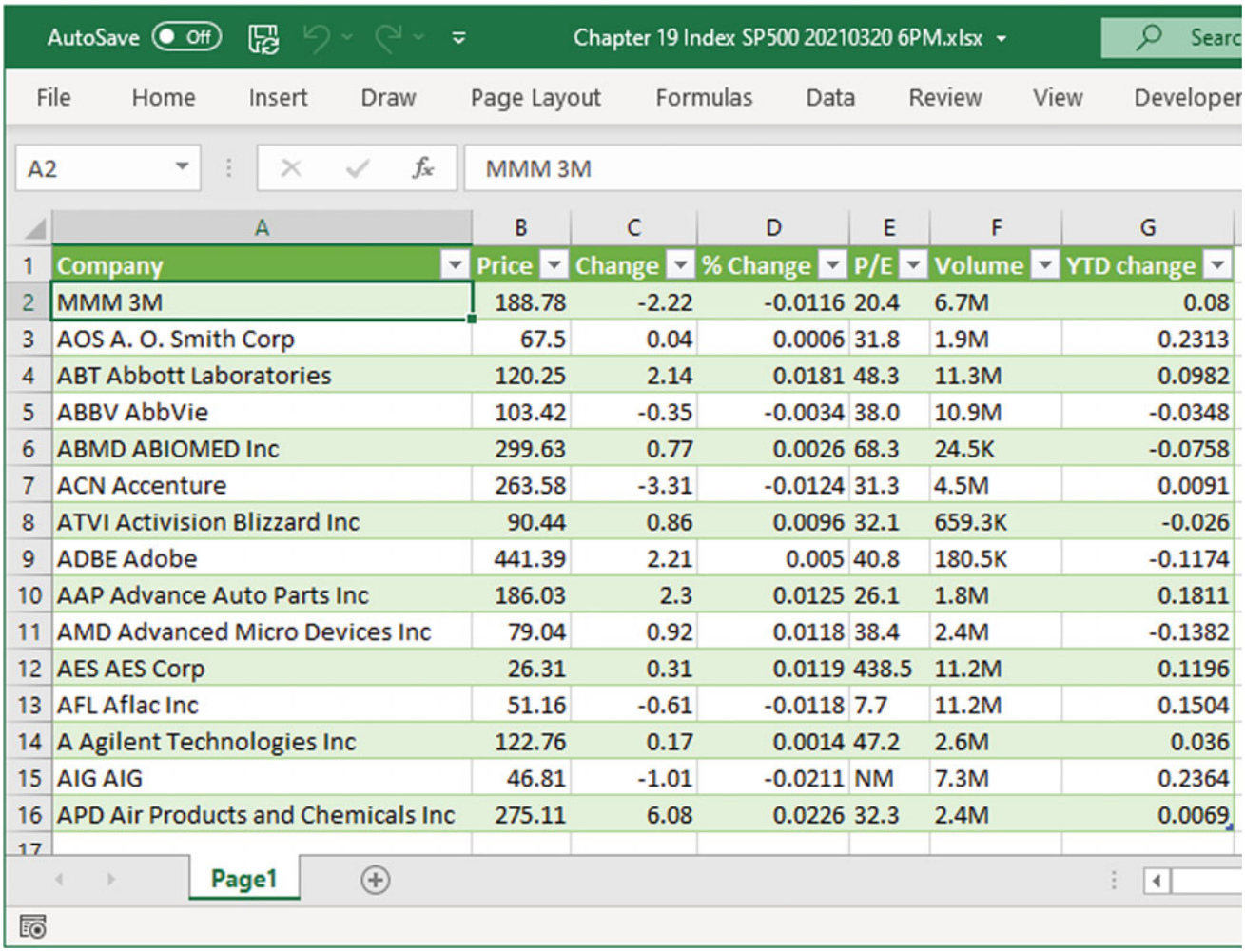
The second URL table is URL https://money.cnn.com/quote/quote.html?symb=spx&page=2.

The last URL table is https://money.cnn.com/quote/quote.html?symb=spx&page=34.

19.9.1 Using Power Query to Retrieve the S&P 500 Components

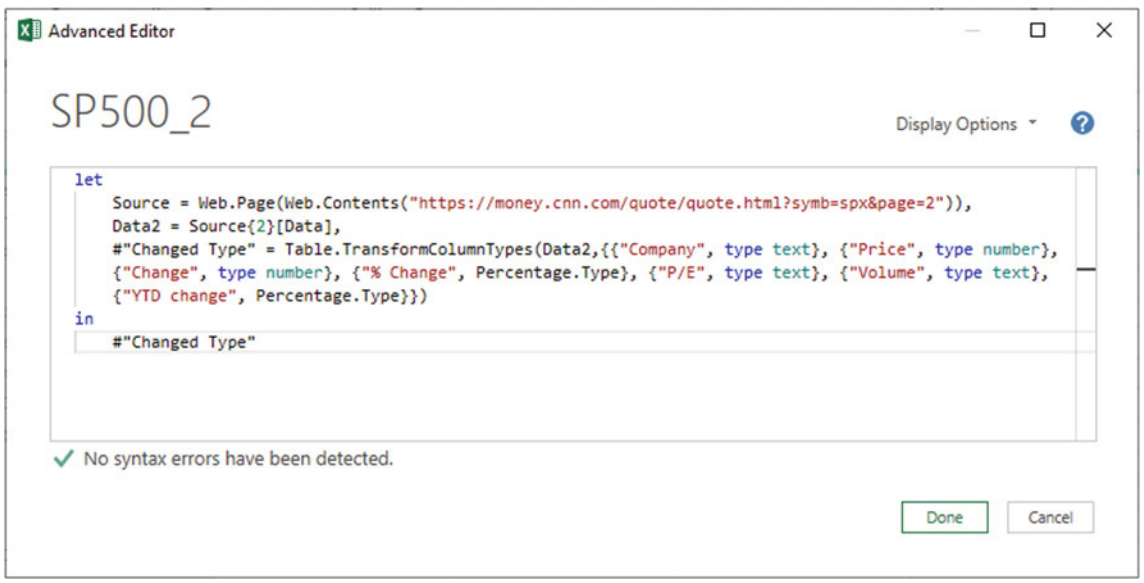
Notice the pattern among the URL shown. The difference among the URLs is the page number shown.

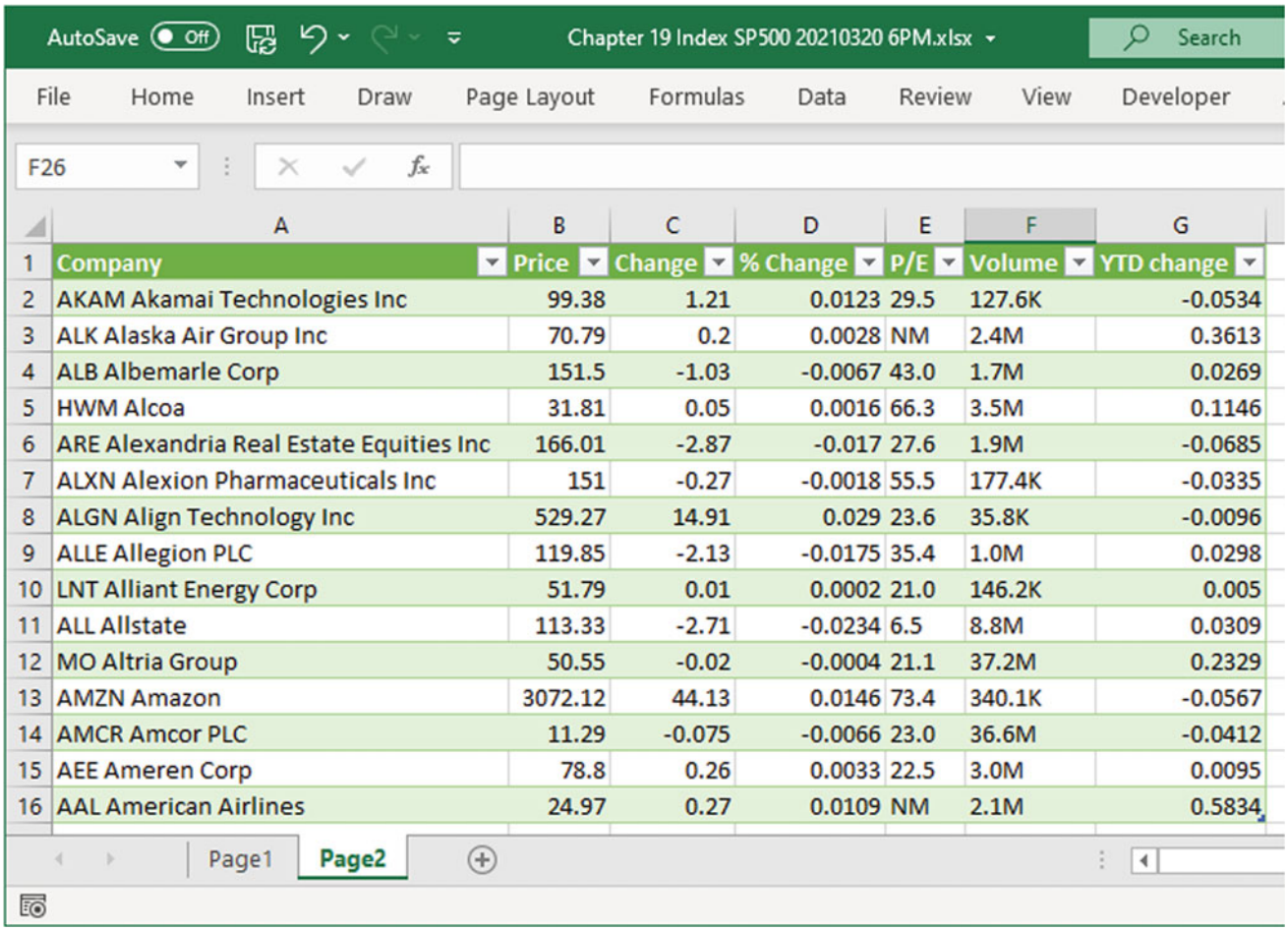
The following is Excel's power query M code to retrieve the first page.



	A	B	C	D	E	F	G
1	Company	Price	Change	% Change	P/E	Volume	YTD change
2	MMM 3M	188.78	-2.22	-0.0116	20.4	6.7M	0.08
3	AOS A. O. Smith Corp	67.5	0.04	0.0006	31.8	1.9M	0.2313
4	ABT Abbott Laboratories	120.25	2.14	0.0181	48.3	11.3M	0.0982
5	ABBV AbbVie	103.42	-0.35	-0.0034	38.0	10.9M	-0.0348
6	ABMD ABIOMED Inc	299.63	0.77	0.0026	68.3	24.5K	-0.0758
7	ACN Accenture	263.58	-3.31	-0.0124	31.3	4.5M	0.0091
8	ATVI Activision Blizzard Inc	90.44	0.86	0.0096	32.1	659.3K	-0.026
9	ADBE Adobe	441.39	2.21	0.005	40.8	180.5K	-0.1174
10	AAP Advance Auto Parts Inc	186.03	2.3	0.0125	26.1	1.8M	0.1811
11	AMD Advanced Micro Devices Inc	79.04	0.92	0.0118	38.4	2.4M	-0.1382
12	AES AES Corp	26.31	0.31	0.0119	438.5	11.2M	0.1196
13	AFL Aflac Inc	51.16	-0.61	-0.0118	7.7	11.2M	0.1504
14	A Agilent Technologies Inc	122.76	0.17	0.0014	47.2	2.6M	0.036
15	AIG AIG	46.81	-1.01	-0.0211	NM	7.3M	0.2364
16	APD Air Products and Chemicals Inc	275.11	6.08	0.0226	32.3	2.4M	0.0069
17							

The following is Excel's power query M code to retrieve the second page.





AutoSave Off | Chapter 19 Index SP500 20210320 6PM.xlsx | Search

File Home Insert Draw Page Layout Formulas Data Review View Developer

F26

	A	B	C	D	E	F	G
1	Company	Price	Change	% Change	P/E	Volume	YTD change
2	AKAM Akamai Technologies Inc	99.38	1.21	0.0123	29.5	127.6K	-0.0534
3	ALK Alaska Air Group Inc	70.79	0.2	0.0028	NM	2.4M	0.3613
4	ALB Albemarle Corp	151.5	-1.03	-0.0067	43.0	1.7M	0.0269
5	HWM Alcoa	31.81	0.05	0.0016	66.3	3.5M	0.1146
6	ARE Alexandria Real Estate Equities Inc	166.01	-2.87	-0.017	27.6	1.9M	-0.0685
7	ALXN Alexion Pharmaceuticals Inc	151	-0.27	-0.0018	55.5	177.4K	-0.0335
8	ALGN Align Technology Inc	529.27	14.91	0.029	23.6	35.8K	-0.0096
9	ALLE Allegion PLC	119.85	-2.13	-0.0175	35.4	1.0M	0.0298
10	LNT Alliant Energy Corp	51.79	0.01	0.0002	21.0	146.2K	0.005
11	ALL Allstate	113.33	-2.71	-0.0234	6.5	8.8M	0.0309
12	MO Altria Group	50.55	-0.02	-0.0004	21.1	37.2M	0.2329
13	AMZN Amazon	3072.12	44.13	0.0146	73.4	340.1K	-0.0567
14	AMCR Amcor PLC	11.29	-0.075	-0.0066	23.0	36.6M	-0.0412
15	AEE Ameren Corp	78.8	0.26	0.0033	22.5	3.0M	0.0095
16	AAL American Airlines	24.97	0.27	0.0109	NM	2.1M	0.5834

Page1 Page2

The following is Excel's power query M code to retrieve the 34th page.

Advanced Editor

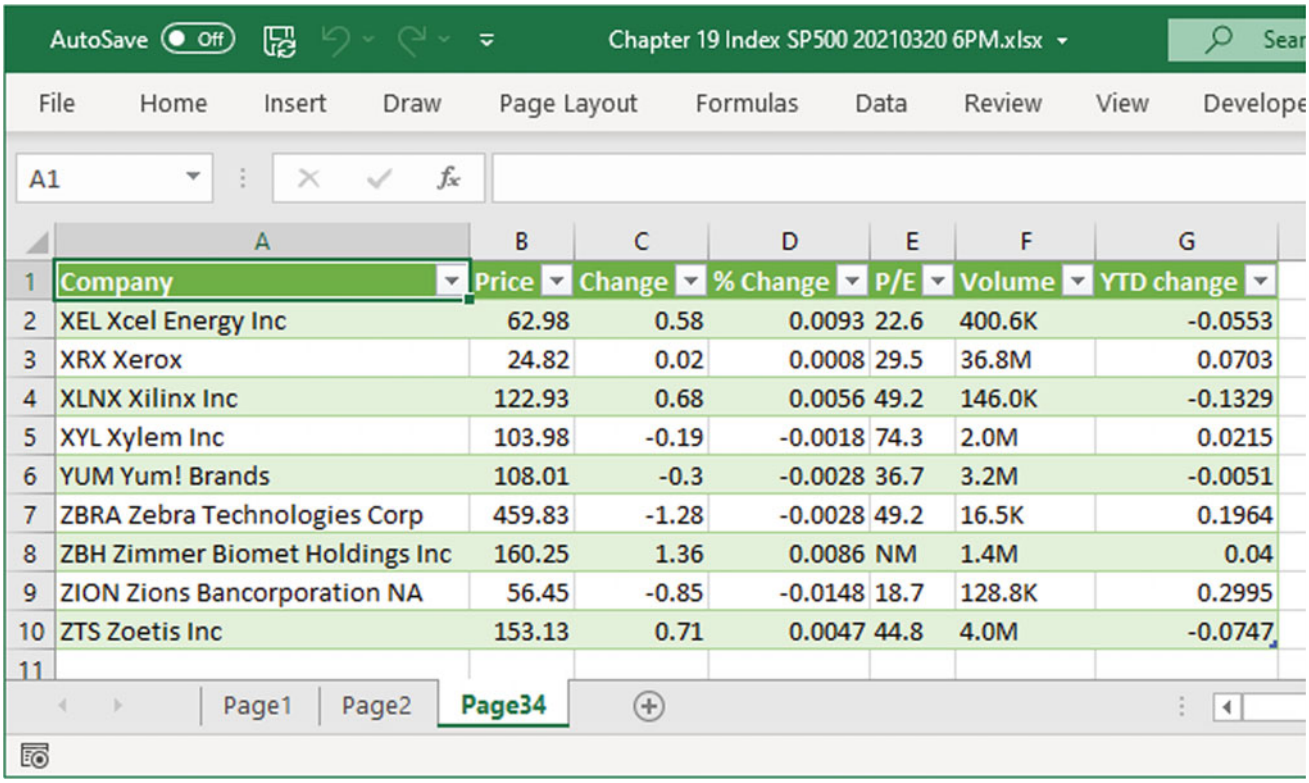
SP500_34

Display Options

```
let
    Source = Web.Page(Web.Contents("https://money.cnn.com/quote/quote.html?symb=spx&page=34")),
    Data2 = Source{2}[Data],
    #"Changed Type" = Table.TransformColumnTypes(Data2,{{"Company", type text}, {"Price", type number},
    {"Change", type number}, {"% Change", Percentage.Type}, {"P/E", type text}, {"Volume", type text},
    {"YTD change", Percentage.Type}})
in
    #"Changed Type"
```

No syntax errors have been detected.

Done Cancel

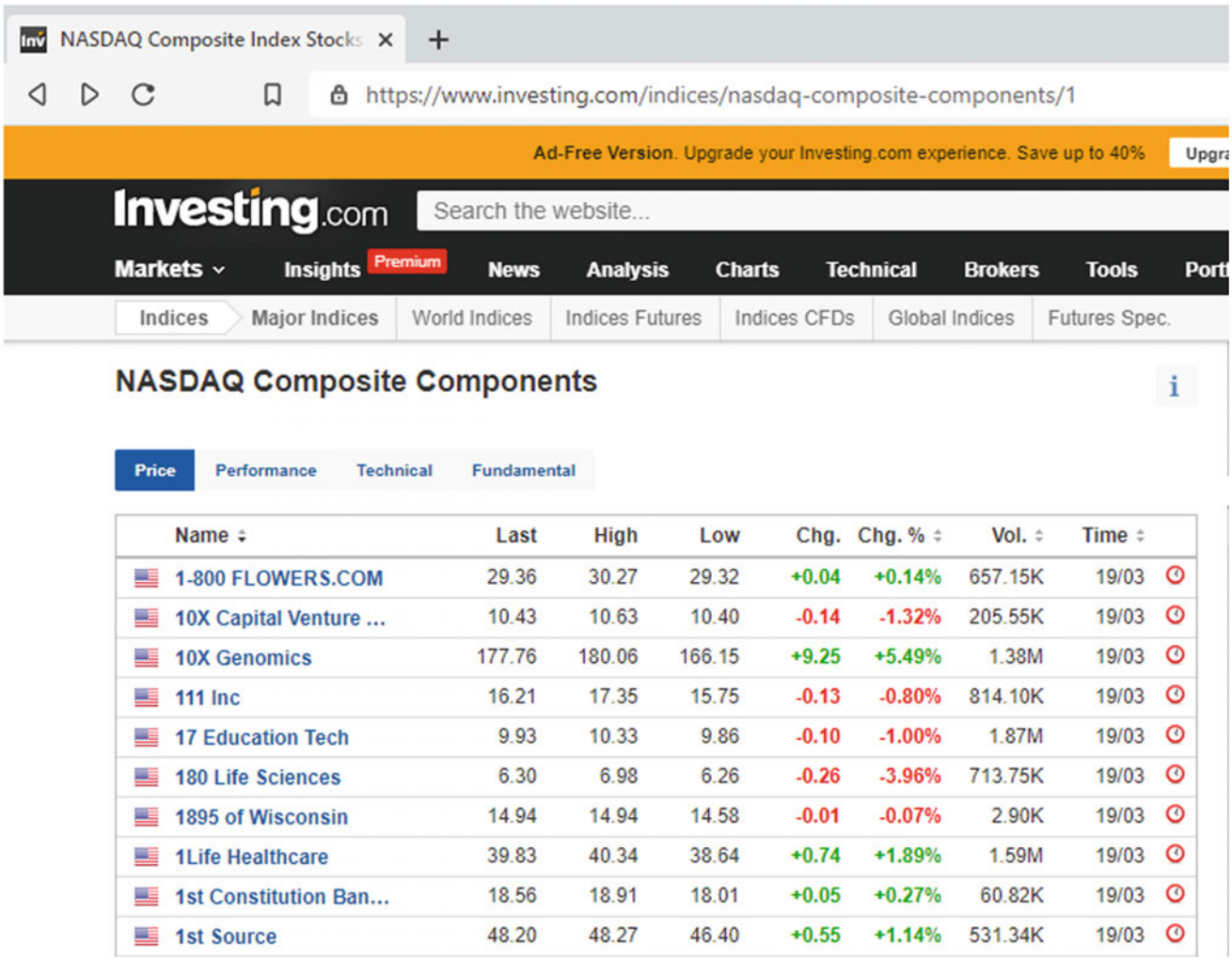


	A	B	C	D	E	F	G
1	Company	Price	Change	% Change	P/E	Volume	YTD change
2	XEL Xcel Energy Inc	62.98	0.58	0.0093	22.6	400.6K	-0.0553
3	XRX Xerox	24.82	0.02	0.0008	29.5	36.8M	0.0703
4	XLNX Xilinx Inc	122.93	0.68	0.0056	49.2	146.0K	-0.1329
5	XYL Xylem Inc	103.98	-0.19	-0.0018	74.3	2.0M	0.0215
6	YUM Yum! Brands	108.01	-0.3	-0.0028	36.7	3.2M	-0.0051
7	ZBRA Zebra Technologies Corp	459.83	-1.28	-0.0028	49.2	16.5K	0.1964
8	ZBH Zimmer Biomet Holdings Inc	160.25	1.36	0.0086	NM	1.4M	0.04
9	ZION Zions Bancorporation NA	56.45	-0.85	-0.0148	18.7	128.8K	0.2995
10	ZTS Zoetis Inc	153.13	0.71	0.0047	44.8	4.0M	-0.0747
11							

The only difference among the M code is the page number indicated.

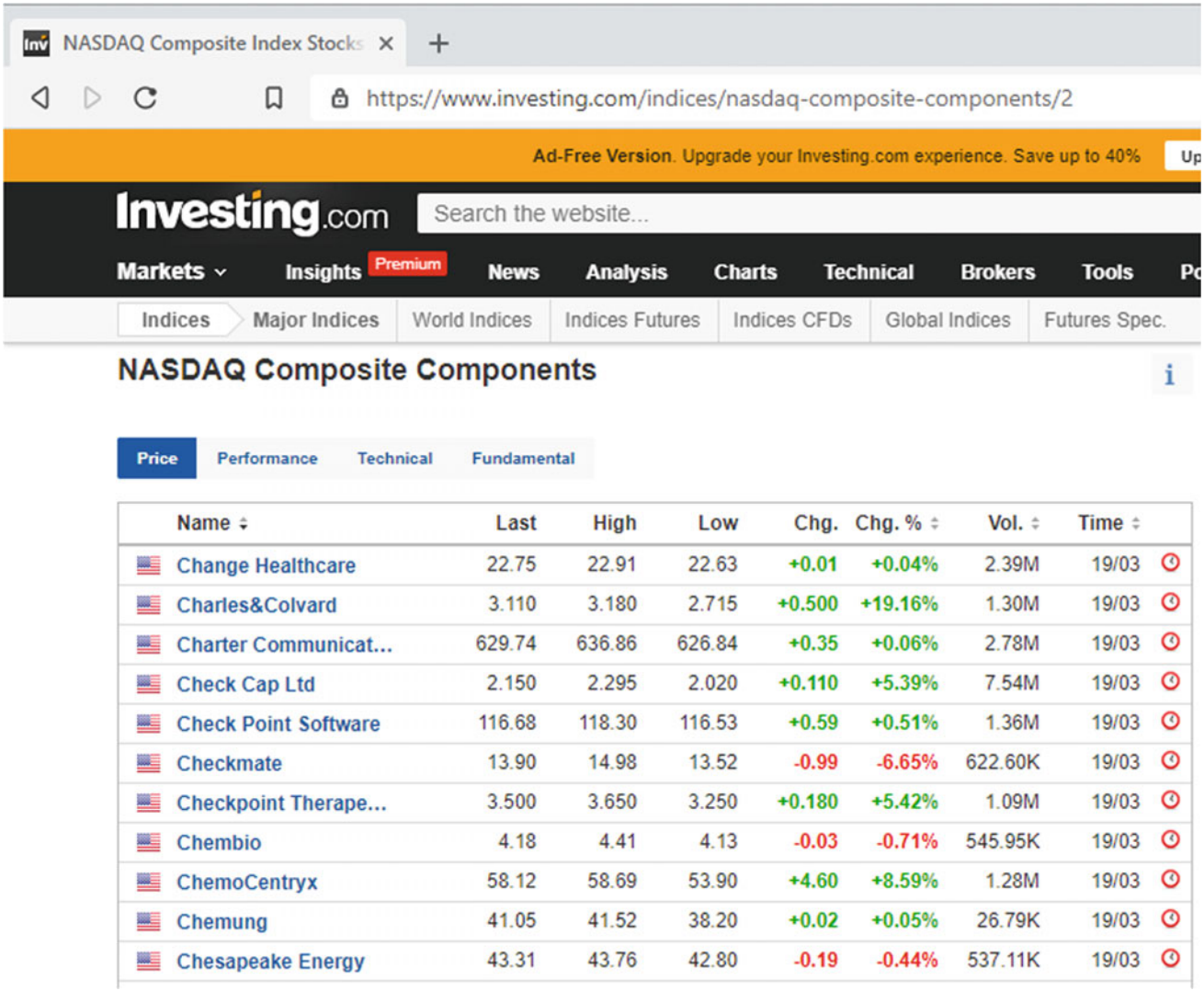
19.10 Components of the NASDAQ Composite Index

The Investing.com website lists the 3,000 plus stocks that are in the NASDAQ composite index. Instead of listing all the stocks in one big table, the list is in six tables. Every table shows 600 stocks. Table 6 shows nine stocks. The list across the six tables is sorted by description in alphabetical order. The first URL table for March 19, 2021 is https://www.investing.com/indices/nasdaq-composite-components/1.



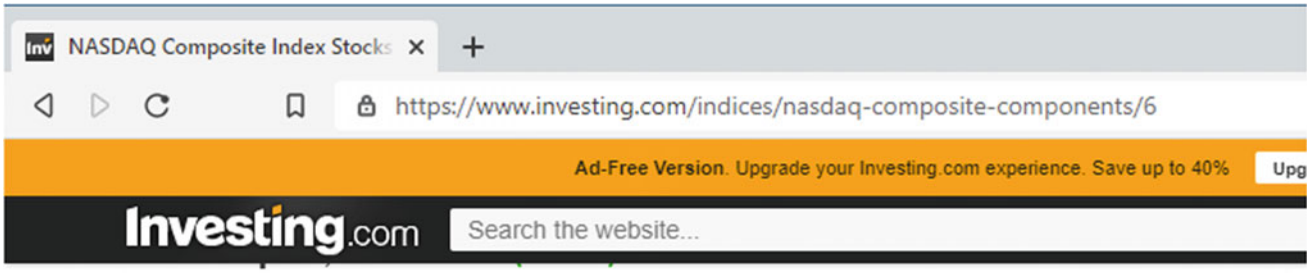
Name	Last	High	Low	Chg.	Chg. %	Vol.	Time
1-800 FLOWERS.COM	29.36	30.27	29.32	+0.04	+0.14%	657.15K	19/03
10X Capital Venture ...	10.43	10.63	10.40	-0.14	-1.32%	205.55K	19/03
10X Genomics	177.76	180.06	166.15	+9.25	+5.49%	1.38M	19/03
111 Inc	16.21	17.35	15.75	-0.13	-0.80%	814.10K	19/03
17 Education Tech	9.93	10.33	9.86	-0.10	-1.00%	1.87M	19/03
180 Life Sciences	6.30	6.98	6.26	-0.26	-3.96%	713.75K	19/03
1895 of Wisconsin	14.94	14.94	14.58	-0.01	-0.07%	2.90K	19/03
1Life Healthcare	39.83	40.34	38.64	+0.74	+1.89%	1.59M	19/03
1st Constitution Ban...	18.56	18.91	18.01	+0.05	+0.27%	60.82K	19/03
1st Source	48.20	48.27	46.40	+0.55	+1.14%	531.34K	19/03

The second URL table is URL https://www.investing.com/indices/nasdaq-composite-components/2.



Name	Last	High	Low	Chg.	Chg. %	Vol.	Time
Change Healthcare	22.75	22.91	22.63	+0.01	+0.04%	2.39M	19/03
Charles&Colvard	3.110	3.180	2.715	+0.500	+19.16%	1.30M	19/03
Charter Communicat...	629.74	636.86	626.84	+0.35	+0.06%	2.78M	19/03
Check Cap Ltd	2.150	2.295	2.020	+0.110	+5.39%	7.54M	19/03
Check Point Software	116.68	118.30	116.53	+0.59	+0.51%	1.36M	19/03
Checkmate	13.90	14.98	13.52	-0.99	-6.65%	622.60K	19/03
Checkpoint Therape...	3.500	3.650	3.250	+0.180	+5.42%	1.09M	19/03
Chembio	4.18	4.41	4.13	-0.03	-0.71%	545.95K	19/03
ChemoCentryx	58.12	58.69	53.90	+4.60	+8.59%	1.28M	19/03
Chemung	41.05	41.52	38.20	+0.02	+0.05%	26.79K	19/03
Chesapeake Energy	43.31	43.76	42.80	-0.19	-0.44%	537.11K	19/03

The last URL table is https://money.cnn.com/quote/quote.html?symb=spx&page=34.



NASDAQ Composite Components

Price | Performance | Technical | Fundamental

Name	Last	High	Low	Chg.	Chg. %	Vol.	Time
Xiaobai Maimai	2.0100	2.0736	2.0000	-0.0700	-3.37%	71.68K	19/03
Xilinx	122.88	123.85	120.72	+0.63	+0.52%	6.38M	19/03
XOMA	34.79	40.24	34.65	-2.45	-6.58%	140.94K	19/03
Xp	39.64	39.88	37.65	+1.65	+4.34%	2.41M	19/03
Xpel	56.07	56.62	54.11	+0.12	+0.21%	334.98K	19/03
Xperi	24.46	24.84	23.82	+0.42	+1.75%	2.22M	19/03
XpresSpa	2.0000	2.1200	1.9501	0.0000	0.00%	6.35M	19/03
XTL Biopharma	3.390	3.470	3.270	+0.070	+2.11%	41.48K	19/03
Xunlei Ltd Adr	9.25	9.37	8.62	+0.42	+4.76%	3.14M	19/03
Y mAbs Therapeutics	35.00	36.36	33.50	-1.05	-2.91%	1.58M	19/03
Yandex	65.55	66.57	64.85	+0.32	+0.49%	1.71M	19/03
Yatra Online	2.48	2.53	2.33	+0.05	+2.06%	1.62M	19/03

19.10.1 Using Power Query to Retrieve the NASDAQ Composite Components

Notice the pattern among the URL shown. The difference among the URLs is the page number shown.

The following is Excel's power query M code to retrieve the first page.

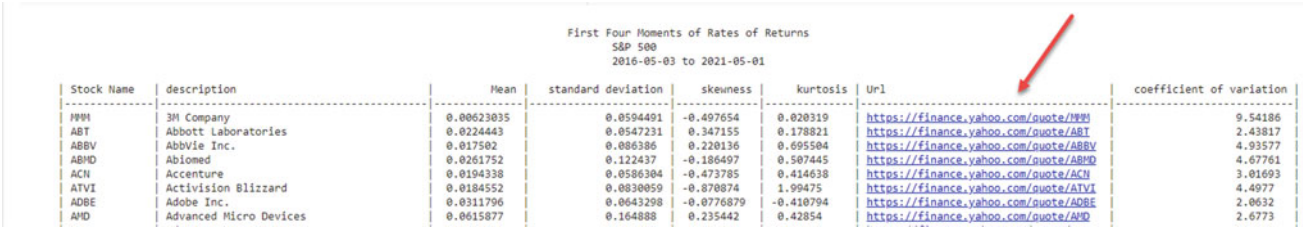
19.11 Using Python to Calculate the Four Statistical Moments of the Rate of Returns of Every Component in the S&P 500

The Python code below calculates the four statistical moments of the rate of return of every component of the S&P 500. The source for the S&P 500 components is

https://en.wikipedia.org/wiki/List_of_S%26P_500_companies.

The data source for every component is Yahoo! Finance. The Python code uses the Python library *yfinance* to the pricing data from Yahoo! Finance.

The Python code shows the Yahoo! Finance URL for every component of the S&P 500. For example, The URL for 3M is https://finance.yahoo.com/quote/MMM and the URL for Advanced Micro Devices is https://finance.yahoo.com/quote/AMD.



First Four Moments of Rates of Returns
S&P 500
2016-05-03 to 2021-05-01

Stock Name	description	Mean	standard deviation	skewness	kurtosis	Url	coefficient of variation
MMM	3M Company	0.00623035	0.0594491	-0.497654	0.020319	https://finance.yahoo.com/quote/MMM	9.54186
ABT	Abbott Laboratories	0.0224443	0.0547231	0.347155	0.178821	https://finance.yahoo.com/quote/ABT	2.43817
ABBV	AbbVie Inc.	0.017502	0.086386	0.220136	0.695504	https://finance.yahoo.com/quote/ABBV	4.93577
ABMD	Abiomed	0.0261752	0.122437	-0.186497	0.507445	https://finance.yahoo.com/quote/ABMD	4.67761
ACN	Accenture	0.0194338	0.0586304	-0.473785	0.414638	https://finance.yahoo.com/quote/ACN	3.01693
ATVI	Activision Blizzard	0.0184552	0.0830059	-0.870874	1.99475	https://finance.yahoo.com/quote/ATVI	4.4977
ADBE	Adobe Inc.	0.0311796	0.0643298	-0.0776879	-0.410794	https://finance.yahoo.com/quote/ADBE	2.0632
AMD	Advanced Micro Devices	0.0615877	0.164888	0.235442	0.42854	https://finance.yahoo.com/quote/AMD	2.6773

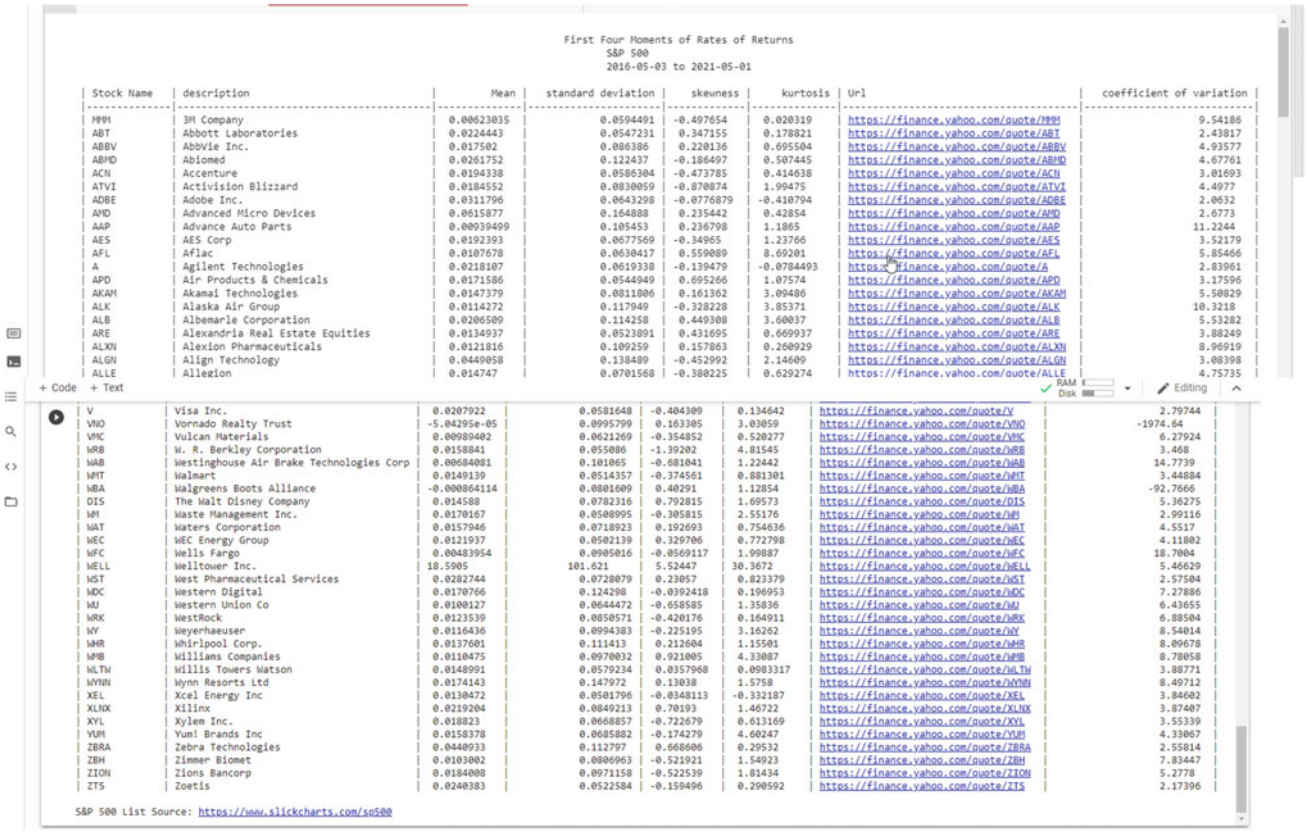
Below is the complete Python code and the output from the Python code.

```
"""get all data"""
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

Data = []
Url = []

for i in range(1,len(result)):

  try:
    Data.append(yf.download(result[i][0].replace(".","-"), start = Start, interval = T_interval)['Adj Close'].dropna())
    Url.append('https://finance.yahoo.com/quote/'+result[i][0])
  except Exception:
    print("Problem retrieving info for ticker: " + result[i][0])
    continue

  if i%20 == 0:
    r = np.random.randint(3,8)
    time.sleep(r)
    print(str(i) + ' of 500   Getting S&P 500 data ...')

Data = pd.concat(Data,axis=1)
```

```
[*********************100%***********************]  1 of 1 completed
480 of 500   Getting S&P 500 data ...
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
500 of 500   Getting S&P 500 data ...
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
```

```
[8]  #calculate the 4 moments of statistics of the rates of returns of every ticker of the S&P 500
     #https://medium.com/analytics-vidhya/statistics-moments-of-a-distribution-1bcfc4cbbd48
     Ret = Data.pct_change()
     summury = pd.concat([Ret.mean(),Ret.std(),Ret.skew(),Ret.kurtosis(),np.divide(Ret.std(),Ret.mean())],axis=1)
     summury.columns = ['Mean','standard deviation','skewness','kurtosis','coefficient of variation']
     Ticker  = [i[0] for i in result]
     Ticker = Ticker[1:]
     description = [i[1] for i in result]
     description = description[1:]
     summury.insert(0, "description", description, True)
     summury.insert(0, "Stock Name", Ticker, True)
     summury.insert(6, "Url", Url, True)
```

```
[9]  #There are more then 500 tickers in the S&P500 Index
     print("")
     print("                                                                    First Four Moments of Rates of Returns")
     print("                                                                           S&P 500 ")
     print ("                                                                          " + Start + ' to ' + Default_end           )
     print("")
     print(tabulate(summury,headers=summury.columns,tablefmt="github",showindex=False))
     print("")
     print("S&P 500 List Source: https://www.slickcharts.com/sp500")
```



First Four Moments of Rates of Returns
S&P 500
2016-05-03 to 2021-05-01

Stock Name	description	Mean	standard deviation	skewness	kurtosis	Url	coefficient of variation
MMM	3M Company	0.00623035	0.0594491	-0.497654	0.020319	https://finance.yahoo.com/quote/MMM	9.54186
ABT	Abbott Laboratories	0.0224443	0.0547231	0.347155	0.178821	https://finance.yahoo.com/quote/ABT	2.43817
ABBV	AbbVie Inc.	0.017502	0.086386	0.220136	0.695504	https://finance.yahoo.com/quote/ABBV	4.93577
ABMD	Abiomed	0.0261752	0.122437	-0.186497	0.507445	https://finance.yahoo.com/quote/ABMD	4.67761
ACN	Accenture	0.0194338	0.0586304	-0.473785	0.414638	https://finance.yahoo.com/quote/ACN	3.01693
ATVI	Activision Blizzard	0.0184552	0.0830059	-0.870874	1.99475	https://finance.yahoo.com/quote/ATVI	4.4977
ADBE	Adobe Inc.	0.0311796	0.0643298	-0.0776879	-0.410794	https://finance.yahoo.com/quote/ADBE	2.0632
AMD	Advanced Micro Devices	0.0615877	0.164888	0.235442	0.42854	https://finance.yahoo.com/quote/AMD	2.6773
AAP	Advance Auto Parts	0.00939499	0.105453	0.236798	1.1865	https://finance.yahoo.com/quote/AAP	11.2244
AES	AES Corp	0.0192393	0.0677569	-0.34965	1.23766	https://finance.yahoo.com/quote/AES	3.52179
AFL	Aflac	0.0107678	0.0630417	0.559089	8.69201	https://finance.yahoo.com/quote/AFL	5.85466
A	Agilent Technologies	0.0218107	0.0619338	-0.139479	-0.0784493	https://finance.yahoo.com/quote/A	2.83961
APD	Air Products & Chemicals	0.0171586	0.0544949	0.695266	1.07574	https://finance.yahoo.com/quote/APD	3.17596
AKAM	Akamai Technologies	0.0147379	0.0811806	0.161362	3.09486	https://finance.yahoo.com/quote/AKAM	5.50829
ALK	Alaska Air Group	0.0114272	0.117949	-0.328228	3.85371	https://finance.yahoo.com/quote/ALK	10.3218
ALB	Albemarle Corporation	0.0206509	0.114258	0.449308	3.60037	https://finance.yahoo.com/quote/ALB	5.53282
ARE	Alexandria Real Estate Equities	0.0134937	0.0523891	0.431695	0.669937	https://finance.yahoo.com/quote/ARE	3.88249
ALXN	Alexion Pharmaceuticals	0.0121816	0.109259	0.157863	0.260929	https://finance.yahoo.com/quote/ALXN	8.96919
ALGN	Align Technology	0.0449058	0.138489	-0.452992	2.14609	https://finance.yahoo.com/quote/ALGN	3.08398
ALLE	Allegion	0.014747	0.0701568	-0.380225	0.629274	https://finance.yahoo.com/quote/ALLE	4.75735
V	Visa Inc.	0.0207922	0.0581648	-0.404309	0.134642	https://finance.yahoo.com/quote/V	2.79744
VNO	Vornado Realty Trust	-5.042956-05	0.0995799	0.163305	3.03059	https://finance.yahoo.com/quote/VNO	-1974.64
VMC	Vulcan Materials	0.00989402	0.0621269	-0.354852	0.520277	https://finance.yahoo.com/quote/VMC	6.27924
WRB	W. R. Berkley Corporation	0.0158841	0.055086	-1.39202	4.81545	https://finance.yahoo.com/quote/WRB	3.468
WAB	Westinghouse Air Brake Technologies Corp	0.00684081	0.101065	-0.681041	1.22442	https://finance.yahoo.com/quote/WAB	14.7739
WMT	Walmart	0.0149139	0.0514357	-0.374561	0.881301	https://finance.yahoo.com/quote/WMT	3.44884
WBA	Walgreens Boots Alliance	-0.000864114	0.0801609	0.40291	1.12854	https://finance.yahoo.com/quote/WBA	-92.7666
DIS	The Walt Disney Company	0.014588	0.0782316	0.792815	1.69573	https://finance.yahoo.com/quote/DIS	5.36275
WM	Waste Management Inc.	0.0170167	0.0508995	-0.305815	2.55176	https://finance.yahoo.com/quote/WM	2.99116
WAT	Waters Corporation	0.0157946	0.0718923	0.192693	0.754636	https://finance.yahoo.com/quote/WAT	4.5517
WEC	WEC Energy Group	0.0121937	0.0502139	0.329706	0.772798	https://finance.yahoo.com/quote/WEC	4.11802
WFC	Wells Fargo	0.00483954	0.0905016	-0.0569117	1.99887	https://finance.yahoo.com/quote/WFC	18.7004
WELL	Welltower Inc.	18.5905	101.621	5.52447	30.3672	https://finance.yahoo.com/quote/WELL	5.46629
WST	West Pharmaceutical Services	0.0282744	0.0728079	0.23057	0.823379	https://finance.yahoo.com/quote/WST	2.57504
WDC	Western Digital	0.0170766	0.124298	-0.0392418	0.196953	https://finance.yahoo.com/quote/WDC	7.27886
WU	Western Union Co	0.0100127	0.0644472	-0.658585	1.35836	https://finance.yahoo.com/quote/WU	6.43655
WRK	WestRock	0.0123539	0.0850571	-0.420176	0.164911	https://finance.yahoo.com/quote/WRK	6.88504
WY	Weyerhaeuser	0.0116436	0.0994383	-0.225195	3.16262	https://finance.yahoo.com/quote/WY	8.54014
WHR	Whirlpool Corp.	0.0137601	0.111413	0.212604	1.15501	https://finance.yahoo.com/quote/WHR	8.09678
WMB	Williams Companies	0.0110475	0.0970032	0.921005	4.33087	https://finance.yahoo.com/quote/WMB	8.78058
WLTW	Willis Towers Watson	0.0148991	0.0579234	0.0357968	0.0983317	https://finance.yahoo.com/quote/WLTW	3.88771
WYNN	Wynn Resorts Ltd	0.0174143	0.147972	0.13038	1.5758	https://finance.yahoo.com/quote/WYNN	8.49712
XEL	Xcel Energy Inc	0.0130472	0.0501796	-0.0348113	-0.332187	https://finance.yahoo.com/quote/XEL	3.84602
XLNX	Xilinx	0.0219204	0.0849213	0.70193	1.46722	https://finance.yahoo.com/quote/XLNX	3.87407
XYL	Xylem Inc.	0.018823	0.0668857	-0.722679	0.613169	https://finance.yahoo.com/quote/XYL	3.55339
YUM	Yum! Brands Inc	0.0158378	0.0685882	-0.174279	4.60247	https://finance.yahoo.com/quote/YUM	4.33067
ZBRA	Zebra Technologies	0.0440933	0.112797	0.668606	0.29532	https://finance.yahoo.com/quote/ZBRA	2.55814
ZBH	Zimmer Biomet	0.0103002	0.0806963	-0.521921	1.54923	https://finance.yahoo.com/quote/ZBH	7.83447
ZION	Zions Bancorp	0.0184008	0.0971158	-0.522539	1.81434	https://finance.yahoo.com/quote/ZION	5.2778
ZTS	Zoetis	0.0240383	0.0522584	-0.159496	0.290592	https://finance.yahoo.com/quote/ZTS	2.17396

S&P 500 List Source: https://www.slickcharts.com/sp500

```
[ ]  1 #download four moment calculations to a CSV file
     2 summury.to_csv('chapter19_SP500FourMoments.csv')
     3 print("Saving SP500 report to 'Download' folder")
     4 files.download('chapter19_SP500FourMoments.csv')
     5 print("Finish saving SP500 report to 'Download' folder")
```

```
Saving SP500 report to 'Download' folder
Finish saving SP500 report to 'Download' folder
```

19.12 Summary

In this chapter, we discussed several price and quantity indexes that are used to analyze economic activities. We also looked at three popular stock market indexes. We saw how to find the components for the Dow Jones Industrial Average, S&P 500, and the NASDAQ composite.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edition, AbeBooks Seller, 2020.

Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.

Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.

Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O’Reilly Media, 2017.

Check for updates

Sampling Surveys: Methods and Applications 20

20.1 Introduction

In previous chapters, we investigated sampling but only in terms of simple random sampling, in which each potential sample of n members has an equal chance of being chosen. Most of the time, this requirement is satisfied due to the fact that the sample size is small compared to the population. But when the sample size becomes a large part of the population, some adjustments must be made.

This chapter can be broken down into the following sections. In Sect. 20.2, we discuss random number tables, in Sect. 20.3, we discuss the confidence interval for the population mean, and in Sect. 20.4, we discuss the confidence interval for the population proportion. In Sect. 20.5, we determine the sample size, and finally, in Sect. 20.6, we summarize the chapter.

20.2 Random Number Tables

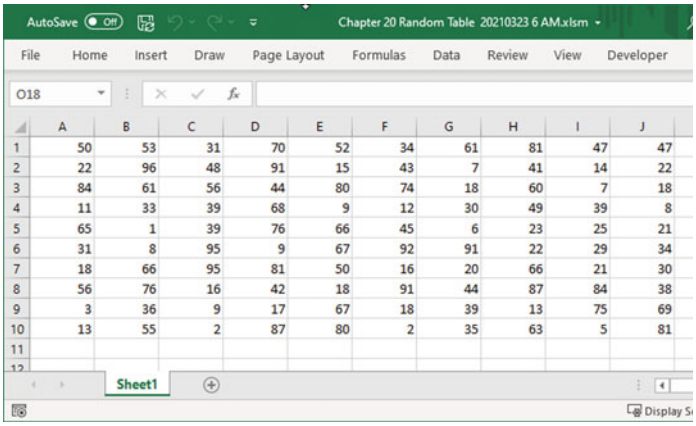
Random tables are very often used to obtain random samples. The distribution that we will use here to generate random tables is the uniform distribution because, in many applications, we are interested in every number having an equal chance of occurring. The uniform distribution has this property.

20.2.1 Excel VBA

Suppose we are interested in creating a 10-by-10 table that has numbers randomly generated from 0 to 100. Below is the Excel VBA code to create a 10 by 10 table with numbers randomly generated from 0 to 100.

```
'/*******************************************************
'/Purpose: To generate 100 numbers from 0 to 100 on the
'/          active sheet
'/*******************************************************
Sub Random100()
    Dim rRandom As Range
    Dim rCell As Range
    Set rRandom = ActiveSheet.Range(``a1:j10'')
    For Each rCell In rRandom.Cells
        Randomize  ' Initialize random-number generator.
        rCell.Value = Rnd * 100
    Next

    rRandom.NumberFormat = ``0''
End Sub
```

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_20



	A	B	C	D	E	F	G	H	I	J
1	50	53	31	70	52	34	61	81	47	47
2	22	96	48	91	15	43	7	41	14	22
3	84	61	56	44	80	74	18	60	7	18
4	11	33	39	68	9	12	30	49	39	8
5	65	1	39	76	66	45	6	23	25	21
6	31	8	95	9	67	92	91	22	29	34
7	18	66	95	81	50	16	20	66	21	30
8	56	76	16	42	18	91	44	87	84	38
9	3	36	9	17	67	18	39	13	75	69
10	13	55	2	87	80	2	35	63	5	81
11										

20.2.2 Python Code

Below shows the Python code to generate a 10-by-10 table that has numbers randomly generated from 0 to 100.

20.3 Confidence Interval for the Population Mean

In previous chapters, we assumed either that the population size was sufficiently large or the sample was small enough compared to the population for the sample to be random. When this is not the case, we have to make adjustments. One such adjustment is the variance of the sample mean. The adjusted variance of the sample mean is defined as

$$\widehat{\sigma_{\bar{x}}^2} = \frac{S^2}{n} * \frac{N-n}{N-1}$$

As a rule of thumb, if $n/N > 0.05$, we should use the above-adjusted sample variance. If $n/N < 0.05$, we should use the calculated sample variance, s^2.

20.3.1 Example 20.1

Suppose an investment adviser is trying to decide whether a small retirement community consisting of 1,000 residents represents a promising source of potential clients. To determine the potential business, the investment adviser decides to analyze the size of the residents' investment portfolios. A random sample of 75 residents, who were able to respond anonymously, produces a sample mean of $375,000 with a sample standard deviation of $120,000. Calculate the 95% confidence interval for the mean value of the investment portfolio.

The ratio of the sample size to the population size is $n/N = 75/1000 = 0.08$. Since the ratio of the sample size to the population size is greater than 0.05, we should use the adjusted sample variance.

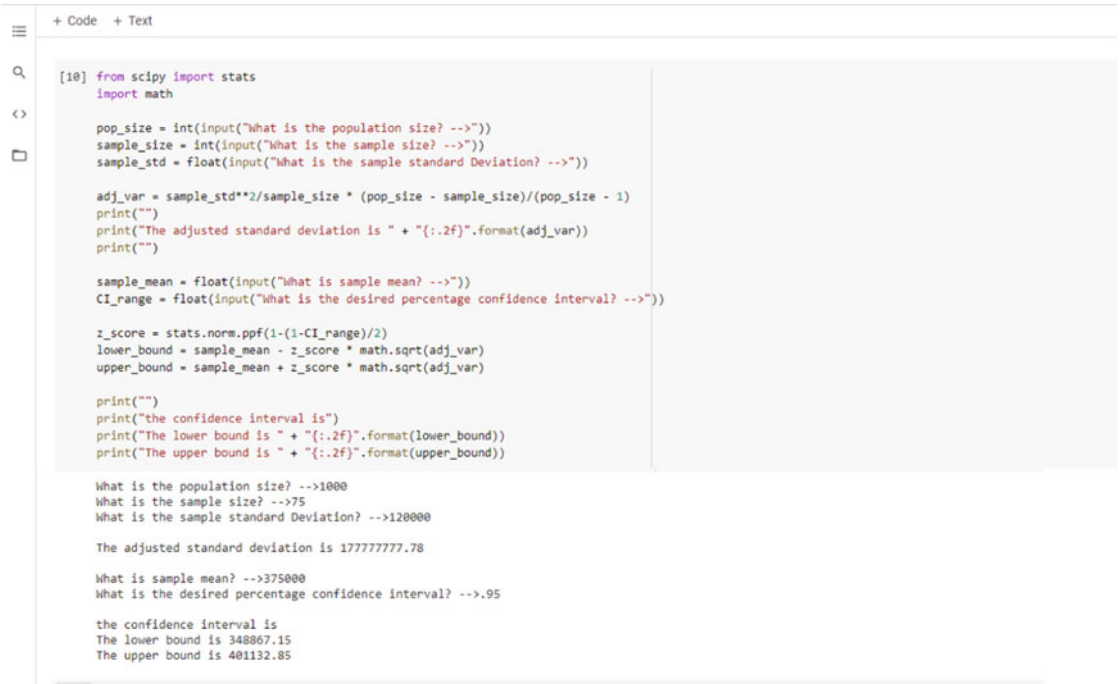
When we do not have to make an adjustment to the variance of the sample mean, the formula for the confidence interval would be

$$\overline{x} - z_{\alpha/2}(\sigma/\sqrt{n}) < \mu < \overline{x} + z_{\alpha/2}(\sigma/\sqrt{n})$$

With the adjustment to the variance of the sample mean, the confidence interval formula is

$$\overline{x} - z_{\alpha/2}\widehat{\sigma_{\overline{x}}} < \mu < \overline{x} + z_{\alpha/2}\widehat{\sigma_{\overline{x}}}$$

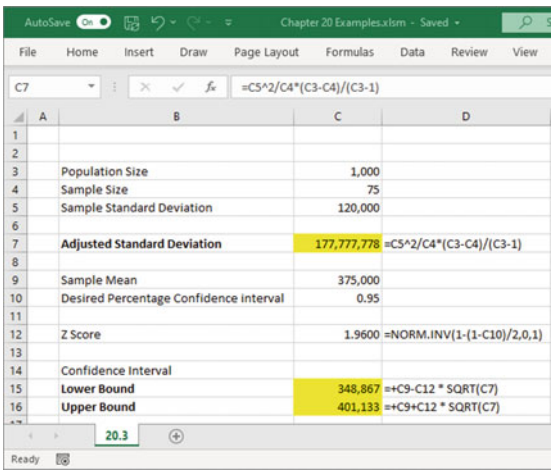
20.3.1.1 Python Code



```
+ Code  + Text

[10] from scipy import stats
     import math

     pop_size = int(input("What is the population size? -->"))
     sample_size = int(input("What is the sample size? -->"))
     sample_std = float(input("What is the sample standard Deviation? -->"))

     adj_var = sample_std**2/sample_size * (pop_size - sample_size)/(pop_size - 1)
     print("")
     print("The adjusted standard deviation is " + "{:.2f}".format(adj_var))
     print("")

     sample_mean = float(input("What is sample mean? -->"))
     CI_range = float(input("What is the desired percentage confidence interval? -->"))

     z_score = stats.norm.ppf(1-(1-CI_range)/2)
     lower_bound = sample_mean - z_score * math.sqrt(adj_var)
     upper_bound = sample_mean + z_score * math.sqrt(adj_var)

     print("")
     print("the confidence interval is")
     print("The lower bound is " + "{:.2f}".format(lower_bound))
     print("The upper bound is " + "{:.2f}".format(upper_bound))

     What is the population size? -->1000
     What is the sample size? -->75
     What is the sample standard Deviation? -->120000

     The adjusted standard deviation is 177777777.78

     What is sample mean? -->375000
     What is the desired percentage confidence interval? -->.95

     the confidence interval is
     The lower bound is 348867.15
     The upper bound is 401132.85
```

From the above Python code, we can see that the 95% confidence interval is from $348,867 to $401,132.

Below shows the calculations done in Excel.



AutoSave On — Chapter 20 Examples.xlsm - Saved

File Home Insert Draw Page Layout Formulas Data Review View

C7 =C5^2/C4*(C3-C4)/(C3-1)

	A	B	C	D
1				
2				
3		Population Size	1,000	
4		Sample Size	75	
5		Sample Standard Deviation	120,000	
6				
7		**Adjusted Standard Deviation**	177,777,778	=C5^2/C4*(C3-C4)/(C3-1)
8				
9		Sample Mean	375,000	
10		Desired Percentage Confidence interval	0.95	
11				
12		Z Score	1.9600	=NORM.INV(1-(1-C10)/2,0,1)
13				
14		Confidence Interval		
15		**Lower Bound**	348,867	=+C9-C12 * SQRT(C7)
16		**Upper Bound**	401,133	=+C9+C12 * SQRT(C7)

20.3

Ready

20.4 Confidence Interval for the Population Proportion

We will also need to adjust the variance of the sample proportion when the sample relative to the population is not so small. The adjusted variance for a sample proportion is

$$\widehat{\sigma_{\bar{x}}^2} = \frac{\hat{p}(1-\hat{p})}{n} * \frac{N-n}{N-1}$$

20.4.1 Example 20.2

Suppose we want to determine the proportion of college-bound high school seniors in a class of 500. A survey of 30 randomly selected students reveals that 19 will be attending college. Create a 90% confidence interval for the population proportion p.

The ratio of the sample size to the population size is 30/500 = 0.06. Since this is greater than 0.05, we will have to make an adjustment to the sample variance.

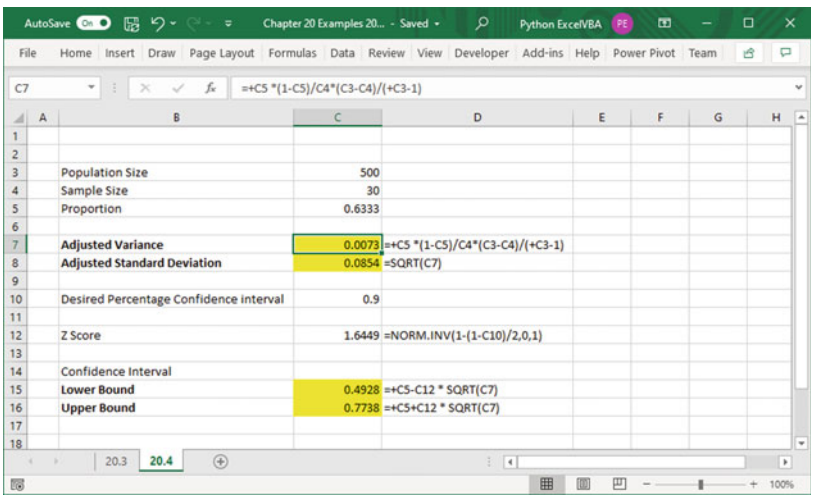
20.4.1.1 Python Code

Below is a Python program to calculate both the adjusted variance and the confidence interval for a population proportion.

```
+ Code  + Text

[7] import math

    pop_size = int(input("What is the population size? -->"))
    sample_size = int(input("What is the sample size? -->"))
    prop = float(input("What is the propotion? -->"))
    adjvar = prop * (1-prop) / sample_size * (pop_size - sample_size) / (pop_size - 1)

    print("")
    print("The adjusted var is " + "{:.4f}".format(adjvar))
    print("The adjusted std is " + "{:.4f}".format(math.sqrt(adjvar)))

    print("")
    print("The upper bound is " + "{:.4f}".format(-1.64 * math.sqrt(adjvar) + prop))
    print("The lower bound is " + "{:.4f}".format(1.64 * math.sqrt(adjvar) + prop))

What is the population size? -->500
What is the sample size? -->30
What is the propotion? -->.6333

The adjusted var is 0.0073
The adjusted std is 0.0854

The upper bound is 0.4933
The lower bound is 0.7733
```

From the Python program, we see that the 90% confidence interval is from 49.33% to 77.33%.

Below shows the calculations done in Excel.



20.5 Determining Sample Size

Up to now, we have obtained the sample size first and then found the standard deviation. What if we did the reverse and knew a specific standard deviation but did not know the sample size? The formula to find the required sample size for a required standard deviation would be

$$n = \frac{N\sigma^2}{(N-1)\widehat{\sigma_{\bar{x}}^2} + \sigma^2}$$

20.5.1 Example 20.3

Crow Company's accountant decides that the best way to determine the company's mean accounts receivable is to take a simple random sample of the 1,025 accounts. Assume that the population variance, SYMBOL 115 \f "Symbol"^{2}, is $2,425. What sample size should the accountant take if she would like to have a level of precision of $75?

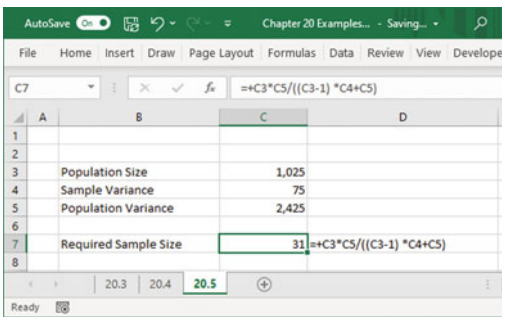
20.5.1.1 Python Code

Below is the Python code to calculate the sample size.

```
+ Code  + Text

N = int(input("What is the population size? --> "))
s_var = float(input("What is the sample variance? --> "))
p_var = float(input("What is the population variance? --> "))
n = N * p_var /((N-1) * s_var + p_var)
print("The required sample size is "+ str(int(n + 1)))

What is the population size? --> 1025
What is the sample variance? --> 75
What is the population variance? --> 2425
The required sample size is 32
```

From the Python output, we can see that a sample of 32 accounts will produce the desired result.

Below shows the calculations done in Excel.



20.6 Summary

In this chapter, we first discussed how random number tables could be generated. Then we discussed samples that have a ratio of a sample size to population size greater than 0.05. This is important because in such a case, we have to make some adjustments to make the estimate unbiased.

We also learned how to determine the required sample size for a desired sample variance.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.
Kusleika, Dick and Alexander, Michael. “Excel 2019 Power Programming With Vba.”, Wiley, 2019.
Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013.
Telles, Matt. “Python for Professionals: Learning Python as a Second Language.”, BPB Publications, 2019.
Wickham, Hadley and Grolemund, Garrett. “R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.”, O'Reilly Media, 2017.

Statistical Decision Theory

21

21.1 Introduction

This chapter was originally in Lee et al. (2013). It discusses the statistical decision theory. The main topics covered in this chapter are (i) Four Key Elements of a Decision, (ii) Decisions Based on Extreme Values, (iii) Expected Monetary Value and Utility Analysis, (iv) Bayes Strategies, (v) Decision Trees and Expected Monetary Values, (vi) Mean and Variance Trade-off Analysis (Optional), (vii) The Mean and Variance Method for Capital Budgeting Decisions.

The excel applications will be emphasized in this chapter. Section 21.2 discusses the excel applications in the decision tree approach to calculate the expected monetary values. Section 21.3 presents excel applications for NPV and IRR methods for capital budgeting decision under certainty. Section 21.4 discusses excel application of z-statistic method for capital budgeting decision under uncertainty. Section 21.5 summarizes the chapter.

21.2 Decision Trees and Expected Monetary Values

Following the example from Sect. 21.6 of Lee et al. (2013), we assume that an oil company is trying to decide whether to test for the presence of oil or to drill for oil. If oil is struck, the revenues are $1 million, with a cost of $100,000 to drill. The firm has to decide whether to test first for the presence of oil. Without testing, the probability of striking oil is 0.1. Thus the firm's expected value without testing is 0 ($1 million times 0.1, less drilling fees of $100,000). The cost of the oil test is $50,000. If the test is positive, there is a 0.6 probability that oil will be struck; if the test is negative, the probability of striking oil is 0.05.

The expected gain for a negative test result is 0.05(1,000,000)–100,000–50,000 = −$100,000. If the test is positive, the expected profit is (1,000,000)(0.6)–100,000–50,000 = $450,000. If the firm's test is negative, the firm should not drill; however, if the test is positive, it should drill. This analysis makes it clear that the test should be done. The above-mentioned information can be summarized as in Table 21.1. The excel functions of decision trees in terms of Table 21.1 are presented in Fig. 21.1. The results of decision trees in terms of Fig. 21.1 are presented in Fig. 21.2.

21.3 NPV and IRR Method for Capital Budgeting Decision Under Certainty

Both Net present value (NPV) method and internal rate of return (IRR) method can be used to do the capital budgeting decision. For example, project A and project B, the initial outlays and net cash inflow for year 0 to year 4 are presented in Table 21.2. In Table 21.2, we know that the initial outlay at year 0 for Project A and B is $80,000 and $50,000, respectively. In year 1, additional investment for project A and B is $20,000 and $50,000, respectively. The net cash inflow of project A for the next 4 years is $20,000, $30,000, $50,000, and $50,000, respectively. The net cash inflow of project B for the next 4 years is $40,000, $60,000, $30,000, and $10,000, respectively.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_21

Table 21.1 The inputs of the example above

Numbers for the oil drilling Problems	
Test cost	50,000
Drilling cost	100,000
Payoff for successful drilling	1,000,000
Probability for test results	
Positive	0.5
Negative	0.5
Probability for drilling results	
Test positive	
Successful	0.6
Fail	0.4
Test negative	
Successful	0.05
Fail	0.05
No test	
Successful	0.1
Fail	0.0

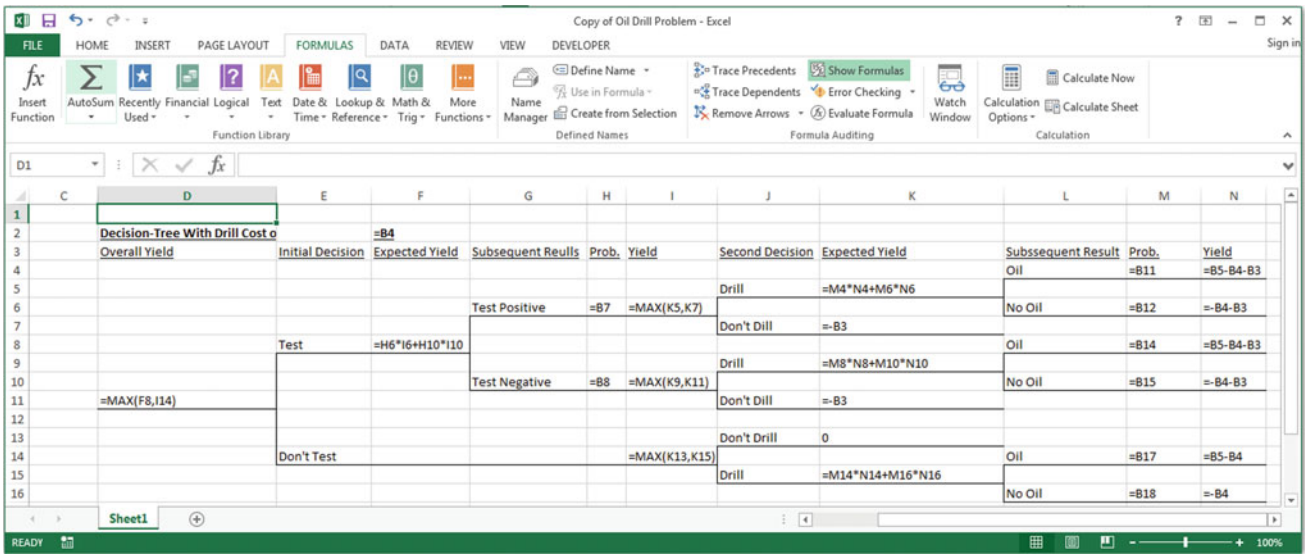


Fig. 21.1 The excel functions of decision trees in terms of Table 21.1

The net present value of a project is computed by discounting the project's cash flows to the present by the appropriate cost of capital. The formula used to calculate NPV can be defined as follow:

$$NPV = \sum_{t=1}^{N} \frac{CF_t}{(1+k)^t} - I \tag{21.1}$$

where

k = the appropriate discount rate.
CFt = Net Cash flow (positive or negative) in period t,
I = Initial outlay,
N = Life of the project.

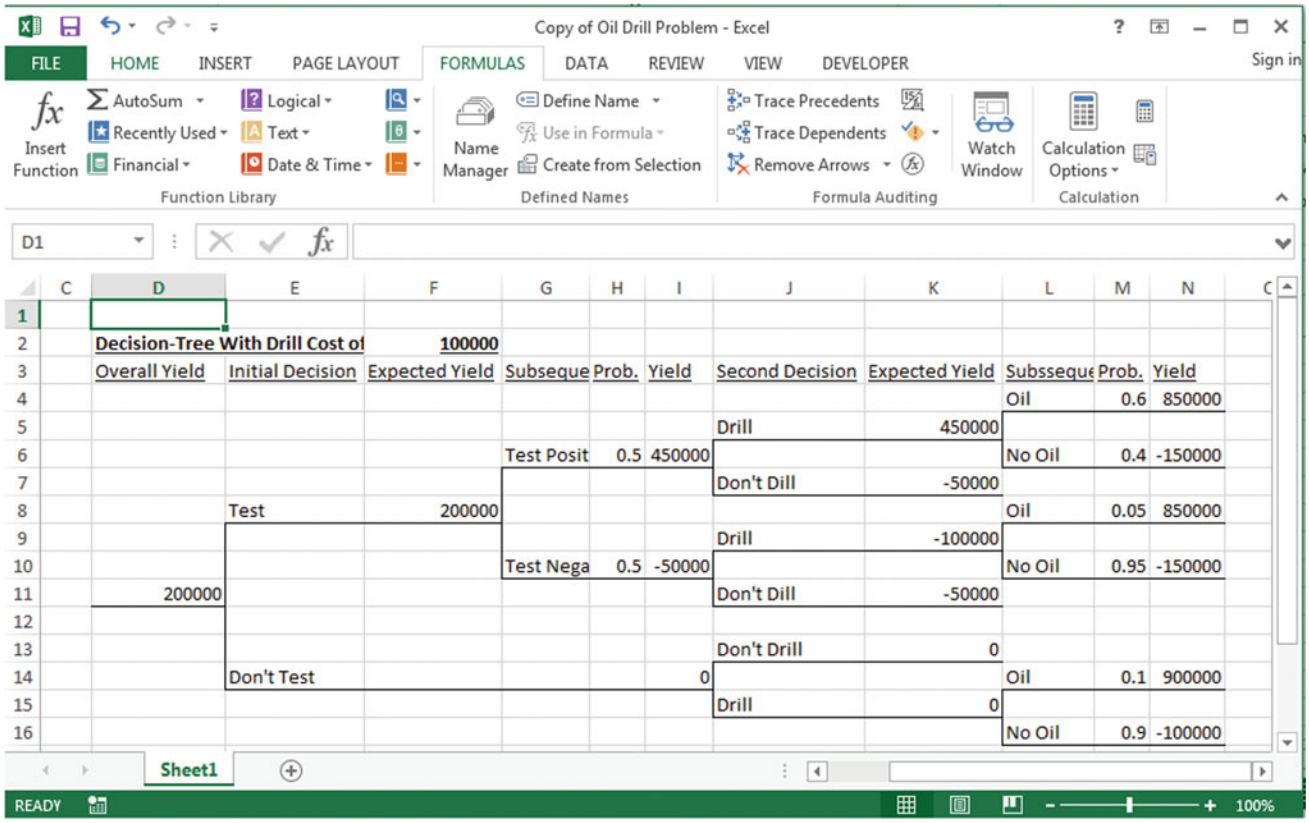


Fig. 21.2 The results of decision trees in terms of Fig. 21.1

Table 21.2 Partial inputs information for NPV method and IRR method

		Year 0	Year 1	Year 2	Year 3	Year 4
Project A	Cost	−$80,000	−$20,000	0	0	0
	Return	0	$20,000	$30,000	$50,000	$50,000
Project B	Cost	−$50,000	−$50,000	0	0	0
	Return	0	$40,000	$60,000	$30,000	$10,000

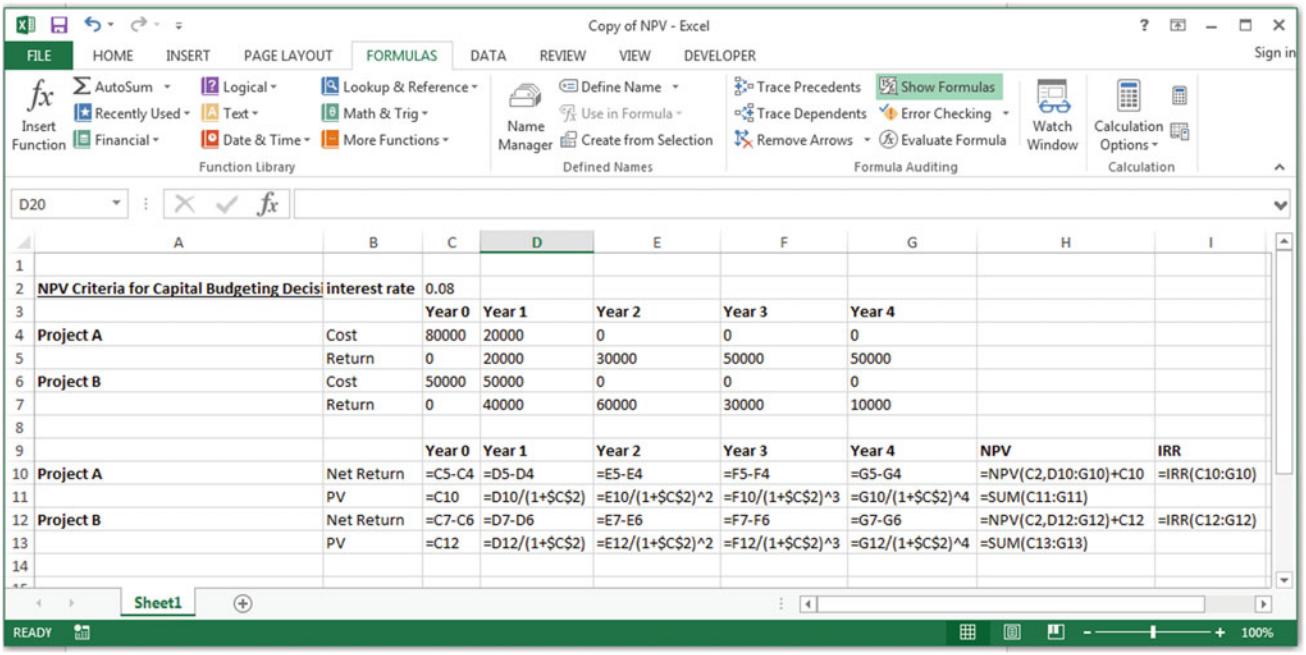
Using the excel function, we can calculate NPV for both projects A and B. We also can calculate Example above using Excel NPV function. NPV is a function to calculate the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values). The NPV function in Cell H10 is equal to

$$= \mathrm{NPV}(\mathrm{C2}, \mathrm{D10{:}G10}) + \mathrm{C10}$$

Based upon the NPV function in Fig. 21.3, the NPV results are shown in Fig. 21.4.

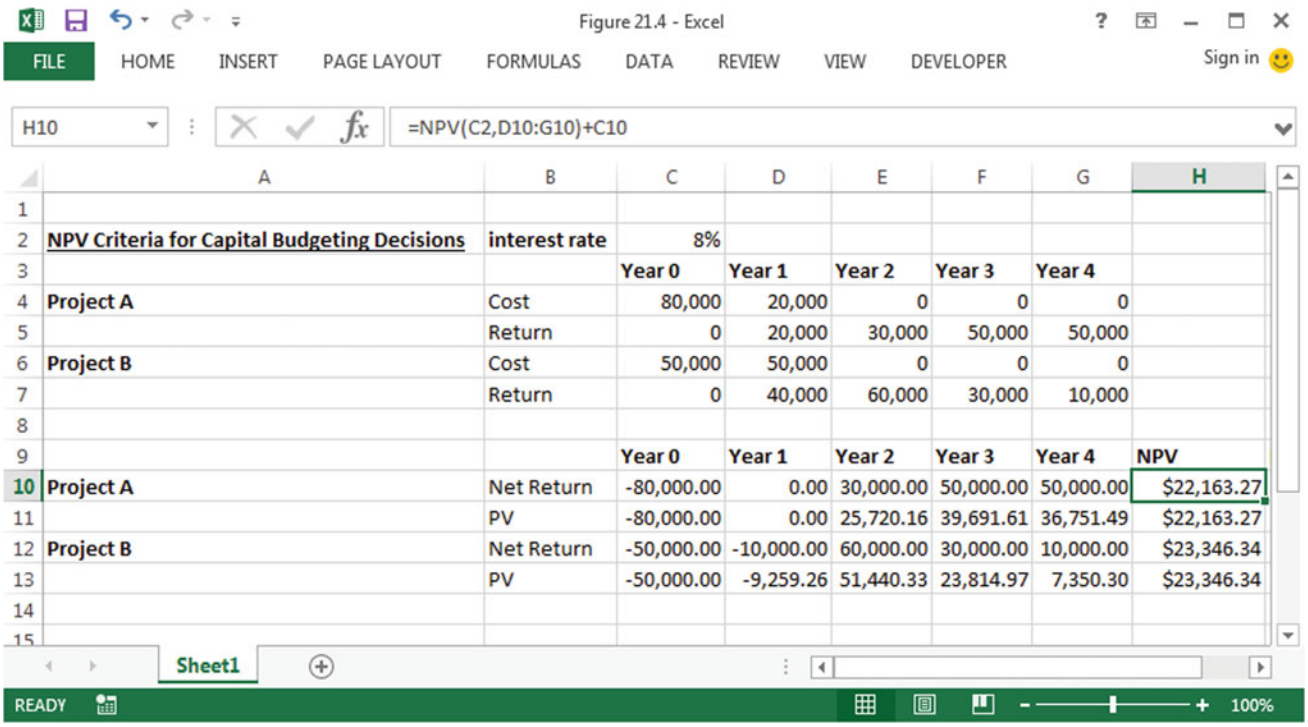
The internal rate of return (IRR, r) is the discount rate that equates the discounted cash flows from a project to its investment. Thus, one must solve iteratively for the r in Eq. (21.2):

$$\sum_{t=1}^{N} \frac{CF_t}{(1+r)^t} = I \tag{21.2}$$



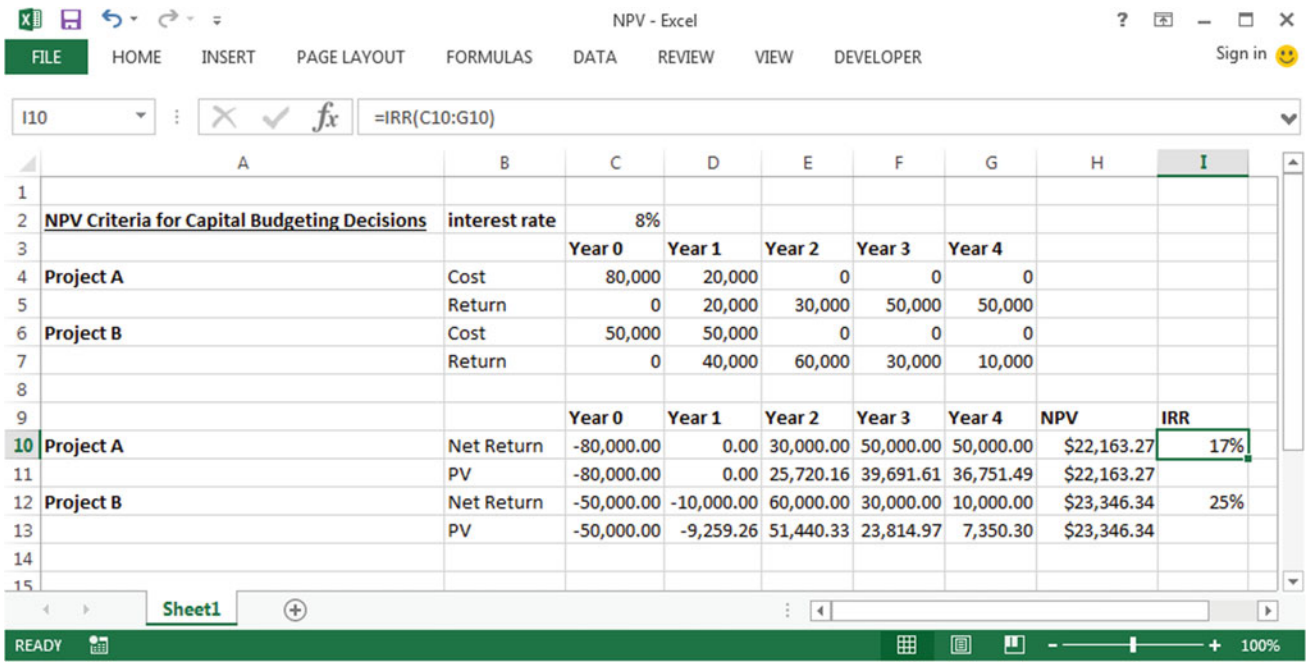
	A	B	C	D	E	F	G	H	I
1									
2	NPV Criteria for Capital Budgeting Decisi	interest rate	0.08						
3			Year 0	Year 1	Year 2	Year 3	Year 4		
4	Project A	Cost	80000	20000	0	0	0		
5		Return	0	20000	30000	50000	50000		
6	Project B	Cost	50000	50000	0	0	0		
7		Return	0	40000	60000	30000	10000		
8									
9			Year 0	Year 1	Year 2	Year 3	Year 4	NPV	IRR
10	Project A	Net Return	=C5-C4	=D5-D4	=E5-E4	=F5-F4	=G5-G4	=NPV(C2,D10:G10)+C10	=IRR(C10:G10)
11		PV	=C10	=D10/(1+$C$2)	=E10/(1+$C$2)^2	=F10/(1+$C$2)^3	=G10/(1+$C$2)^4	=SUM(C11:G11)	
12	Project B	Net Return	=C7-C6	=D7-D6	=E7-E6	=F7-F6	=G7-G6	=NPV(C2,D12:G12)+C12	=IRR(C12:G12)
13		PV	=C12	=D12/(1+$C$2)	=E12/(1+$C$2)^2	=F12/(1+$C$2)^3	=G12/(1+$C$2)^4	=SUM(C13:G13)	
14									

Fig. 21.3 Excel calculation functions for NPV method



	A	B	C	D	E	F	G	H
1								
2	NPV Criteria for Capital Budgeting Decisions	interest rate	8%					
3			Year 0	Year 1	Year 2	Year 3	Year 4	
4	Project A	Cost	80,000	20,000	0	0	0	
5		Return	0	20,000	30,000	50,000	50,000	
6	Project B	Cost	50,000	50,000	0	0	0	
7		Return	0	40,000	60,000	30,000	10,000	
8								
9			Year 0	Year 1	Year 2	Year 3	Year 4	NPV
10	Project A	Net Return	-80,000.00	0.00	30,000.00	50,000.00	50,000.00	$22,163.27
11		PV	-80,000.00	0.00	25,720.16	39,691.61	36,751.49	$22,163.27
12	Project B	Net Return	-50,000.00	-10,000.00	60,000.00	30,000.00	10,000.00	$23,346.34
13		PV	-50,000.00	-9,259.26	51,440.33	23,814.97	7,350.30	$23,346.34
14								

Fig. 21.4 Excel calculation results for NPV method



	A	B	C	D	E	F	G	H	I
1									
2	NPV Criteria for Capital Budgeting Decisions	interest rate	8%						
3			Year 0	Year 1	Year 2	Year 3	Year 4		
4	Project A	Cost	80,000	20,000	0	0	0		
5		Return	0	20,000	30,000	50,000	50,000		
6	Project B	Cost	50,000	50,000	0	0	0		
7		Return	0	40,000	60,000	30,000	10,000		
8									
9			Year 0	Year 1	Year 2	Year 3	Year 4	NPV	IRR
10	Project A	Net Return	-80,000.00	0.00	30,000.00	50,000.00	50,000.00	$22,163.27	17%
11		PV	-80,000.00	0.00	25,720.16	39,691.61	36,751.49	$22,163.27	
12	Project B	Net Return	-50,000.00	-10,000.00	60,000.00	30,000.00	10,000.00	$23,346.34	25%
13		PV	-50,000.00	-9,259.26	51,440.33	23,814.97	7,350.30	$23,346.34	
14									
15									

Fig. 21.5 The excel calculation results for IRR

where

CFt = Net Cash flow (positive or negative) in period t,
I = Initial investment,
N = Life of the project.
r = the internal rate of return.

In addition, we can use Excel function IRR to calculate the internal rate of return. IRR is a function to calculate the internal rate of return which is the rate of return received for an investment consisting of payments (negative values) and income (positive values) that occur at regular periods. The IRR function in Cell I10 is

$$= \text{IRR}(\text{C10} : \text{G10})$$

Based upon the IRR function on Fig. 21.3, the IRR results in terms of excel calculations are shown in Fig. 21.5. Chapter 28 will discuss both NPV and IRR methods in capital budgeting in further detail.

21.4 The Statistical Distribution Method

21.4.1 Methodology

The riskiness of an investment proposal is generally defined as the variability of its possible returns. Decision situations can generally be classified into three types: certainty, risk, and uncertainty. The distinction between risk and uncertainty is that risk involves situations in which the probabilities of a particular event occurring are known, whereas with uncertainty these probabilities themselves are not known. Throughout this chapter, the terms "risk" and "uncertainty" will be used interchangeably, although the strict difference between them is not a matter to be taken lightly.

Economic factors peculiar to investment, competition, technological development, consumer preferences, and labor market conditions are a few of the factors that make it virtually impossible to foretell the future. Consequently, the revenues, costs, and economic life of investment projects are less than certain.

Table 21.3 Variance–covariance matrix

$W_1^2\sigma_1^2$	$W_1W_2\ \mathrm{Cov}(C_1, C_2)$	$W_1W_3\ \mathrm{Cov}(C_1, C_3)$
$W_2W_1\ \mathrm{Cov}(C_2, C_1)$	$W_2^2\sigma_2^2$	$W_1W_3\ \mathrm{Cov}(C_2, C_3)$
$W_3W_1\ \mathrm{Cov}(C_3, C_1)$	$W_2W_3\ \mathrm{Cov}(C_2, C_3)$	$W_3^2\sigma_3^2$

With the introduction of risk, a firm is no longer indifferent between two investment proposals having equal net present values or internal rates of return. Both net present value and its standard deviation (σ_{NPV}) should be estimated in performing capital-budgeting analyses under uncertainty. A NPV approach under uncertainty can be defined as

$$\widetilde{NPV} = \sum_{t=1}^{N} \frac{\widetilde{C_t}}{(1+k)^t} + \frac{S_n}{(1+k)^N} - I_0 \tag{21.3}$$

where C_t is the uncertain net cash flow in period t, k is the risk-adjusted discount rate, S_n is the salvage value, and I_0 is the initial outlay. The mean of the NPV distribution and its standard deviation can be defined as:

$$\overline{NPV} = \sum_{t=1}^{N} \frac{\overline{C}_t}{(1+k)^t} + \frac{S_n}{(1+k)^N} - I_0, \tag{21.4}$$

$$\sigma_{NPV} = \left[\sum_{t=1}^{N} \frac{\sigma_t^2}{(1+k)^{2t}} + \sum_{\tau=1}^{N}\sum_{t=1}^{N} w_t w_\tau\, Cov\,(C_\tau,\, C_t)\right]^{1/2} \quad (\tau \neq t), \tag{21.5}$$

where

σ_t^2 = variance of cash flows in the tth period.

W_t, W_τ = discount factor for the tth and the τth period; respectively, that is,

$$W_t = \frac{1}{(1+K_f)^t} \quad \text{and} \quad W_\tau = \frac{1}{(1+K_f)^\tau}$$

$\mathrm{Cov}(C_t, C_\tau)$ = covariability between cash flows C_τ and C_t.

Cash flows between periods t and τ are generally related. Therefore, $\mathrm{Cov}(C_t, C_\tau)$ is an important factor in the estimation of σ_{NPV}. The magnitude, sign, and degree of the relationships of these cash flows depend on the economic operating conditions and on the nature of the product or service being produced. If there are only three periods, then all terms within parenthesis in Eq. 21.5 can be presented as in Table 21.3. The summation of the diagonal elements $[W_1^2\sigma_1^2, W_2^2\sigma_2^2, W_3^2\sigma_3^2]$ of Table 21.3 results in the first part of Eq. 21.5, or

$$\sum_{t=1}^{N} \frac{\sigma_t^2}{(1+K_f)^{2t}}$$

The summation of all other elements in Table 21.3 gives the second portion of Eq. 21.5, or

$$\sum_{t=1}^{N}\sum_{\tau=1}^{N} W_t W_\tau \,\mathrm{Cov}(C_\tau,\, C_t), \quad t \neq \tau$$

Equation 21.5 is the general equation for σ_{NPV}. Both Eq. 21.7 for σ_{NPV} under perfectly positively correlated cash flow and Eq. 21.8 for independent cash flows are special cases derived from the general Eq. 21.5. If $\rho_{12} = \rho_{13} = \rho_{23} = 1$, then $\mathrm{Cov}(C_1, C_2) = \sigma_1\sigma_2$, $\mathrm{Cov}(C_1, C_2) = \sigma_1\sigma_3$, and $\mathrm{Cov}(C_2, C_3) = \sigma_1\sigma_3$. Therefore, Eq. 21.5 reduces to

$$\begin{aligned}\sigma_{NPV} &= \left(\frac{\sigma_1^2}{(1+K_f)^2} + \frac{\sigma_2^2}{(1+K_f)^4} + \frac{\sigma_3^2}{(1+K_f)^6} + \frac{2\sigma_1\sigma_2}{(1+K_f)^3} + \frac{2\sigma_1\sigma_3}{(1+K_f)^4} + \frac{2\sigma_2\sigma_3}{(1+K_f)^5}\right)^{\frac{1}{2}} \\ &= \sum_{t=1}^{3} \frac{\sigma_t}{(1+K_f)^t}\end{aligned}$$

which is Eq. 21.7.

$\mathrm{Cov}(C_\tau, C_t)$ is used to measure the covariability between the cash flow in the tth and τth periods. It is well known that:

$$\mathrm{Cov}(C_t, C_\tau) = \rho_{\tau t}\sigma_\tau\sigma_t, \tag{21.6}$$

where $\rho_{\tau t}$ is the correlation coefficient between the cash flow in the ith period and that in the τth period. In one special case, the cash flows are perfectly correlated over time. If such is the case, then the deviation of the actual cash flow from the mean of its corresponding profitability distribution of expected cash flows for the period gives us the information that cash flows in future periods will deviate in a similar fashion. In other words, Eq. (21.5) can be rewritten as:[3]

$$\sigma_{NPV} = \sum_{t-1}^{N} \frac{\sigma_t}{(1+k)^t}. \tag{21.7}$$

If the cash flows are mutually independent, then Eq. (21.5) reduces to:

$$\sigma_{NPV} = \sqrt{\sum_{t-1}^{N} \frac{\sigma_t^2}{(1+k)^{2t}}}. \tag{21.8}$$

Hillier (1963) combines the assumption of mutual independence and perfect correlation in developing a model to deal with mixed situations. The model as defined in Eq. (21.9) can be used to analyze investment proposals in which some of the expected cash flows are closely related, and others are fairly independent.

$$\sigma = \sqrt{\sum_{t=1}^{N} \frac{\sigma_{yt}^2}{(1+k)^{2t}} + \sum_{h}^{m} \left(\sum_{t=1}^{N} \frac{\sigma_{zt}^h}{(1+k)^t} \right)^2}, \tag{21.9}$$

where σ_{yt}^2 is the variance for an independent net cash flow in period t, and σ_{zt}^h is the standard deviation for stream h of a perfectly correlated stream of cash flows t. It is clear that Eq. (21.9) is a combination of Eqs. (21.7) and (21.8). Chen and Moore (1982) have generalized this model by introducing the estimation risk.

They argue that Hillier's (1963) approach is unrealistic due to the assumption of parameter certainty. In reality, the parameters of the probability distribution are never known with certainty and thus the capital budgeting decisions based on Hillier's approach may be potentially misleading because this approach ignores estimation risk. Chen and Moore (1982), in hence, extend Hillier's approach with a Bayesian framework, which leads to the derivation of a predicative distribution that is not conditioned on unknown parameters. Therefore, the effect of estimation risk in capital budgeting decisions can be eliminated in Chen and Moore (1982) approach.

In addition, Wang and Lee (2010) provide the fuzzy real option valuation approach to solve the capital budgeting decisions under uncertainty environment. In Wang and Lee's model framework, the concept of probability is employed in describing fuzzy events, and the estimated cash flows are based on the fuzzy numbers that can better reflect the uncertainty in the project. By using a fuzzy real option valuation, the managers can select fuzzy projects and determine the optimal time to abandon the project under the assumption of limited capital budget.

To illustrate the model of Eq. (21.9), suppose that a firm is considering the introduction of a new product with returns expected over the next 5 years. Because the product's market reception is uncertain, management feels that, if initial reception exceeds expectations, actual acceptance in later years will also exceed expectations, in approximately the same proportions. For simplicity, it is believed that the net marketing cash flow (sales less marketing and advertising expenses) can be treated as perfectly correlated over time.

On the other hand, estimates of the initial investment in the project and the production costs are reasonably reliable, so that any deviation from expectations is assumed to be attributable to random fluctuations. Consequently, initial investment and net production cash flows are regarded as being mutually independent over time. The probability information for the introduction of the new product is shown in Table 21.4. We assume that each of the probability distributions involved can be regarded as normal. If 10% is used as the risk-free rate, the expected value of the net present value for the proposal is

Table 21.4 Expected cash flow for new product

Year	Source	Expected value of net cash flow (in thousands)	Standard deviation (in thousands)
0	Initial investment	−$600	$50
1	Production cash outflow	−250	20
2	Production cash outflow	−200	10
3	Production cash outflow	−200	10
4	Production cash outflow	−200	10
5	Production outflow − salvage value	−100	15
1	Marketing	300	50
2	Marketing	600	100
3	Marketing	500	100
4	Marketing	400	100
5	Marketing	300	100

$$NPV = \sum_{t=0}^{5} \frac{C_t}{(1.04)^t} = -600 + \frac{300-250}{(1.04)} + \frac{600-200}{(1.04)^2} + \frac{500-200}{(1.04)^3} + \frac{400-200}{(1.04)^4} + \frac{300-100}{(1.04)^5} = \$419.95 = \$420.00.$$

Following Eq. (21.7b), we can calculate the standard deviation as:

$$\sigma_{NPV} = \left[50^2 + \frac{20^2}{(1.04)^2} + \cdots + \frac{15^2}{(1.04)^{10}} + \left(\frac{50}{1.04} + \ldots + \frac{100}{(1.04)^5}\right)^2\right]^{1/2} = \$398.$$

Reprinted by permission of Hillier, "The derivation of probabilistic information for the evaluation of risky investments," *Management Science* 9 (April 1963). Copyright 1963 by The Institute of Management Sciences.

Thus the expected net present value of the proposal is $420, and the standard deviation of the distribution relating to NPV is $398. Using this information, we can express an interval inference for NPV as;

$$\Pr(\$420 \pm \$398) = 68.27\%;$$

$$\Pr(\$420 \pm (2)(\$398)) = 95.45\%;$$

$$\Pr(\$420 \pm (3)(\$398)) = 99.73\%.$$

Hillier's approach takes us a long way toward coping with the correlation of cash flows over time, but his method is not always practically applicable. For many investment proposals, cash flows fall into neither of these categories, since in practice they show less than perfect correlation over time. If the correlation is moderate, then the classification suggested by Hillier will not be appropriate. The method suggested by Bonini (1975) and others for handling the problem of moderate correlation is with a series of conditional-probability distributions. Wang and Lee (2010) have expended Bonini's result in terms of fuzzy set and real option approach.

To illustrate the use of this model, we refer to Table 21.5, where we consider a project requiring an initial outlay of $10,000 (Column 1). In each of the following time periods (Columns 3, 5, and 7), the cash flow to be received is not known with perfect certainty, but the probabilities associated with each cash flow in each period are assumed to be known (Columns 2, 4, and 6), so that we are dealing with a case of risk, and not strict uncertainty. We note that Columns (4) and (6) are conditional-probability figures, where the later periods' expected cash flows depend highly on what occurs in earlier time periods. Given our cash flow and simple probability estimates for the periods 1, 2, and 3, we find 27 possible joint

Table 21.5 Illustration of conditional-probability distribution approach

Initial outlay Period 0 (1)	Initial Probability P(1) (2)	Net Cash Flow (3)	Conditional probability P(3|2,1) (4)	Net cash flow (5)	Conditional probability P(3|2,1) (6)	Cash flow (7)	Joint probability (8)
			0.25	2000	0.2	1000	0.015
					0.5	2000	0.0375
					0.3	3000	0.0225
					0.2	2000	0.03
10,000	0.3	2000	0.5	3000	0.5	3000	0.075
					0.3	4000	0.045
					0.2	3000	0.015
			0.25	4000	0.5	4000	0.0375
					0.3	5000	0.0225
					0.3	2000	0.045
			0.3	3000	0.4	4000	0.06
					0.3	6000	0.045
					0.3	4000	0.06
	0.5	4000	0.4	5000	0.4	5000	0.08
					0.3	6000	0.06
					0.3	5000	0.045
			0.3	7000	0.4	7000	0.06
					0.3	9000	0.045
					0.25	4000	0.0125
			0.25	5000	0.5	6000	0.025
					0.25	8000	0.0125
					0.25	5000	0.025
	0.2	6000	0.5	7000	0.5	7000	0.05
					0.25	9000	0.025
					0.25	7000	0.0125
			0.25	8000	0.5	9000	0.025
					0.25	11,000	0.0125

probabilities (Column 8), each of which corresponds to a cash-flow series. Thus in the uppermost path, we find the joint probability 0.015 being associated with a cash flow of $2000 in Period 1, followed by cash flows of $2000 and $1000 in Periods 2 and 3, respectively. This particular path is the worst possible result; in nondiscounted dollar terms there is a 50 percent loss on the investment. At the other end of the spectrum, we find the best possible outcome offers a return $12,945 (the last line of Table 21.6) with a joint probability of 0.0125. The returns and joint probabilities can be calculated similarly for the other 25 possible patterns.

While the primary hurdle in this process is in estimating the cash flows and the probabilities, we recognize that it is the NPV figures we are ultimately interested in. Table 21.8b gives the present value of the cash flows and the net present value of the project for each cash-flow series, where a constant 4 percent discount rate was employed. When we multiply each joint probability by the expected net present value associated with that probability, we obtain the expected net present value for the project as a whole, there shown to be $2,517.18. Also, from all 27 expected net-present-value figures, we can calculate the variance and the standard deviation for the project as a whole, shown in the lower portion of Table 21.8b as $20,359,090.91 and $4,512.11, respectively, where these figures can be used to assess the riskiness of the project.

The use of this form of conditional-probability distribution enables us to take account of the dependence or correlation in the cash flows over time. In the above example, the cash flow in Period 3 depends upon the size of the cash flow in Periods 2 and 1. Given the cash flow in Period 1, the cash flow in Period 2 can vary within a range, but more of a known range than if we did not know the cash flow in Period 1. Similarly, the cash flows in Period 3 are allowed to vary within a range of values,

Table 21.6 NPV and joint probability

PV	NPV	Probability
4,661.2	−5,338.8	0.015
5,550.2	−4,449.8	0.0375
6,439.2	−3,560.8	0.0225
6,474.76	−3,525.24	0.03
7,363.76	−2,636.24	0.075
8,252.76	−1,747.24	0.045
8,288.32	−1,711.68	0.015
9,177.32	−822.68	0.0375
10,066.32	66.32	0.0225
8,297.84	−1,602.16	0.045
10,175.84	175.84	0.06
11,953.84	1,953.84	0.045
12,024.96	2,024.96	0.06
12,913.96	2,913.96	0.08
13,802.96	3,802.96	0.06
14,763.08	4,763.08	0.045
16,541.08	6,541.08	0.06
18,319.08	8,319.08	0.045
13,948.04	3,948.04	0.0125
15,726.04	5,726.04	0.025
17,504.04	7,504.04	0.0125
16,686.16	6,686.16	0.025
18,464.16	8,464.16	0.05
20,242.16	10,242.16	0.025
19,388.72	9,388.72	0.0125
21,166.72	11,166.72	0.025
22,944.72	12,944.72	0.0125

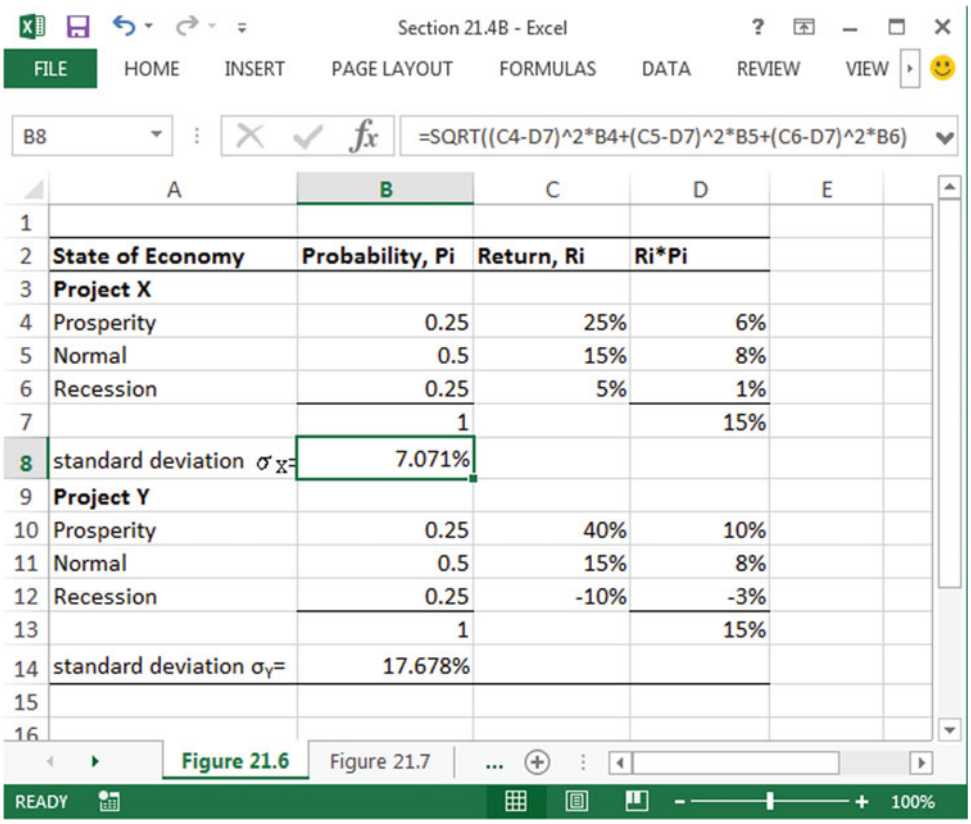
the range being determined in part by the realized cash flow in the prior period. Most importantly it should be noted that we assume we know what the distribution of each cash-flow figure is, and the probabilities associated with each flow, and, further, that once we make the investment decision we more or less have locked ourselves into the project. In the following section, we relax this latter assumption and allow sequential decisions to be introduced into the conditional-probability decision framework.

21.4.2 Excel and VBA Application

In this section, we will show how Excel and VBA can be used to calculate Eqs. 21.4, 21.5, and 21.7. First, we will show how excel program can be used to calculate mean and standard deviation in a continuous data present in Table 21.6. In Table 21.6, the first column PV represents the summation of present value of future net cash inflow. The second column represents net present value, which is PV minus initial outlay. Finally, this third column represents the probability of occurrence.

Discount rate = 4%
NPV = $2,517.182.
Variance = $20,359,090.9093.
Standard deviation = $4,512.105.

Fig. 21.6 Input data



	A	B	C	D	E
1					
2	**State of Economy**	**Probability, Pi**	**Return, Ri**	**Ri*Pi**	
3	**Project X**				
4	Prosperity	0.25	25%	6%	
5	Normal	0.5	15%	8%	
6	Recession	0.25	5%	1%	
7		1		15%	
8	standard deviation σ_X=	7.071%			
9	**Project Y**				
10	Prosperity	0.25	40%	10%	
11	Normal	0.5	15%	8%	
12	Recession	0.25	-10%	-3%	
13		1		15%	
14	standard deviation σ_Y=	17.678%			
15					
16					

Now we discuss how we can use either Excel program or VBA to calculate mean and variance in terms of the data present in Fig. 21.6. In this table, column A presents state of economy, column B presents probability of occurrence, column C presents rate of return and column D presents column B times column C. By using this information, we can use excel program to calculate both mean and standard deviation for projects X and Y. From Fig. 21.6, we found both rate of return for projects X and Y are 15%, however, standard deviations for project X is 7.071% and for project Y is 17.678%. Therefore, we can conclude that we should prefer project X than project Y.

Figure 21.7 presents how excel program can be used to calculate Eqs. 21.4 and 21.7. In Fig. 21.7, we have the information of initial investments, net cash inflow, and standard deviation associated with net cash inflow for both project A and B. From the excel output, we found that NPV for projects A and B are both $18.92. However, the standard deviation for project A is 15.16 that is higher than the standard deviation of project B that is 7.58. Therefore, we will conclude project B will perform better than project A.

Now we will generalize the previous example by allowing the cash flow between different periods are not positively and perfectly correlated. In other words, we will use Eqs. 21.4 and 21.5 to perform capital budgeting decision under uncertainty. There are two approaches to calculate σ_{NPV}.

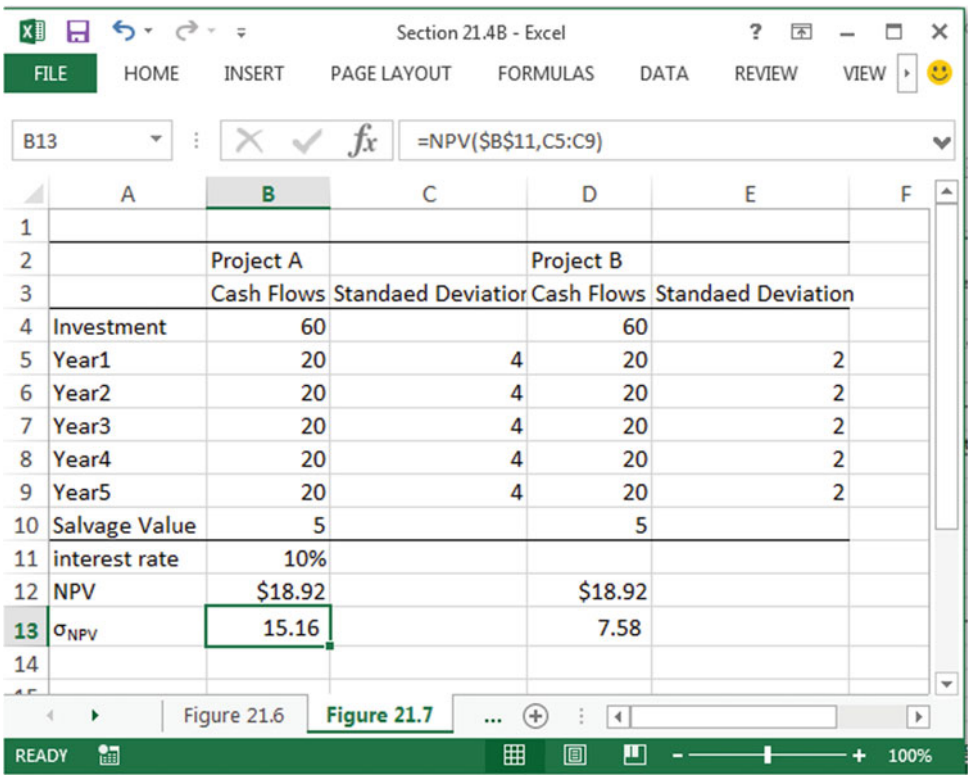
The first one use Excel matrix multiply and transpose to calculate σ_{NPV}. The function in Cell B10 is

$$= \text{SQRT}(\text{MMULT}(\text{MMULT}(\text{TRANSPOSE}(\text{G4} : \text{G8}), \text{B4} : \text{F8}), \text{G4} : \text{G8}))$$

Because it is one kind of matrix arithmetic, after type the function in the cell we have to press Ctrl + Shift + Enter to get the result, which is presented in Fig. 21.8.

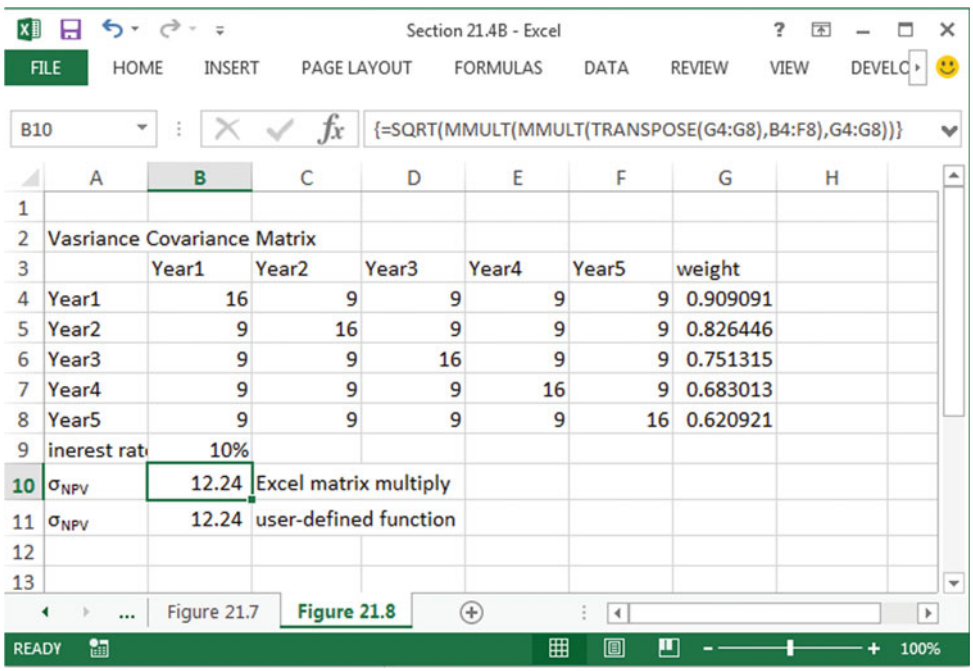
The second one uses the VBA to write a user-defined function to calculate standard deviation of NPV. Here is the VBA code.

```
Function SigmaNPV(vcvm, rate)
    Dim n, i, j
    Dim sigmasum() As Variant
    n = vcvm.Columns.Count
```



	A	B	C	D	E	F
1						
2		Project A		Project B		
3		Cash Flows	Standaed Deviatior	Cash Flows	Standaed Deviation	
4	Investment	60		60		
5	Year1	20	4	20	2	
6	Year2	20	4	20	2	
7	Year3	20	4	20	2	
8	Year4	20	4	20	2	
9	Year5	20	4	20	2	
10	Salvage Value	5		5		
11	interest rate	10%				
12	NPV	$18.92		$18.92		
13	σ_{NPV}	15.16		7.58		
14						

Fig. 21.7 Information for capital budgeting decision



	A	B	C	D	E	F	G	H
1								
2	Vasriance Covariance Matrix							
3		Year1	Year2	Year3	Year4	Year5	weight	
4	Year1	16	9	9	9	9	0.909091	
5	Year2	9	16	9	9	9	0.826446	
6	Year3	9	9	16	9	9	0.751315	
7	Year4	9	9	9	16	9	0.683013	
8	Year5	9	9	9	9	16	0.620921	
9	inerest rat	10%						
10	σ_{NPV}	12.24	Excel matrix multiply					
11	σ_{NPV}	12.24	user-defined function					
12								
13								

Fig. 21.8 A general formula for the standard deviation of NPV

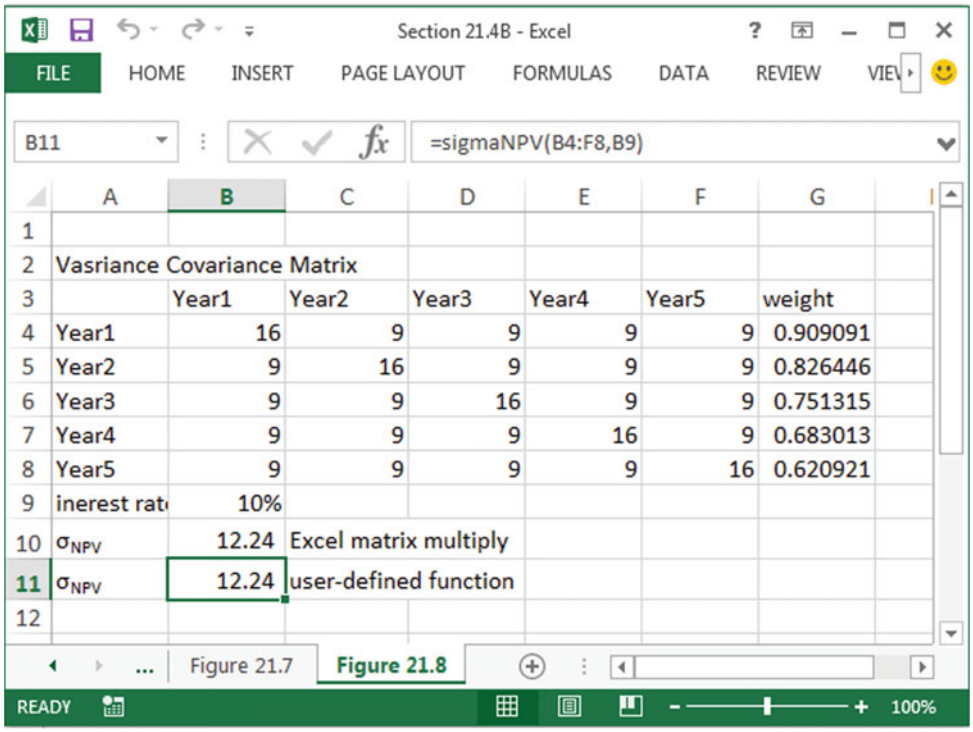


Fig. 21.9 A general formula for the standard deviation of NPV by using VBA

```
    ReDim sigmasum(n, n)
    For i = 1 To n
        For j = 1 To n
          sigmasum(i, j) = vcvm(i, j) / (1 + rate) ^ (i + j)
        Next j
    Next i
    SigmaNPV = Sqr(Application.Sum(sigmasum))
End Function
```

Figure 21.9 shows the result of using user-defined VBA function to calculate standard deviation (sigma) of NPV. Below in the Cell B11

$$= \text{sigmaNPV}(\text{B4} : \text{F8}, \text{B9})$$

*variance needs to be changed in Fig. 21.9 to variance.

In Fig. 21.9, we input the information of Variance–Covariance Matrix among five periods. This information is generally calculated by input data of cash flow information related to the five periods of a project.

By either method, we find the standard deviation of NPV is 12.24. By following the same procedure discussed in the previous example, we can perform capital budgeting under uncertainty.

Both the NPV method and statistical method for capital budgeting decision will be discussed in Chap. 28.

21.5 Summary

In this chapter, we have shown how excel program can be used to do (i) Decision Trees and expected monetary values, (ii) NPV and IRR methods for capital budgeting decision under certainty. (iii) We use either excel program or VBA program to perform statistical distribution method for capital budgeting under uncertainty. In Part B of this book, we will discuss excel and VBA programs in further detail.

Bibliography

Anderson, David R, and Sweeney, Dennis J, and Williams, Thomas A, Statistics for Business and Financial Economics 14th edision, AbeBooks Seller, 2020.

Chen, SN and WT Moore (1982). Investment decision under uncertainty: Application of estimation risk in the Hillier approach. *Journal of Financial and Quantitative Analysis*, 17 (September), 425-440.

Excel 2013 is published by Microsoft Inc., One Microsoft Way, Redmond, WA, USA.

Hillier, F (1963). The derivation of probabilistic information for the evaluation of risky investments. *Management Science*, 9 (April), 443-457.

Kusleika, Dick and Alexander, Michael. "Excel 2019 Power Programming With Vba.", Wiley, 2019.

Lee, CF, JC Lee and AC Lee, "Statistics for Business and Financial Economics", Springer, New York 2013.

Minitab 17, Minitab Inc., State College, Pennsylvania, USA.

SAS 2014 is published by SAS Institute Inc., 100 SAS Campus Drive, Cary, NC, USA.

Telles, Matt. "Python for Professionals: Learning Python as a Second Language.", BPB Publications, 2019.

Wang, SY and CF Lee (2010) A fuzzy zeal option valuation approach. International Journal of Information Technology and Decision Making Volume 09: Issue 05, Sep.2010, 695–714.

Wickham, Hadley and Grolemund, Garrett. "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data.", O'Reilly Media, 2017.

Part II
Portfolio Analysis

Risk Classification, Estimation, and Diversification

22

22.1 Introduction

This chapter presents the basic concepts of risk in the context of business operations, corporate financing decisions, and diversification. Methods of measuring risk are explained by using basic statistical procedures. Also included are the concepts and applications of the dominance principle and portfolio theory.

We begin the chapter by discussing the types and classifications of risk in Sect. 22.2. This is followed by Sect. 22.3 on the concepts of portfolio analysis and its applications. Determination of the commercial lending rate in accordance with risk and return concepts is discussed in Sect. 22.4. Section 22.5 discusses the dominance principle and the necessity of using performance measures to compare the performance of different portfolios. Finally, in Sect. 22.6, the results of this chapter are summarized. Appendix 22A explores the normal distribution. Appendix 22B discusses how minimum-variance approach can be used to derive the optimal weight for a two-security portfolio. In Appendix 22C we show how the Sharpe performance measure approach can be used to derive the optimal weight for a two-security portfolio. Finally, in Appendix 22D we calculate beta coefficients, Sharpe performance measures, and optimal portfolio weights in terms of five companies and the S&P index.

22.2 Risk Classification

Earlier, we discussed the rates of returns on stocks. In this chapter, we discuss the risk associated with the rate of return. Risk is defined as the probability of success or failure, where success is making a profit and failure is losing money on an investment. In finance and accounting, we refer to risk as the degree of fluctuation associated with either return on investment (ROI) or the rates of return on securities. Depending on its sources, risk can be classified as business risk or financial risk. Sources of risk are useful for both corporate managers and shareholders to determine and assess a firm's market value.

22.2.1 Business Risk

Business risk, sometimes called operating risk, refers to the degree of fluctuation of net income and cash flow associated with different types of businesses and operating strategies. The fluctuation of operating income represents a business risk. Business risk to shareholders generally arises from uncertainty about the firm's product markets, input markets, or operations. Business risk also can be inherent in the firm's commercial activities. If a firm is entirely equity funded, then the fluctuation of return on investment, earnings-per-share (EPS), and rate of equity (ROE) can be used to determine business risk.

The possible ROIs of General Motors (company A) and Ford (company B) for the coming year are listed in Table 22.1.

In Table 22.1, we see that there is a 20% chance of a boom economy in the coming year, in which both GM and Ford will have a high ROI; and a 20% chance of a recession, which will mean low ROI for both GM and Ford.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_22

Table 22.1 ROI and state of economy

State of economy	State occurring (p_i)	ROI (A_1) of GM(X_{AI}) (%)	ROI (B_i) of Ford (X_{Bi}) (%)
Boom	0.20	15	14
Normal	0.60	12	11
Recession	0.20	4	3

The expected ROI and the variance of ROI for these two auto manufacturers can be calculated by Eqs. 22.1 and 22.2:

$$\text{Expected ROI} = \overline{X} = \sum_{i=1}^{n} X_i P_i \tag{22.1}$$

$$\text{Variance of ROI} = \sigma^2 = \sum_{i=1}^{n} (Xi - \overline{X})^2 P_i \tag{22.2}$$

where P_i = probability of occurrence of the *i*th state of the economy, and X_i = ROI in the *i*th state of the economy.

Substituting the data of Table 22.1 into Eqs. 22.1 and 22.2, we have.

GM: $\overline{X}_A = (0.15)(0.2) + (0.12)(0.6) + (0.4)(0.2) = 11\%$

Ford: $\overline{X}_B = (0.14)(0.2) + (0.11)(0.6) + (0.3)(0.2) = 10\%$

GM: $\sigma_A^2 = (0.15 - 0.11)^2(0.2) + (0.12 - 0.11)^2(0.6) + (0.4 - 0.11)^2(0.2) = 0.00136$.

Ford: $\sigma_B^2 = (0.14 - 0.10)^2(0.2) + (0.11 - 0.10)^2(0.6) + (0.3 - 0.1)^2(0.2) = 0\ .00136$.

These results show that the expected ROIs are 11% for GM and 10% for Ford, with a variance of ROI for both firms of 0.00316. Thus, the absolute business risk is about the same for both firms.

To simultaneously compare these firms in terms of expected returns and risk, we use the coefficient of variation as indicated in Eq. 22.3:

$$\text{Coefficient of variation(CV)} = \frac{\sigma}{\overline{X}} \tag{22.3}$$

The CV for GM and Ford is determined as follows:

$$\text{GM:} \quad \text{CV}_A \frac{\sqrt{0.00136}}{0.11} = \frac{0.369}{0.11} = 0.3353$$

$$\text{Ford:} \quad \text{CV}_B = \frac{\sqrt{0.00136}}{0.10} = \frac{0.0369}{0.10} = 0.3690$$

These coefficients of variation show that the risk per unit of return for Ford is greater than the risk per unit of return for GM. That is, even though both companies face the same business risk, when we simultaneously consider both risk and return, GM is superior. It should be noted that this result does not necessarily hold in a portfolio context, where the appropriation risk measure is not only the variance or the coefficient of variation but is also the correlation coefficient. (This is discussed in greater detail later in this chapter.)

If the probability distribution of the firms' ROI is normal, then the mean and standard deviation of ROIs for both GM and Ford can be used to calculate the probability that ROI will be greater or less than a certain amount. We will now discuss the method that is used to standardize the distribution in order to determine how many standard deviations a specific ROI is from the mean. Suppose we want to know the probability that the actual ROI for GM will be greater than 15.44%. In standardizing the distribution (see Appendix 22A), we obtain:

$$Z = \frac{15.44\% - 11\%}{3.69\%} = 1.20 \text{ Standard deviations}$$

Referring to the normal distribution table found in the appendix at the end of this book, we find that there is a 0.885 probability that actual ROI will be greater than 15.44%.

For a normal distribution, the actual ROI will lie within ±1 standard deviation of expected ROI 68.27% of the time the actual ROI will be within ±2 standard deviation of expected ROI 95.45% of the time, and the actual ROI will lie within ±3

standard deviation of expected ROI 99.73% of the time. These interval estimates of ROI for both GM and Ford are shown below:

	Interval	Chance of occurrence (%)
GM	0.11 – 0.037 < X < 0.11 + 0.037	68.27
	0.11 – 0.074 < X < 0.11 + 0.074	95.45
	0.11 – 0.111 < X < 0.11 + 0.111	99.73
Ford	0.10 – 0.037 < X < 0.10 + 0.037	68.27
	0.11 – 0.074 < X < 0.11 + 0.074	95.45
	0.11 – 0.111 < X < 0.11 + 0.111	99.73

This information is useful for investors because it enables them to make risk comparisons between GM and Ford. In this case, the standard deviations of ROI for both companies are the same, implying that the risk for each is the same. Hence, GM is preferable because it has a higher return for the same level of risk.

22.2.2 Financial Risk

Financial leverage is the magnification of profit or loss through the use of debt financing. Financial risk refers to the risk of not being able to meet the fixed cash flow requirement of debt financing, which could lead to bankruptcy. Hence, financial risk is related to financial leverage. The amount of financial leverage is determined by the amount of short- and long-term debt that the firm employs in its capital structure. One of the ways to measure financial risk is by the firm's debt/equity ratio. The relationship between total risk and financial leverage (debt/equity) is indicated in Fig. 22.1. Business risk is unrelated to the amount of financial leverage, as shown by line AB. But the financial risk is directly related to the amount of leverage, as shown by line AC. The sum of business risk and financial risk is the firm's total risk. The combination of business risk and financial risk plays a central role in the theory and practice of corporate finance.

Business risk and financial risk are not necessarily constant over time. Both can be affected by fluctuations in the business cycle or by changes in the federal government's fiscal policy and the Federal Reserve System's monetary policy. Hence, economic information is relevant to the financial manager when assessing the riskiness of a given situation.

22.2.3 Total Risk

Total risk refers to the combined effects of business risk and financial risk, as indicated in Fig. 22.1. Total risk is defined as

$$\text{Total risk} = \text{Business Risk} + \text{Financial Risk}$$

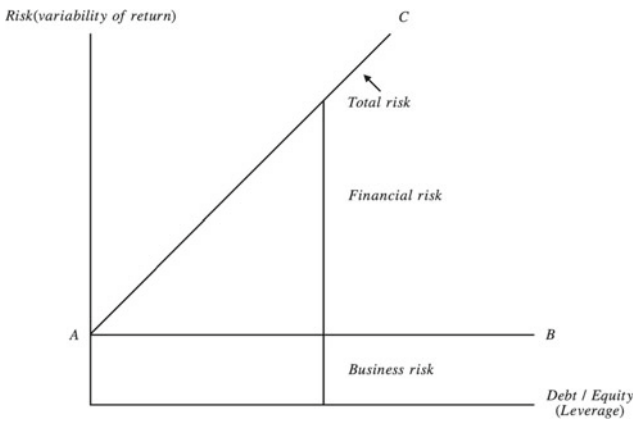


Fig. 22.1 Relationship of total risk to financial leverage

In the case of multinational firms or firms with international investments, business risk can take on an international dimension. Exchange rate fluctuations, government stability, and segmented markets are some components of international risk. Exchange rate risk deals with the foreign exchange rate between the firm's home country and the foreign country where its investments are being made. The exchange rate is the price of foreign currency in terms of the domestic monetary unit. A change is the price of foreign currency in terms of the domestic monetary unit. A change in the exchange rate will produce a change in price of foreign goods in terms of domestic currency, as well as a proportionate change in the price of domestic goods in terms of the foreign currency.

For example, on September 24, 2007, the exchange rate between the U.S. dollar and the Chinese Yuan was $0.1331 to 1 Yuan. Suppose that on September 25, 2007, the Yuan was devalued 10%, such that $0.1198 was equal to 1 Yuan. For a U.S. consumer, an item costing 1,000 Yuan in China would cost $133.1 on September 24. On September 25, this same item would be $119.8. Therefore, the U.S. will import more goods from China because the depreciation in the value of the Chinese currency makes Chinese exports cheaper. For the Chinese consumer, an item worth $133.1 would cost 1,000 Yuan on September 24 and 1,110 Yuan on September 25. This increase in price, caused by the Yuan depreciation, would result in a decreased demand for American imports. Therefore, an appreciation in the Yuan would cause an opposite effect, an increase in demand for imports. It is apparent that firms with business dealings in foreign countries must be aware of foreign exchange fluctuations. A profit made in a foreign country could be easily translated into a loss with a sudden appreciation in the U.S. dollar.

A second type of international business risk is political risk, which refers to the political climate and conditions of the country. For instance, a country with an unstable government may experience wild fluctuations in interest rates because of a lack of confidence from the business community. Or there may be a chance that the government will expropriate plant and equipment or negate contracts. These are all political risks that a firm must include in its risk analysis when it operates internationally. Such risks are also included in the business risk component of total risk.

22.3 Portfolio Analysis and Application

A portfolio is any combination of assets or investments. Portfolio analysis is used to determine the return and risk characteristics for any combination of assets. As first described by Markowitz (1959), portfolio analysis involves the identification of specific statistical characteristics that enable managers or investors to systematically reduce risk through diversification; that is, holding more than a single asset in a portfolio.

22.3.1 Expected Rate of Return on a Portfolio

The rate of return on a portfolio is simply the weighted average of the returns of individual securities in the portfolio. This is a linear relationship. For example, suppose 40% of the portfolio is invested in security a and 30% is invested in security b and c, respectively. If the expected rate of return of securities a, b, and c is 10%, 5%, and 12%, respectively, then the expected rate of return for this portfolio is

$$\begin{aligned}\overline{R}_P &= W_a\overline{R}_a + W_b\overline{R}_b + W_c\overline{R}_c \\ &= (0.4)(0.1) + (0.3)(0.05) + (0.3)(0.12) \\ &= 0.091\end{aligned}$$

where W_a, W_b, and W_c are the propositions of the portfolio invested in securities a, b, and c. The summation of these weights equals 1. $\overline{R}_a, \overline{R}_b,$ and $\overline{R}_c$ represent expected rates of return on securities a, b, and c. $\overline{R}_p$ represents the expected rate of return of return for the portfolio, which is the weighted average of the securities' expected rates of return. In general, the expected return on an n-asset portfolio is defined by Eq. 22.4:

$$\overline{R}_p = \sum_{i=1}^{n} W_i\overline{R}_i \left(\text{where } \sum_{i=1}^{n} W_i = 1\right) \tag{22.4}$$

where W_i is the proportion of the individual's investment allocated to security i, and $\overline{R}_i$ is the expected rate of return for security i.

If an investor chooses not to invest the entire portfolio in financial assets, then one of the investor's assets will be cash. Hence, the individual investor must hold 100% of his or her financial wealth in some form of profitable assets and cash.

22.3.1.1 Variance and Standard Deviation of a Portfolio

A portfolio's risk is measured by its standard deviation, or variance. The standard deviation is the square root of the variance. To calculate a portfolio's standard deviation, we should first identify the covariance among the securities in the portfolio. The covariance between two such securities is defined as follows:

$$\begin{aligned} COV(W_1R_1, W_2R_2) &= \frac{1}{N}\sum_{t=1}^{N}(W_1R_{1t} - W_1\overline{R}_1)(W_2R_{2t} - W_2\overline{R}_2) \\ &= \frac{W_1W_2}{N}\sum_{t=1}^{N}(R_{1t} - \overline{R}_1)(R_{2t} - \overline{R}_2) \\ &= W_1W_2Cov(R_1, R_2) \end{aligned} \tag{22.5}$$

where R_{1t} is the rate of return for the first security in period t and R_{2t} is the rate of return for the second security in period t. $\overline{R}_1$ and $\overline{R}_2$ are average rates of return for the securities, while COV (R_1, R_2) represents the covariance between R_1 and R_2.

$$\begin{aligned} \sigma_P^2 = VAR(W_1R_{1t} + W_2R_{2t}) &= \frac{1}{N}\sum_{t=1}^{N}\left[(W_1R_{1t} + W_2R_{2t}) - \left(W_1\overline{R}_1 + W_2\overline{R}_2\right)\right]^2 \\ = W_1^2\sigma_1^2 + W_2^2\sigma_2^2 + 2W_1W_2COV(R_1, R_2) \end{aligned} \tag{22.6}$$

Now that we have shown how to find covariance and variance of a portfolio, we proceed to the standard deviation of returns for a portfolio, which is determined as follows:

$$\sigma_P = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(R_{pt} - \overline{R}_P)^2} \tag{22.7}$$

where σ_p is the standard deviation of the portfolio's return, R_{pt} is the portfolio's rate of return in period t, and $\overline{R}_p$ is the average portfolio rate of return. Figure 22.2 illustrates two possible distributions for two portfolios with the same expected return but with different degrees of risk. Portfolio B's variability is larger than portfolio A's. Consequently, investors regard portfolio B as riskier than portfolio A. Therefore, portfolio A is more attractive than portfolio B because it offers less risk with the same expected return.

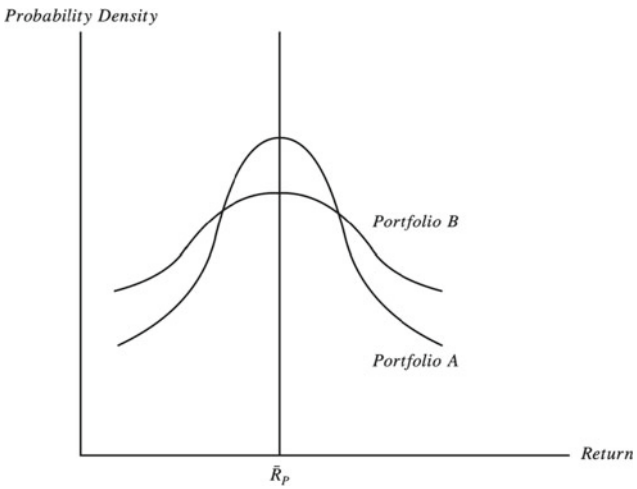


Fig. 22.2 Probability distribution of returns for two portfolios

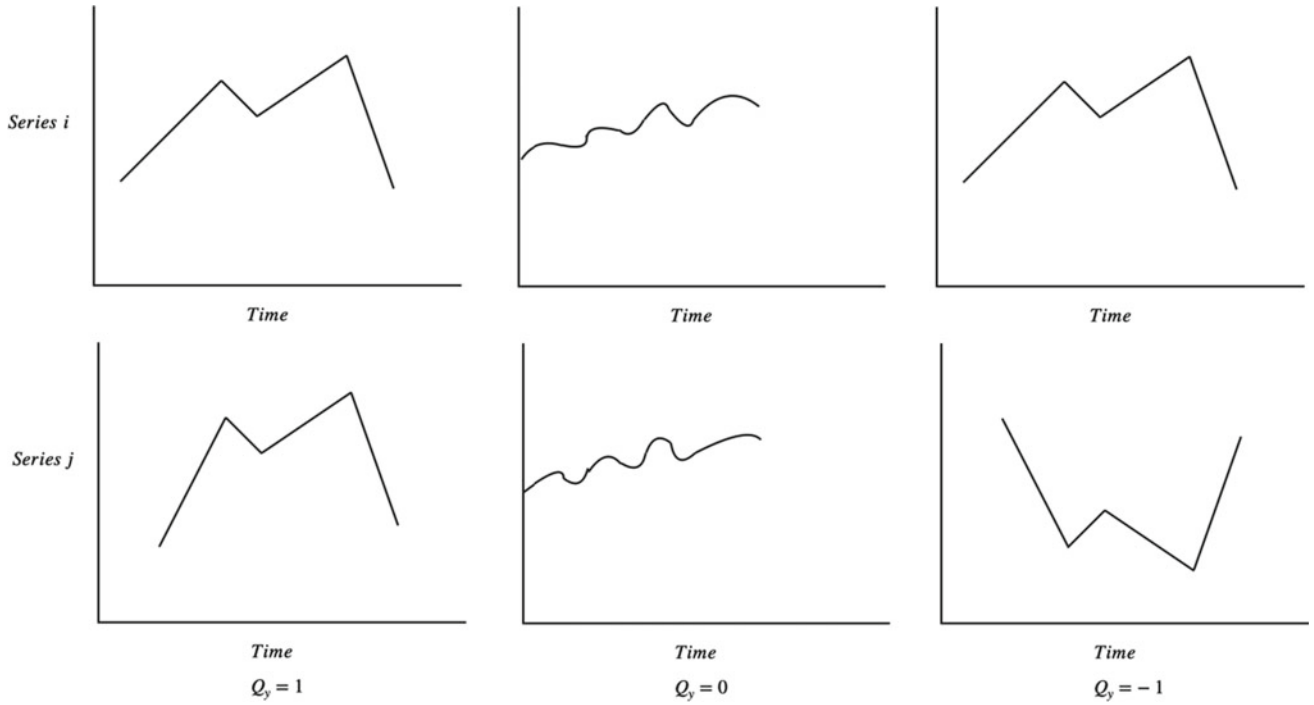


Fig. 22.3 Relationship of total risk to financial leverage

22.3.2 The Two-Asset Case

To explain the fundamental aspect of portfolio risk diversification, we consider the two-asset case. Following Eqs. 22.6 and 22.7, we define σ_p as follows:

$$\sigma_P = \left[W_1^2 VAR(R_1) + W_2^2 VAR(R_2) + 2W_1 W_2 COV(R_1, R_2)\right]^{\frac{1}{2}} \tag{22.8}$$

where $W_1 + W_2 = 1$. The covariance term can be expressed in terms of the correlation coefficient between R_1 and R_2 and the respective standard deviations of R_1 and R_2:

$$\mathrm{COV}(R_1, R_2) = \rho_{12}\sigma_1\sigma_2 \tag{22.9}$$

The correlation coefficient measures the degree of similarity of movement for a time series. The ρ_{ij} is defined over the range of +1 to –1. Figure 22.3 depicts various values of ρ_{ij}. At $\rho_{ij} = 1$, the fluctuation of returns for both series over time is identical; at $\rho_{ij} = 0$, the returns are unrelated; and at $\rho_{ij} = -1$, the fluctuations are moving in opposite directions.

Equation 22.7 is a quadratic equation, and some value of W_1 minimizes σ_p. To obtain this value, we differentiate Eq. 22.8 with respect to W_1 and set this derivative equal to 0 to solve for W_1[1]:

$$W_1 = \frac{\sigma_2(\sigma_2 - \rho_{12}\sigma_1)}{\sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2} \tag{22.10}$$

Equation 22.10 can be used to estimate the optimal weight of security 1. From the constraint $W_1 + W_2 = 1$, the optimal weight of security 2 can also be estimated. From estimated W_1 and W_2, we can estimate $\overline{R}_p$ and σ_p for a two-security minimum-variance portfolio.

[1] In Appendix 22B, we have shown how to obtain optimal weights for a two-security portfolio. In addition, Appendix 22C discusses how the Sharpe performance measure approach can be used to derive the optimal portfolio weights.

22.3.3 The N-asset Case

An expanded form of Eqs. 22.8 for a portfolio of N securities is ρ_{ij}, the correlation coefficient between i and j, and n is the number of securities included in the portfolio.

$$\sigma_p = \sqrt{\sum_{i=1}^{n} W_i^2 \sigma_i^2 + \sum_{j=1}^{n} \sum_{i=1}^{n} W_i W_j \rho_{ij} \sigma_i \sigma_j} \quad i \neq j \tag{22.11}$$

where W_i and W_j are the investor's investment allocated to securities, i and j. ρ_{ij} is the correlation coefficient between i and j, and n is the number of securities included in the portfolio.

Since Eq. 22.11 has n securities, there are n variance terms (that is, $W_i^2\ \sigma_i^2$) and.

($n^2 - n$) covariance terms (that is $W_i W_j\ \rho_{ij}\ \sigma_i\ \sigma_j$). If $n = 20$, Eq. 22.11 will have 20 variance terms and 380 covariance terms.

Using a matrix notation, we can express these terms as follows:

$$\begin{bmatrix} W_1^2\sigma_1^2 & \cdots & \cdots & W_1 W_{20}\sigma_{1,20} \\ \vdots & W_2^2\sigma_2^2 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ W_{20} W_1 \sigma_{20,1} & \cdots & \cdots & W_{20}^2\sigma_{20}^2 \end{bmatrix}$$

This is a variance–covariance matrix for portfolio analysis.

22.3.4 The Efficient Portfolios

Under the mean–variance framework, by definition, a security or a portfolio will be efficient if.

1. No other portfolio has the same expected return at a lower risk
2. No other portfolio has a higher expected return at the same risk

These two rules are called the *dominance principle*, which we discussed in detail later in this chapter.

This suggests that given two investments, A and B, A is preferred to B if

$$\mathrm{E(A)} > \mathrm{E(B)}\ \text{and}\ \mathrm{var(A)} = \mathrm{var(B)}$$

Or

$$\mathrm{E(A)} = \mathrm{E(B)}\ \text{and}\ \mathrm{var(A)} < \mathrm{var(B)}.$$

where E(A) and E(B) are the expected returns of investments A and B, and var(A) and var(B) are their respective variances or risk. The mean return and standard deviation of an investment can be calculated and plotted as a single point on a mean-standard deviation diagram, as shown in Fig. 22.4.

All points below curve EF represent possible portfolio combinations. All points above EF are combinations of risk and returns that do not exist. Point D represents a portfolio of investments with return R_D and risk σ_D. Therefore, point B represents a risk and return combination that cannot be obtained with any combination of existing investment opportunities.

The EF curve is also called the efficient frontier, because all points below the curve are dominated by a point found on the curve. The efficient frontier is defined as the set of portfolios that has the maximum return for every given level of risk or the minimum risk for every level of return. Suppose a firm is willing to assume a maximum risk σ_D. It can obtain a return of R_D with portfolio D or move to point C on the frontier and receive a higher return of R_C with that portfolio at the same level of risk. Therefore, portfolio C dominates portfolio D because it is preferred to portfolio D.

A similar argument could be made in terms of risk. If the firm wants to achieve a return of R_A, it would select portfolio A over D, because A represents the same return at a lower level of risk or standard deviation: $\sigma_A < \sigma_D$. Therefore, point D is

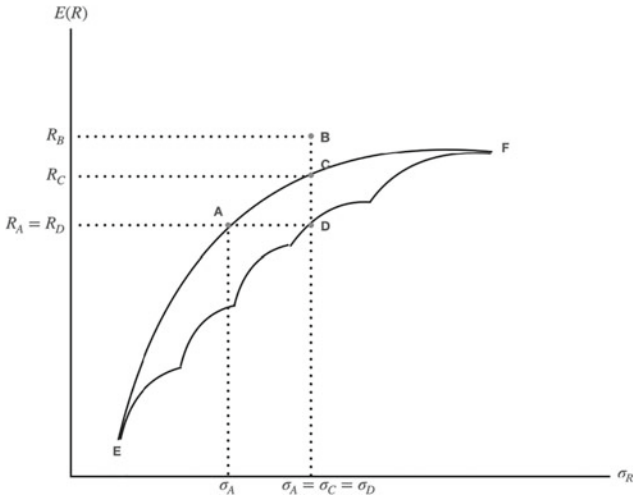


Fig. 22.4 Efficient frontier in portfolio analysis

not efficient but points *A* and *C* are. A decision-maker following these arguments would, therefore, select any point on the frontier as being preferable to any interior point.

The amount of curvature of the efficient frontier depends on the correlation between the two assets. If the correlation between assets equals +1, the combinations lie on the straight line. The lower the correlation coefficient, the more curved the efficient frontier.

Example 22.1 To show how the portfolio concepts and methods discussed in this section can be used in the practical analysis, we evaluate the monthly rates of return during January 2010–December 2018 for Johnson & Johnson (JNJ) and International Business Machines Corporation (IBM).

The basic statistical estimates for these two firms are average monthly rates of return and the variance–covariance matrix. The average monthly rates of return are −0.0011439 for IBM and −0.008967 for JNJ.

The variance and covariance are listed in Table 22.2.

From Eq. 22.10, we have

$$W_{IBM} = \frac{\sigma_{JNJ}^2 - cov(R_{IMB}, R_{JNJ})}{\sigma_{IBM}^2 + \sigma_{JNJ}^2 - cov(R_{IMB}, R_{JNJ})} = \frac{0.0016 - 0.0006}{0.0030 + 0.0016 - 0.0006} = 0.2512$$

$$W_{JNJ} = 1 - 0.2512 = 0.7488$$

Using the weight estimates and Eqs. 22.5 and 22.8,

$$E(R_P) = -0.007002$$

$$\sigma_P^2 = 0.000226$$

$$\rho_{1,2} = 0.2746$$

When ρ_{12} is less than 1, the combination of the two securities results in a total risk less than the sum of their weighted risk. This is the diversification effect. When ρ_{12} equals to 1, the combination of the two securities has no diversification effect at all.

Table. 22.2 Variance–covariance matrix

	IBM	*JNJ*
IBM	0.00298732	
JNJ	0.00060082	0.00160289

Table. 22.3 Mean returns and variance–covariance matrix

Mean returns		
	IBM	*JNJ*
	−0.0011439	−0.008967
Variance–covariance matrix		
	IBM	*JNJ*
IBM	0.00298732	
JNJ	0.00060082	0.00160289
Risk-free returns		
0.00266667		

For the correlation coefficient of $\rho_{12} = 0.2746$, we see that a portfolio that combines JNJ and IBM. shows a greater diversification effect and a substantial reduction in risk then would holding each of the securities separately.

Depending on the correlation coefficient between securities, as long as the securities are not positively correlated, the risk of a portfolio as measured by the standard deviation of return will be less than the weighted average of the risks (standard deviations) of the individual securities. In notational form,

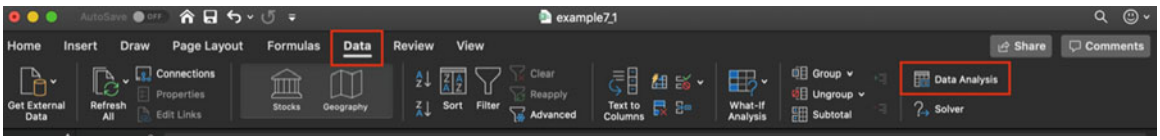
$$\sigma_P < W_1\sigma_1 + W_2\sigma_2 \text{ if } \rho_{12} \neq 1$$

This will always be true and represents a significant fact that investors and managers must keep in mind when making financial decisions.

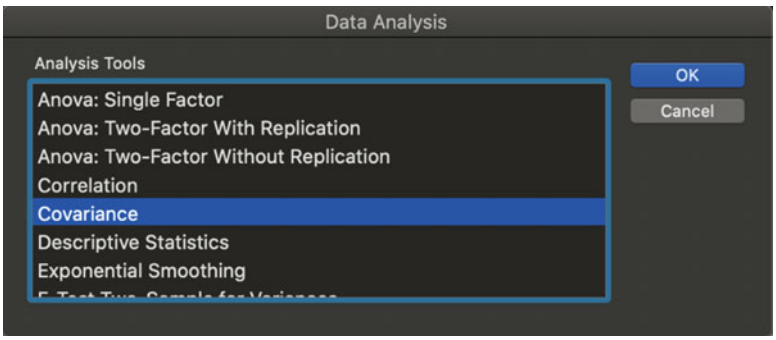
Solving Example 22.1 with Excel

According to the monthly returns, we calculate in Excel the monthly average returns of both firms and the variance–covariance matrix of the two series of returns as presented in Table 22.3. Note that here we assume the annual risk-free rate is 3.2%, and then the monthly risk-free return is therefore 0.00266667.

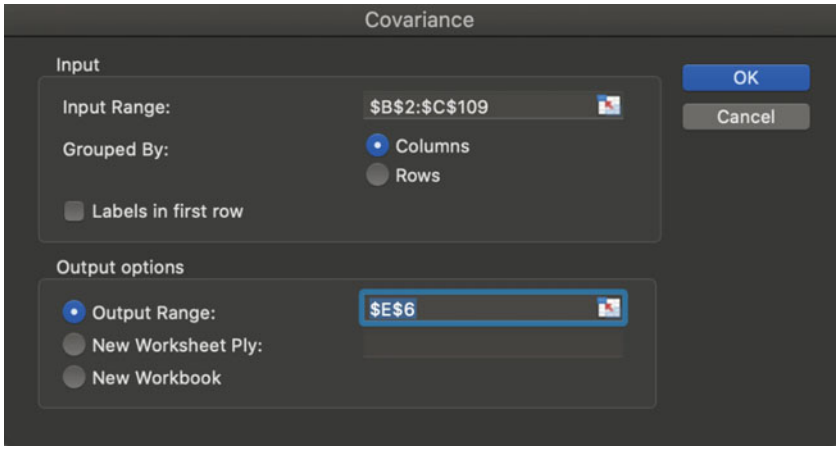
To obtain the variance–covariance matrix, we can use the incorporated function in Excel. Go to the "data" tab, and click the "data analysis". (see below.)



Then choose the option "covariance" as shown below.



For the input, chose the columns of returns. That is, the two columns of monthly returns we have for IBM and JNJ. Then for the output range, you may choose any cell in the sheet where you would like the output matrix to be shown.



Once we obtain the variance and the covariance, we are ready to compute the optimal weights of each stock based on the principle of variance minimization (See the screenshot below.)

From this table, we found the optimal weight of Johnson & Johnson (W_A) is 0.748815, and then, the weight of the other stock is therefore one minus the weight of stock A.

22.3.5 Corporate Application of Diversification

The effects of diversification are not necessarily isolated to security analysis and portfolio management but may have wider applications at the corporate decision-making level. Managers often justify undertaking multiproduct lines because of the effects of diversification. Instead of putting "all eggs in one basket," the investment risks are spread out among many lines of services of products in hopes of reducing overall risk and maximizing returns. To what degree this diversification takes place in other types of corporate decisions depends on the decision-maker's preference for risk and return. The overall goal is to reduce business risk fluctuations in net income.

This type of corporate diversification can also be taken to the multinational level. For example, General Motors has overseas divisions throughout the world that all produce the same product—autos and auto parts. Because GM is a multinational corporation, it can take advantage of diversifying effects of different exchange rates and political and economic climates, because each overseas division is less than perfectly positively correlated with each other.

The strategy of diversifying at the corporate level has been criticized because shareholders can attain diversification in their own portfolios, which makes corporate diversification redundant. There is the agency theory argument that corporate diversification is in fact a form of risk-sharing between stockholders and bondholders, where the bondholders have less risky fixed-income securities due to corporate diversification and the shareholders suffer a loss in the value of their equity position. Finally, there is a reduction in the risk/return deals with options, which deals with mergers and divestitures.

22.4 Determination of Commercial Lending Rates

In this section, we explain the process of estimating a lending rate that a financial institution would extend to a firm, or a borrowing rate a firm would feel is reasonable. The loan officers of the bank and the financial analysts of the firm need to consider the firm's business, financial, and total risks to make a fair analysis of the cost of borrowing for the firm or the rate the bank should charge when lending money to the firm.

The lending rate is based in part on the risk-free rate. Therefore, we must forecast the risk-free rate (R_f) for boom, normal, and recession economies.

The second component of the lending rate is the risk premium (R_p), which is calculated individually for each firm by examining the change in EBIT under the three economic conditions. The EBIT is an indicator of the lender as to the ability of the potential borrower to repay borrowed funds.

Table 22.4 Input information

Economic condition	R_t	(A) Probability	(B) EBIT	(C) Probability	(D) R_p
Boom	10%	0.25	$2.5 m 1.5 0.5	0.40 0.30 0.30	2% 3 5
Normal	9	0.50	$2.5 m 1.5 0.5	0.40 0.30 0.30	2 3 5
Poor	8	0.25	$2.5 m 1.5 0.5	0.40 0.30 0.30	2 3 5

Table 22.5 Calculation table

Economic condition	(A) R_t	(B) Probability	(C) R_p	(D) Probability	(B X D) Probability of occurrence	(A + C) Lending rate
Boom	10%	0.25	2% 3 5	0.40 0.30 0.30	0.100 0.075 0.075	12% 13 15
Normal	9	0.50	2 3 5	0.40 0.30 0.30	0.200 0.150 0.150	11 12 14
Poor	8	0.25	2 3 5	0.40 0.30 0.30	0.100 0.075 0.075 1.000	10 11 13

Tables 22.4 and 22.5 provide the information required for the analysis. In Table 22.4, column B lists probabilities relating to economic conditions for the entire economy, and column D lists probabilities of various levels of EBIT for the firm.

There are nine possible lending rates for the three economic conditions. The probabilities of these rates for the three economic conditions are elaborated in Table 22.5.

These tables show that during a boom the risk-free rate is 10%, but the risk premium can take on different values. There is a 40% chance that the risk premium will be 2%, a 30% chance it will be 3%, and a 30% chance it will be 5%. By multiplying the two probabilities, R_f and R_p, we get the joint probability of occurrence. Therefore, there is a 10% chance of receiving a 13% or 15% rate. This process also applies for normal and recession conditions.

For the problem set up in Table 22.5, the weighted average lending rate is

$$\begin{aligned}\overline{R} &= (0.100)(12\%) + (0.75)(13) + (0.075)15 + (0.200)11 + (0.150)(12) \\ &\quad + (0.150)(14) + (0.100)10 + (0.074)(13\%) \\ &= 12.2\%\end{aligned}$$

with a standard deviation of

$$\begin{aligned}\sigma = &\Big[(0.100)(12 - 12.2)^2 + (0.075)(13 - 12.2)^2 + (0.075)(15 - 12)^2 \\ &+ (0.200)(11 - 12.2)^2 + (0.150)(12 - 12.2)^2 + (0.150)(14 - 12.2)^2 \\ &+ (0.100)(10 - 12.2)^2 + (0.075)(11 - 12.2)^2 + (0.075)(13 - 12.2)^2\Big]^{0.5} \\ = &\,(0.004 + 0.048 + 0.588 + 0.288 + 0.006 + 0.436 + 0.484 + 0.108 + 0.048)^{0.5} \\ = &\,1.42\%\end{aligned}$$

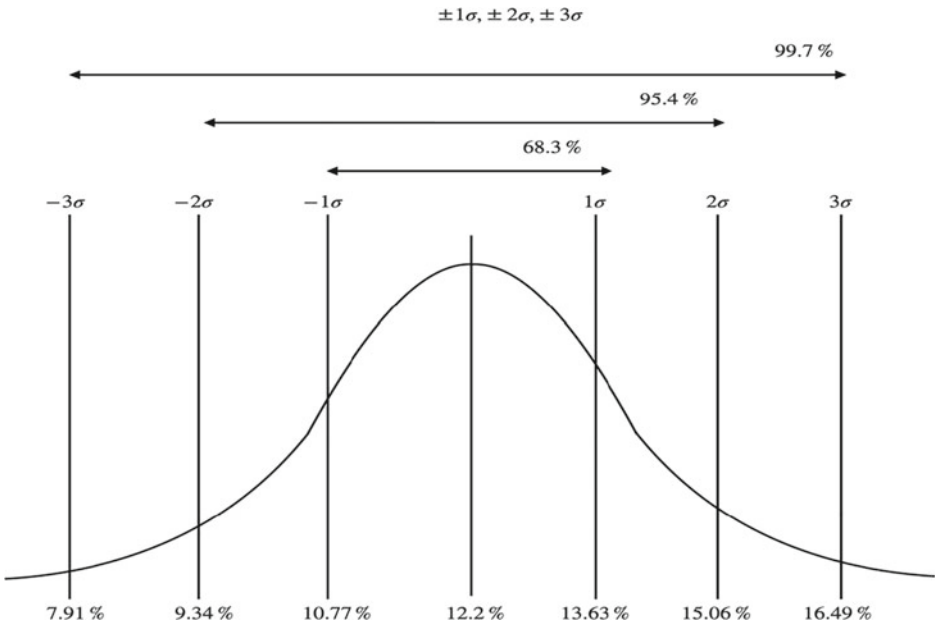


Fig. 22.5 Probability distribution of lending rate

If the lending rate distribution is normal (as discussed in Appendix 22A). Given the mean and standard deviation for such a distribution, in Fig. 22.5 we see that 68.3% of the observations of a normal distribution are within a standard deviation of 1 of the mean, 95.4% are within a standard deviation of 2, and 99.7% within a standard deviation of 3.

Based on the mean and standard deviation of the estimated lending rate, we depict the expected lending rate and its standard deviation in Fig. 22.5. The percentages in Fig. 22.5, along with the mean and standard deviations, illustrate the normal distribution. The average lending rate is normally distributed with a mean of 12.2% and a standard deviation of +1.43%. This implies that almost all (99.7%) of the rates lie in the range from 7.91% to 16.49%. We also know 68.3% of the rates lie in the range of 10.77% to 13.63%.

This example is based on expected values for determining required rates of return. From a financial theory view point, equilibrium models such as capital asset pricing model (CAPM) and option pricing model (OPM) are superior for determining required rates of return.

22.5 The Dominance Principle and Performance EVALUATION

The *dominance principle* can be used to understand the risk/return tradeoff. When making financial decisions, we are dealing with two-dimensional analysis—risk and return. The dominance principle allows us to hold one variable constant while we make comparisons based on the other variable. However, in some cases this may not be sufficient. When dealing simultaneously with different levels of risk and return, we need to use the method of *performance evaluation*.

As with efficient frontier analysis, we must assume an investor prefers returns and dislikes risks. For example, as depicted in Fig. 22.6, if prepared to experience the risk associated with σ_A, an investor can obtain a higher expected return with portfolio $A\left(\overline{R}_A\right)$ than with portfolio $B\left(\overline{R}_B\right)$. Thus, A dominates B and would be preferred. Similarly, if satisfied with a return of $\overline{R}_B$, an investor would select portfolio B over C because the risks associated with B are less: $\sigma_B < \sigma_C$. Points B and C were directly comparable because of a common return, $R_B = R_C$. How about portfolio D? How does its risk versus return compare with other portfolios in the figure, especially B? Portfolio D is not directly comparable using the dominance principle, and this is a basic limitation: portfolios without a common risk or return factor are not directly comparable. The problem can be solved using a technique called *stochastic dominance*, which is a subject covered in advanced corporate finance or investment courses. Stochastic dominance allows for joint consideration of the mean and variance.[2]

A simplified approach for comparing portfolios A and D is to use a performance measure. Basically, there are three portfolio performance measures: the Sharpe, Treynor, and Jensen methods. Here, we elaborate on the Sharpe measure, which

[2] Complete coverage of this topic can be found in a text by Levy and Sarnat (1984).

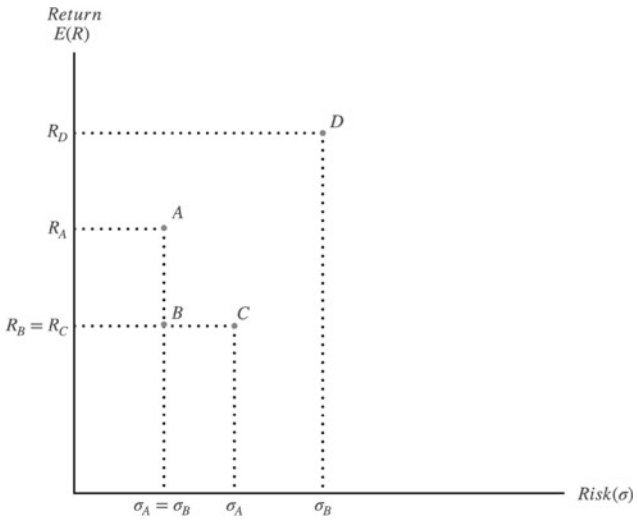


Fig. 22.6 Dominance principle

is the most basic of the three. Using this method, we can compare the relative risk–return tradeoff performance of portfolios B and D as follows:

$$SP^D = \frac{\overline{R}_D - R_f}{\sigma_D}$$
$$SP^B = \frac{\overline{R}_B - R_f}{\sigma_B}$$

where SP^D and SP^B = Sharpe performance measures; $\overline{R}_D$ and $\overline{R}_B$ = average return of each respective portfolio; R_f = average risk-free rate; and σ_D and σ_B = respective standard deviation of returns of each portfolio.

Because the numerator is the average return reduced by the risk-free rate, it represents the average risk premium of each portfolio. Dividing the risk premium by the total risk per portfolio results in a measure of the unit return (premium) per unit risk for each portfolio. The Sharpe performance measure equation would, therefore, allow direct comparison of any portfolio given its risks and returns.

Example 22.3 An insurance firm is considering two investment funds. From past performance, they calculate the average returns and variables for both funds, as listed in Table 22.6. The T-bill rate is 9.5%, which the firm uses as a measure of the risk-free rate.

Using this information, the Sharpe performance measure is calculated as

$$SP_S = \frac{0.18 - 0.095}{0.2} = 0.425$$
$$SP_J = \frac{0.16 - 0.095}{0.15} = 0.433$$

From these calculations, we see that the Jones fund will have a slightly better performance and thus would be the better alternative of the two investments.

Table 22.6 Information for two mutual funds

	Smith fund	Jones fund
Average return($\overline{R}$)	18%	16%
Standard deviation (σ)	20%	15%
Risk-free rate = R_f = 9.5%		

In Appendix 22D, we use the monthly data of Johnson & Johnson, IBM, Apple, Coca-Cola, Caterpillar, and S&P 500 from January 1, 2016 to March 1, 2021 to calculate the Sharpe performance measures for these five companies and S&P 500 index. In addition, the optimal weights in terms of two securities portfolio is also calculated. In the next chapter, we will use this set of data to show how optimal weights in terms of three-security portfolios will be demonstrated.

22.6 Summary

In this chapter, we defined the basic concepts of risk and risk measurement. Based on the relationship of risk and return, we demonstrated the efficient portfolio concept and its implementation, as well as the dominance principle and performance measures. Interest rates and market rates of return were used as measurements to show how the commercial lending rate and the market risk premium are calculated. Following a surveyed paper by Graham and Harvey (2001), we discussed in this chapter that management evaluates firm risk, rather than project risk, when evaluating new investment opportunities. Also, firms focus on financial flexibility and credit ratings when issuing debt and earnings-per-share dilution and recent stock price appreciation when issuing equity.

Appendix 22A: The Normal Distribution

In this chapter, we introduced the idea of thinking about the uncertainty involved in future corporate returns on assets by regarding such returns not as fixed numbers but rather as random variables. Specifically, we briefly discussed the possibility of using the normal distribution for this purpose. The normal distribution is generally described by its probability density function. If the random variable X follows such a distribution function, the probability density function is

$$f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean, or expected value, of X, and σ^2 its variance. Here, $\pi = 3.14158$ and $e = 2.71828\ldots$ are physical constants.

Fig. 22.7 shows a graph of this probability density function, exhibiting a characteristic "bell shape" centered on the mean. Probability density functions are constructed in such a way that the area under the curve, between any two possible values of the random variable, measures the probability that the random variable will take a value in the corresponding range, illustrated in Fig. 22.7.

The mean of a normal distribution determines the center of the distribution but does not influence its dispersion. Therefore, two normally distributed random variables with the same variance but different means will have identical probability density functions, except that one will be shifted to the right of the other. This is shown graphically in Fig. 22.8.

The variance of a normal distribution determines the spread, or dispersion, about the mean, but does not affect the center of the distribution. The higher the variance, the greater is the dispersion. Figure 22.9 shows the probability density functions of two normal distributions with identical means but different variances.

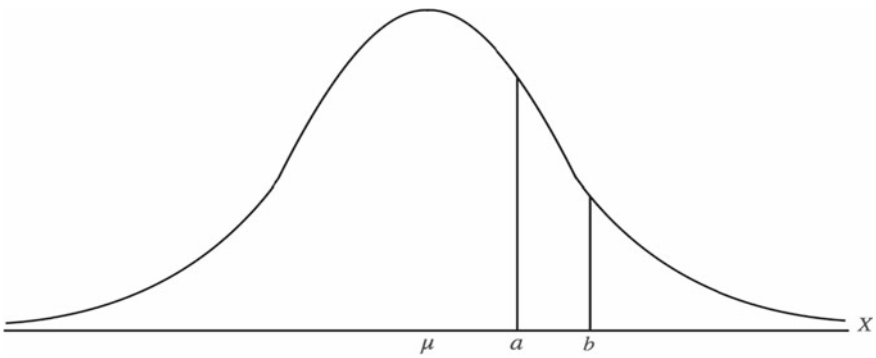


Fig. 22.7 Probability density function for a normal distribution, showing the probability that a normal random variable lies between a and b (shaded area)

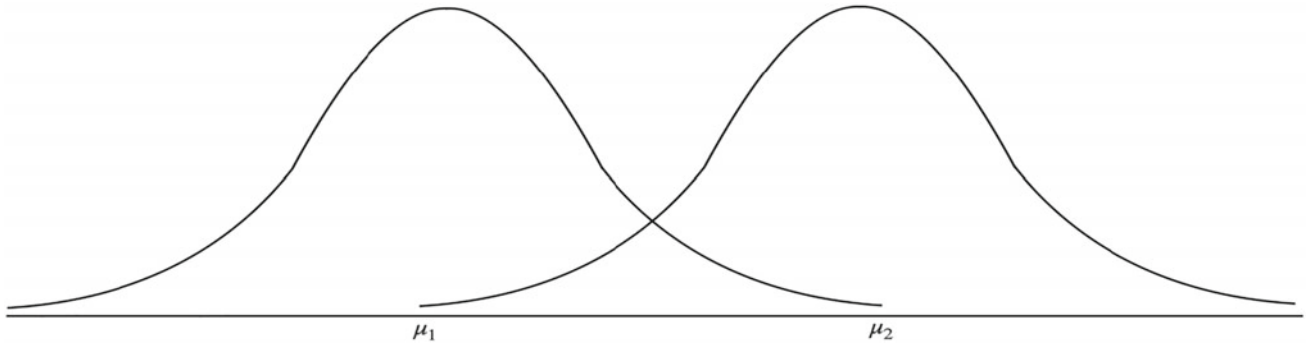


Fig. 22.8 Probability density function of normal random variables with equal variances: mean μ_2 is Greater Than μ_1

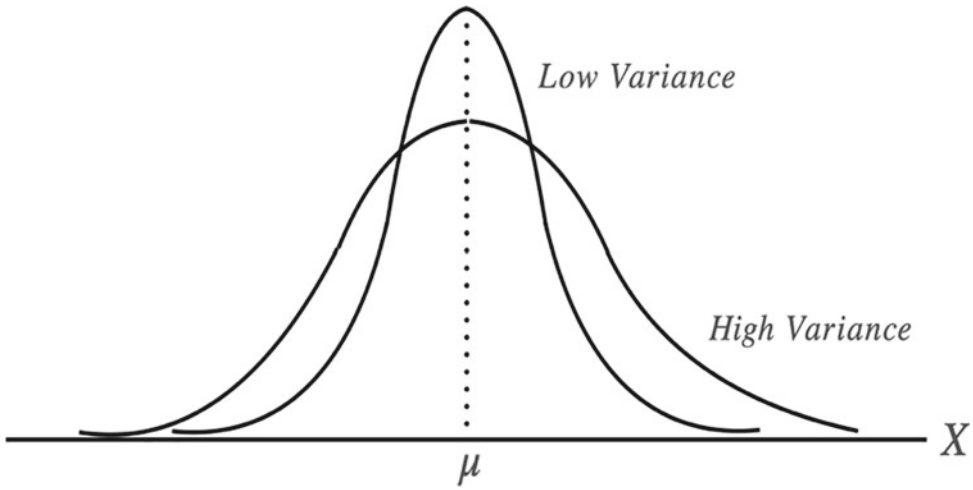


Fig. 22.9 Probability density functions of normal distributions with equal means and different variances

It follows that the greater the variance, the larger the probability that a normally distributed random variable differs from its mean by any fixed amount. For this reason, we associate variance with risk when considering the distribution of possible rates of return.

In fact, using the normal distribution table listed at the end of the book, it is possible to calculate the probability that a normally distributed random variable lies in any specified range. For convenience, we tabulate some useful values in Table 22.7. The table shows the probability that a random variable lies between $\mu - K$ and $\mu + K$ for particular values of K/σ, where the standard deviation is σ, as depicted by the shaded area in Fig. 22.10.

Table 22.7 Probability, *P*, that a normal random variable with mean μ and standard deviation σ lies between $K - \sigma$ and $K - \sigma$

P	K/ σ
0.50	0.674
0.60	0.842
0.70	1.036
0.80	1.281
0.90	1.645
0.95	1.960

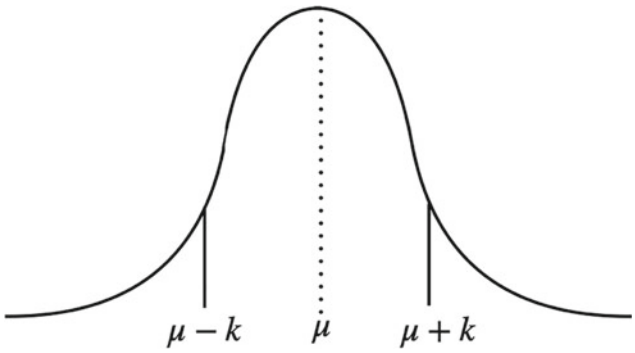


Fig. 22.10 Probability that a normal random variable is between μ-K and μ + K (Shaded Area)

Example 22A-1

An investor is interested in the prospects of a particular corporation. The investor expects the corporation to achieve a rate of return on assets of 12% next year and believes there is a 50% probability. If this uncertainty can be expressed as a normal distribution, find the mean and variance of that distribution.

The mean is the expected return, so that we have

$$\mu = 12$$

According to this investor, the probability is 0.50 that the actual return will be between 10.5 and 13.5%, so that, in our notation,

$$K = \frac{13.5 - 10.5}{2} = 1.5$$

From Table 22.7, we have

$$\frac{K}{\sigma} = 0.674$$

so that

$$\sigma = \frac{K}{0.674} = \frac{1.5}{0.674} = 2.2255$$

Finally, the variance is the square of the standard deviation, hence

$$\sigma^2 = (2.2255)^2 = 4.95$$

This is illustrated in Fig. 22.11.

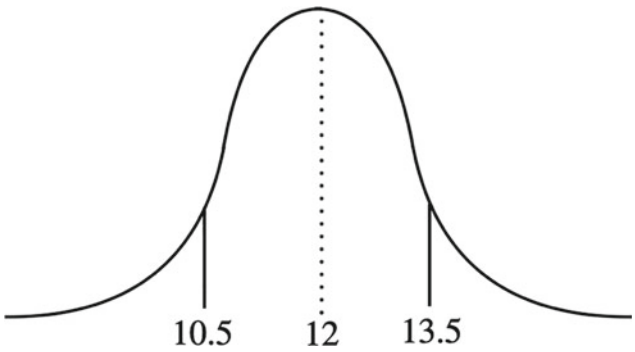


Fig. 22.11 For a normal random variable with mean 12, standard deviation 4.95, the probability is 0.5 of a value between 10.5 and 13.5

You might want to extend these calculations to demonstrate that variance does indeed provide a useful measure of uncertainty. Suppose our investor is less uncertain about future prospects. This could be reflected in one of two ways:

A. The investor may believe there is a 70% chance that next year's return on assets will lie between 10.5 and 13.5%.
B. The investor may believe there is a 50% chance that next year's return on assets will be between 11 and 13%.

It is left as an exercise to the reader to verify that, in either case, assuming a normal distribution, this uncertainty will be described by a distribution with a mean of 12 and a variance of less than 4.95.

One important property of a normal distribution is that any linear function of a normal random variable also has a normal distribution. Specifically, suppose that the random variable X has a normal distribution with mean μ and variance σ^2, and we define the random variable Y as

$$\mathrm{Y} = \mathrm{c} + \mathrm{dX}$$

where c and d are any constant fixed numbers. Then it can be shown that Y has a normal distribution with mean (c + d μ) and variance $\mathrm{d}^2\ \sigma^2$.

Appendix 22B: Minimum-Variance Approach to Derive Optimal Weight

If there is a two-security portfolio, its variance can be defined as

$$\sigma_p^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2Cov(r_1, r_2) \tag{22.12}$$

where w_1 and w_2 are weight associated with security 1 and 2, respectively.

σ_1^2 and σ_2^2 are variance of security 1 and 2, respectively.

The problem is choosing optimal w_1 to minimize the portfolio variance, $\sigma_p{}^2$

$$\underset{w_1}{Min}\ .\sigma_P^2 \tag{22.13}$$

We can solve the minimization problem by differentiating the $\sigma_p{}^2$ with respect to w_1, and setting the derivative equal to zero, i.e., we want to solve

$$\frac{\partial\sigma_p^2}{\partial w_1} = 0 \tag{22.14}$$

Since $w_1 + w_2 = 1$ or $w_2 = 1 - w_1$.

Therefore, the variance, $\sigma_p{}^2$, can be rewritten as

$$\begin{aligned}\sigma_p^2 &= w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2Cov(r_1, r_2)\\ &= w_1^2\sigma_1^2 + (1 - w_1)^2\sigma_2^2 + 2w_1(1 - w_1)Cov(r_1, r_2)\\ &= w_1^2\sigma_1^2 + \sigma_2^2 - 2w_1\sigma_2^2 + w_1^2\sigma_2^2 + 2w_1Cov(r_1, r_2) - 2w_1^2Cov(r_1, r_2)\end{aligned}$$

Now, the first-order conditions of equation (E3) can be written as

$$2w_1\sigma_1^2 - 2\sigma_2^2 + 2w_1\sigma_2^2 + 2Cov(r_1, r_2) - 4w_1Cov(r_1, r_2) = 0$$

Rearrange the above equation

$$w_1\sigma_1^2 + w_1\sigma_2^2 - 2w_1Cov(r_1, r_2) = \sigma_2^2 - Cov(r_1, r_2)$$

$$[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1, r_2)]w_1 = \sigma_2^2 - Cov(r_1, r_2)$$

Finally, we have

$$w_1 = \frac{\sigma_2^2 - Cov(r_1, r_2)}{\sigma_1^2 + \sigma_2^2 - 2Cov(r_1, r_2)}$$

Appendix 22C: Sharpe Performance Approach to Derive Optimal Weight

In Appendix 22B, we derived the optimal portfolio weight in terms of the minimum-variance approach. In this case, we assume that the investor is only concerned with the risk without considering the return. If the investor is concerned with both return and risk, then the Sharpe performance measure approach can be used to derive the optimal weights, which will be discussed in this appendix.

Solution for the weights of the optimal risky portfolio includes two securities that can be found by solving the following maximization problem:

$$\underset{w_1}{Max}\, S_p = \frac{E(r_p) - r_f}{\sigma_p}$$

where $E(r_p)$ = expected rates of return for portfolio P.

r_f = risk-free rates of return.

S_p = Sharpe performance measure.

σ_p as defined in equation (22.12) of Appendix 22B.

We can solve the maximization problem by differentiating the S_p with respect to w_1, and setting the derivative equal to zero, i.e., we want to solve

$$\frac{\partial S_p}{\partial w_1} = 0 \tag{22.15}$$

In the case of two securities, we know that

$$E(r_p) = w_1 E(r_1) + w_2 E(r_2) \tag{22.16}$$

$$\sigma_p = [w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1 w_2 Cov(r_1, r_2)]^{1/2} \tag{22.17}$$

$$w_1 + w_2 = 1 \tag{22.18}$$

From the above Eqs. (22.16), (22.17), and (22.18), we can rewrite $E(r_p) - r_f$ and σ_p as

$$E(r_p) - r_f = w_1 E(r_1) + w_2 E(r_2) - r_f = w_1 E(r_1) + (1 - w_1) E(r_2) - r_f \equiv f(w_1) \tag{22.19}$$

$$\begin{aligned} \sigma_p &= [w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1 w_2 Cov(r_1, r_2)]^{1/2} \\ &= [w_1^2\sigma_1^2 + (1 - w_1)^2\sigma_2^2 + 2w_1(1 - w_1) Cov(r_1, r_2)]^{1/2} \equiv g(w_2) \end{aligned} \tag{22.20}$$

Equation (22.15) becomes

$$\frac{\partial S_p}{\partial w_1} = \frac{\partial \frac{f(w_1)}{g(w_2)}}{\partial w_2} = \frac{f'(w_1)g(w_2) - f(w_1)g'(w_2)}{[g(w_2)]^2} = 0 \tag{22.21}$$

where

$$f'(w_1) = \frac{\partial f(w_1)}{\partial w_1} = E(r_1) - E(r_2) \tag{22.22}$$

$$\begin{aligned} g'(w_1) &= \frac{\partial g(w_1)}{\partial w_1} = \frac{1}{2} \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)]^{\frac{1}{2}-1} \\ &\quad \times [2w_1\sigma_1^2 + 2w_1\sigma_2^2 - 2\sigma_2^2 + 2Cov(r_1,r_2) - 4w_1Cov(r_1,r_2)] \\ &= [w_1\sigma_1^2 + w_1\sigma_2^2 - \sigma_2^2 + Cov(r_1,r_2) - 2w_1Cov(r_1,r_2)] \\ &\quad \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)]^{1/2} \end{aligned} \tag{22.23}$$

From Eq. (22.21),
$f'(w_1)g(w_1) - f(w_1)g'(w_1) = 0$, or

$$f'(w_1)g(w_1) = f(w_1)g'(w_1) \tag{22.24}$$

Now, plugging $f(w_1)$, $g(w_1)$, $f'(w_1)$, and $g'(w_1)$ [Eqs. (22.19), (22.20), (22.21), and (22.22)] into Eq. (22.23), we have

$$\begin{aligned} &[E(r_1) - E(r_2)] \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)]^{1/2} \\ &= [w_1E(r_1) + (1-w_1)E(r_2) - r_f] \times [w_1\sigma_1^2 + w_1\sigma_2^2 - \sigma_2^2 + Cov(r_1,r_2) - 2w_1Cov(r_1,r_2)] \\ &\quad \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)]^{-1/2} \end{aligned} \tag{22.25}$$

Multiplying by $[w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)]^{-1/2}$ on both hand sides of Eq. (22.24), we have

$$\begin{aligned} &[E(r_1) - E(r_2)] \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)] \\ &= [w_1E(r_1) + (1-w_1)E(r_2) - r_f] \times [w_1\sigma_1^2 + w_1\sigma_2^2 - \sigma_2^2 + Cov(r_1,r_2) - 2w_1Cov(r_1,r_2)] \end{aligned} \tag{22.26}$$

Rearrange all terms on both hand sides of Eq. (22.25), i.e.,
Left-hand side of Eq. (22.25):

$$\begin{aligned} &[E(r_1) - E(r_2)] \times [w_1^2\sigma_1^2 + (1-w_1)^2\sigma_2^2 + 2w_1(1-w_1)Cov(r_1,r_2)] \\ &= [E(r_1) - E(r_2)] \times [w_1^2\sigma_1^2 + \sigma_2^2 - 2w_1\sigma_2^2 + w_1^2\sigma_2^2 + 2w_1Cov(r_1,r_2) - 2w_1^2Cov(r_1,r_2)] \\ &= [E(r_1) - E(r_2)] \times \{w_1^2[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)] + 2w_1[Cov(r_1,r_2) - \sigma_2^2] + \sigma_2^2\} \\ &= [E(r_1) - E(r_2)] \times \{w_1^2[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)]\} \\ &\quad + [E(r_1) - E(r_2)] \times \{2w_1[Cov(r_1,r_2) - \sigma_2^2]\} \\ &\quad + [E(r_1) - E(r_2)] \times \sigma_2^2[E(r_1) - E(r_2)] \times [\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)]w_1^2 \\ &\quad + 2[E(r_1) - E(r_2)] \times [Cov(r_1,r_2) - \sigma_2^2]w_1 + [E(r_1) - E(r_2)] \times \sigma_2^2 \end{aligned}$$

Right-hand side of Eq. (22.25)

$$\begin{aligned} &[w_1E(r_1) + (1-w_1)E(r_2) - r_f] \times [w_1\sigma_1^2 + w_1\sigma_2^2 - \sigma_2^2 + Cov(r_1,r_2) - 2w_1Cov(r_1,r_2)] \\ &= [w_1E(r_1) + E(r_2) - w_1E(r_2) - r_f] \times [w_1\sigma_1^2 + w_1\sigma_2^2 - 2w_1\mathrm{Cov}(r_1,r_2) - \sigma_2^2 + \mathrm{Cov}(r_1,r_2)] \\ &= \{w_1[E(r_1) - E(r_2)] + [E(r_2) - r_f]\} \times \{w_1[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)] + Cov(r_1,r_2) - \sigma_2^2\} \\ &= w_1[E(r_1) - E(r_2)] \times w_1[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)] + w_1[E(r_1) - E(r_2)] \times [Cov(r_1,r_2) - \sigma_2^2] \\ &\quad + [E(r_2) - r_f] \times w_1[\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)] + [E(r_2) - r_f] \times [Cov(r_1,r_2) - \sigma_2^2] \\ &= [E(r_1) - E(r_2)] \times [\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)]w_1^2 \\ &\quad + [E(r_1) - E(r_2)] \times [Cov(r_1,r_2) - \sigma_2^2]w_1 \\ &\quad + [E(r_2) - r_f] \times [\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)]w_1 \\ &\quad + [E(r_2) - r_f] \times [Cov(r_1,r_2) - \sigma_2^2] \end{aligned}$$

Subtracting $[E(r_1) - E(r_2)][\sigma_1^2 + \sigma_2^2 - 2Cov(r_1,r_2)]w_1^2$ and $[E(r_1) - E(r_2)][Cov(r_1,r_2) - \sigma_2]w_1$ from both hand sides of Eq. (22.25), we have

$$\begin{aligned}&[E(r_1)-E(r_2)]\times[Cov(r_1,r_2)-\sigma_2^2]w_1+[E(r_1)-E(r_2)]\times\sigma_2^2\\&=[E(r_2)-r_f]\times[\sigma_1^2+\sigma_2^2-2Cov(r_1,r_2)]w_1+[E(r_2)-r_f]\times[Cov(r_1,r_2)-\sigma_2^2]\end{aligned}\tag{22.27}$$

Moving all the terms with w_1 on one side and leaving the rest terms on the other side from Eq. (22.26), we have

$$\begin{aligned}&[E(r_1)-E(r_2)]\times\sigma_2^2-[E(r_2)-r_f]\times[Cov(r_1,r_2)-\sigma_2^2]\\&=[E(r_2)-r_f]\times[\sigma_1^2+\sigma_2^2-2Cov(r_1,r_2)]w_1-[E(r_1)-E(r_2)]\times[Cov(r_1,r_2)-\sigma_2^2]w_1\end{aligned}\tag{22.28}$$

Rearrange Eq. (22.27) in order to solve for w_1, i.e.,

$$\begin{aligned}&[E(r_1)-E(r_2)+E(r_2)-r_f]\times\sigma_2^2-[E(r_2)-r_f]Cov(r_1,r_2)\\&=\left\{[E(r_2)-r_f]\sigma_1^2+[E(r_2)-r_f]\sigma_2^2-[E(r_2)-r_f][2Cov(r_1,r_2)]-[E(r_D)-E(r_E)]Cov(r_1,r_2)+[E(r_1)-E(r_2)]\sigma_2^2\right\}w_1\\&=\left\{[E(r_1)-r_f]\sigma_2^2+[E(r_2)-r_f]\sigma_1^2-[E(r_1)-r_f+E(r_2)-r_f]Cov(r_1,r_2)]\right\}w_1\end{aligned}$$

Finally, we have the optimum weight of security 1 as

$$w_1=\frac{[E(r_1)-r_f]\sigma_2^2-[E(r_2)-r_f]Cov(r_1,r_2)}{[E(r_1)-r_f]\sigma_2^2+[E(r_2)-r_f]\sigma_1^2-[E(r_1)-r_f+E(r_2)-r_f]Cov(r_1,r_2)}$$

In this appendix, we discussed the Sharpe Performance Measure approach in terms of two securities cases. This kind of method can be extended to n securities case. Chiou et al. (2008) have used this approach to investigate whether international diversification is beneficial to investors. In addition, Lee et al. (2013) have shown this approach in detail.

Appendix 22D: Beta Coefficients, Sharpe Performance Measures, and Optimal Portfolio Weights for Five Companies

Step 1:

Collect monthly adjusted closed price for Johnson & Johnson (JNJ), IBM, Apple (AAPL), Coca-cola (KO), Caterpillar (CAT), and S&P 500 (GSPC) from Yahoo finance historical data. There are 63 months from January 01, 2016 to March 01, 2021 in our sample.

Step 2:

Calculate monthly real return by using the formula: $\frac{Adj.Closed_{t+1}-Adj.Closed_t}{Adj.Closed_t}$.

Date	JNJ monthly Returns	IBM monthly returns	Apple monthly returns	Coca-Cola monthly returns	Caterpillar monthly returns	GSPC monthly Returns
1/1/2016	0.0074	0.0500	−0.0067	0.0049	0.1013	−0.0041
2/1/2016	0.0359	0.1676	0.1333	0.0756	0.1306	0.0660
3/1/2016	0.0359	−0.0364	−0.1399	−0.0267	0.0154	0.0027
4/1/2016	0.0054	0.0534	0.0653	−0.0045	−0.0579	0.0153
5/1/2016	0.0841	−0.0032	−0.0368	0.0164	0.0455	0.0009
6/1/2016	0.0324	0.0582	0.0901	−0.0301	0.0917	0.0356
7/1/2016	−0.0470	−0.0108	0.0181	−0.0046	−0.0002	−0.0012
8/1/2016	−0.0035	0.0084	0.0713	−0.0256	0.0832	−0.0012
9/1/2016	−0.0181	−0.0325	0.0043	0.0101	−0.0598	−0.0194
10/1/2016	−0.0404	0.0555	−0.0266	−0.0483	0.1551	0.0342
11/1/2016	0.0423	0.0325	0.0533	0.0362	−0.0295	0.0182

(continued)

Date	JNJ monthly Returns	IBM monthly returns	Apple monthly returns	Coca-Cola monthly returns	Caterpillar monthly returns	GSPC monthly Returns
12/1/2016	−0.0170	0.0514	0.0477	0.0027	0.0315	0.0179
1/1/2017	0.0791	0.0304	0.1289	0.0094	0.0188	0.0372
2/1/2017	0.0259	−0.0239	0.0532	0.0114	−0.0403	−0.0004
3/1/2017	−0.0087	−0.0795	−0.0001	0.0257	0.1024	0.0091
4/1/2017	0.0387	−0.0478	0.0634	0.0538	0.0396	0.0116
5/1/2017	0.0383	0.0177	−0.0533	−0.0136	0.0193	0.0048
6/1/2017	0.0033	−0.0595	0.0327	0.0305	0.0604	0.0193
7/1/2017	−0.0026	−0.0113	0.1027	−0.0063	0.0385	0.0005
8/1/2017	−0.0116	0.0250	−0.0566	−0.0119	0.0615	0.0193
9/1/2017	0.0723	0.0619	0.0968	0.0297	0.0889	0.0222
10/1/2017	−0.0006	−0.0006	0.0166	−0.0046	0.0456	0.0281
11/1/2017	0.0089	0.0064	−0.0117	0.0106	0.1164	0.0098
12/1/2017	−0.0110	0.0670	−0.0106	0.0373	0.0330	0.0562
1/1/2018	−0.0601	−0.0481	0.0638	−0.0918	−0.0457	−0.0389
2/1/2018	−0.0070	−0.0057	−0.0542	0.0049	−0.0469	−0.0269
3/1/2018	−0.0130	−0.0552	−0.0150	0.0037	−0.0205	0.0027
4/1/2018	−0.0543	−0.0252	0.1308	−0.0049	0.0576	0.0216
5/1/2018	0.0219	−0.0004	−0.0056	0.0200	−0.1069	0.0048
6/1/2018	0.0921	0.0374	0.0280	0.0726	0.0599	0.0360
7/1/2018	0.0164	0.0107	0.1962	−0.0442	−0.0285	0.0303
8/1/2018	0.0327	0.0434	−0.0048	0.0363	0.0982	0.0043
9/1/2018	0.0132	−0.2366	−0.0305	0.0454	−0.2044	−0.0694
10/1/2018	0.0494	0.0766	−0.1840	0.0526	0.1254	0.0179
11/1/2018	−0.1159	−0.0736	−0.1136	−0.0531	−0.0634	−0.0918
12/1/2018	0.0312	0.1825	0.0552	0.0165	0.0479	0.0787
1/1/2019	0.0268	0.0276	0.0403	−0.0580	0.0380	0.0297
2/1/2019	0.0298	0.0334	0.1017	0.0335	−0.0135	0.0179
3/1/2019	0.0101	−0.0059	0.0564	0.0561	0.0290	0.0393
4/1/2019	−0.0712	−0.0947	−0.1276	0.0014	−0.1354	−0.0658
5/1/2019	0.0693	0.0988	0.1349	0.0364	0.1376	0.0689
6/1/2019	−0.0650	0.0750	0.0764	0.0417	−0.0339	0.0131
7/1/2019	−0.0143	−0.0857	−0.0202	0.0458	−0.0893	−0.0181
8/1/2019	0.0155	0.0856	0.0770	−0.0109	0.0614	0.0172
9/1/2019	0.0206	−0.0804	0.1107	0.0071	0.0910	0.0204
10/1/2019	0.0413	0.0054	0.0743	−0.0189	0.0585	0.0340
11/1/2019	0.0683	0.0087	0.1021	0.0443	0.0204	0.0286
12/1/2019	0.0206	0.0723	0.0540	0.0551	−0.1106	−0.0016
1/1/2020	−0.0967	−0.0945	−0.1168	−0.0841	−0.0475	−0.0841
2/1/2020	−0.0187	−0.1388	−0.0676	−0.1727	−0.0660	−0.1251
3/1/2020	0.1442	0.1319	0.1554	0.0462	0.0029	0.1268
4/1/2020	−0.0086	−0.0053	0.0822	0.0172	0.0417	0.0453
5/1/2020	−0.0480	−0.0201	0.1505	−0.0428	0.0530	0.0184
6/1/2020	0.0365	0.0180	0.1651	0.0669	0.0504	0.0551
7/1/2020	0.0525	0.0030	0.2144	0.0485	0.0790	0.0701

(continued)

Date	JNJ monthly Returns	IBM monthly returns	Apple monthly returns	Coca-Cola monthly returns	Caterpillar monthly returns	GSPC monthly Returns
8/1/2020	−0.0231	−0.0004	−0.1009	−0.0032	0.0481	−0.0392
9/1/2020	−0.0791	−0.0823	−0.0600	−0.0187	0.0530	−0.0277
10/1/2020	0.0552	0.1062	0.0936	0.0737	0.1121	0.1075
11/1/2020	0.0953	0.0339	0.1165	0.0711	0.0486	0.0371
12/1/2020	0.0365	−0.0538	−0.0055	−0.1220	0.0045	−0.0111
1/1/2021	−0.0286	−0.0015	−0.0811	0.0174	0.1870	0.0261
2/1/2021	0.0436	0.1355	0.0088	0.0759	0.0741	0.0424
3/1/2021	−0.0084	−0.0002	0.0070	0.0045	0.0038	0.0118

Step 3:

By using statistical calculations, we can get monthly average returns (mean), monthly standard deviation (variance), monthly skewness, and monthly kurtosis for each company.

	JNJ	IBM	AAPL	KO	CAT	S&P 500
Monthly Average Returns	0.0105	0.0064	0.0341	0.0080	0.0236	0.0122
Monthly Std. Deviation	0.0491	0.0713	0.0864	0.0452	0.0728	0.0436
Monthly Skewness	−0.1373	−0.3763	−0.3242	−1.2618	−0.7115	−0.5396
Monthly Kurtosis	0.5526	1.7864	-0.1893	3.3690	0.5872	1.9974

By dividing monthly average returns with monthly standard deviation, we can obtain coefficient variation for each company. The coefficient variance for Johnson & Johnson is 4.6904, for IBM is 11.1454, for Apple is 2.5311, for Coca-Cola is 5.6497, for Caterpillar is 3.0843, and for S&P 500 is 3.5750.

Additionally, we can obtain variance–covariance matrix by using monthly real return data.

Variance–Covariance Matrix:

	JNJ	IBM	AAPL	KO	CAT	S&P 500
JNJ	0.0023					
IBM	0.0015	0.0050				
AAPL	0.0017	0.0021	0.0071			
Cola	0.0011	0.0014	0.0011	0.0022		
Cat	0.0009	0.0026	0.0011	0.0006	0.0053	
SP 500	0.0012	0.0022	0.0022	0.0011	0.0017	0.0018

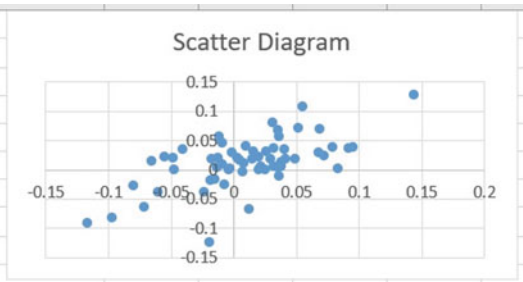
Assuming monthly risk-free rate is 0.00167, we can calculate Sharpe performance measure. The Sharpe performance measure for Johnson & Johnson is 0.1792, for IBM is 0.0663, for Apple is 0.3758, for Coca-Cola is 0.1400, for Caterpillar is 0.3013, and for S&P 500 is 0.2414.

Step 4:

Using excel program, we can obtain Beta Coefficient for each company. As the following figures show, Beta Coefficient for Johnson & Johnson is 0.7032, for IBM is 1.2442, for Apple is 1.2345, for Coca-Cola is 0.6406, and for Caterpillar is 0.9651.

Regression result for Johnson & Johnson

SUMMARY OUTPUT



Regression Statistics	
Multiple R	0.621722749
R Square	0.386539177
Adjusted R Square	0.376482442
Standard Error	0.037955638
Observations	63

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.055371859	0.055372	38.43585	5.38E-08
Residual	61	0.087878456	0.001441		
Total	62	0.143250315			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	0.001673129	0.004988496	0.335397	0.738476	-0.0083	0.011648	-0.0083	0.011648
GSPC Returns	0.70316021	0.113419056	6.199666	5.38E-08	0.476365	0.929956	0.476365	0.929956

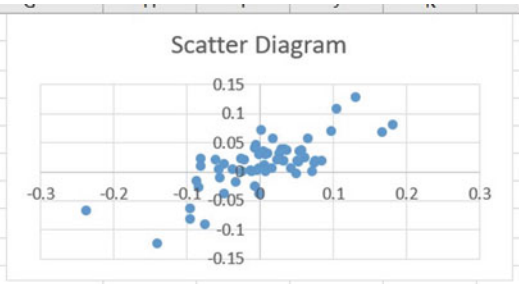
$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Beta Coefficient of JNJ	0.7032
Average Monthly Returns of JNJ	0.0105
Monthly Risk Free Rate	0.00167
Annual	
Beta Coefficient of JNJ	0.7032
Annual Returns of JNJ	0.125510126
Annual Risk Free Rate	0.02
Long-term Market Risk Premium	0.07
Cost of Equity of JNJ	0.069221215

$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Regression result for IBM

SUMMARY OUTPUT



Regression Statistics	
Multiple R	0.741223083
R Square	0.54941166
Adjusted R Square	0.542024965
Standard Error	0.04827716
Observations	63

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.173352927	0.173353	74.37856	3.74E-12
Residual	61	0.142171733	0.002331		
Total	62	0.31552466			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-0.008319285	0.006345049	-1.31115	0.194723	-0.02101	0.004368	-0.02101	0.004368
GSPC Returns	1.244157398	0.144261834	8.6243	3.74E-12	0.955688	1.532627	0.955688	1.532627

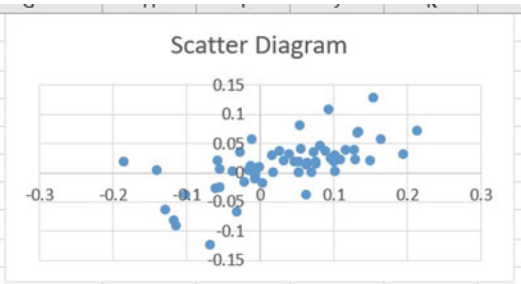
$$(Cost\ of\ Equity)_i = R_f + \beta_i\left[E(R_M) - R_f\right]$$

Beta Coefficient of IBM	1.2442
Average Monthly Returns of IBM	0.0105
Monthly Risk Free Rate	0.00167
Annual	
Beta Coefficient of IBM	1.2442
Annual Returns of IBM	0.125510126
Annual Risk Free Rate	0.02
Long-term Market Risk Premium	0.07
Cost of Equity of IBM	0.107091018

$$(Cost\ of\ Equity)_i = R_f + \beta_i\left[E(R_M) - R_f\right]$$

Regression result for Apple

SUMMARY OUTPUT



Regression Statistics	
Multiple R	0.616784808
R Square	0.3804235
Adjusted R Square	0.370266508
Standard Error	0.067503346
Observations	63

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.170668292	0.170668	37.45435	7.34E-08
Residual	61	0.277958809	0.004557		
Total	62	0.4486271			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	0.015386694	0.00887194	1.73431	0.087914	-0.00235	0.033127	-0.00235	0.033127
GSPC Returns	1.234485966	0.201713535	6.119996	7.34E-08	0.831135	1.637837	0.831135	1.637837

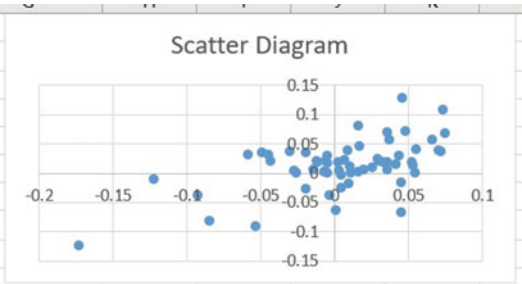
$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Beta Coefficient of Apple	1.2345
Average Monthly Returns of Appl	0.0105
Monthly Risk Free Rate	0.00167
Annual	
Beta Coefficient of Apple	1.2345
Annual Returns of Apple	0.125510126
Annual Risk Free Rate	0.02
Long-term Market Risk Premium	0.07
Cost of Equity of Apple	0.106414018

$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Regression result for Coca-Cola

SUMMARY OUTPUT



Regression Statistics	
Multiple R	0.573210521
R Square	0.328570301
Adjusted R Square	0.317563257
Standard Error	0.039235302
Observations	63

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.045952759	0.045953	29.85091	9.11E-07
Residual	61	0.093903944	0.001539		
Total	62	0.139856703			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-0.000919725	0.005156681	-0.17836	0.859034	-0.01123	0.009392	-0.01123	0.009392
GSPC Returns	0.640568432	0.11724295	5.463599	9.11E-07	0.406127	0.87501	0.406127	0.87501

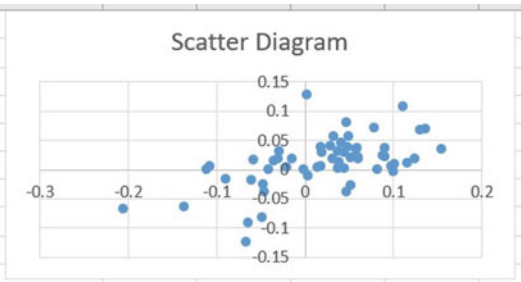
$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Beta Coefficient of KO	0.6406
Average Monthly Returns of KO	0.0105
Monthly Risk Free Rate	0.00167
Annual	
Beta Coefficient of KO	0.6406
Annual Returns of KO	0.125510126
Annual Risk Free Rate	0.02
Long-term Market Risk Premium	0.07
Cost of Equity of KO	0.06483979

$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Regression result for Caterpillar

SUMMARY OUTPUT



Regression Statistics	
Multiple R	0.556611573
R Square	0.309816443
Adjusted R Square	0.298501958
Standard Error	0.061722086
Observations	63

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.104315989	0.104316	27.38229	2.17E-06
Residual	61	0.232386569	0.00381		
Total	62	0.336702558			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	0.014284366	0.008112111	1.760869	0.083273	-0.00194	0.030506	-0.00194	0.030506
GSPC Returns	0.965128579	0.184437969	5.232809	2.17E-06	0.596322	1.333935	0.596322	1.333935

$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Beta Coefficient of CAT	0.9651
Average Monthly Returns of CAT	0.0105
Monthly Risk Free Rate	0.00167
Annual	
Beta Coefficient of CAT	0.9651
Annual Returns of CAT	0.125510126
Annual Risk Free Rate	0.02
Long-term Market Risk Premium	0.07
Cost of Equity of CAT	0.087559001

$$(Cost\ of\ Equity)_i = R_f + \beta_i[E(R_M) - R_f]$$

Step 5:

By using the formulas, we can calculate Variance minimization and Sharpe performance maximization in Excel program.

Variance Minimization formula:

$$W_A = \frac{\sigma_B^2 - Cov(R_A, R_B)}{\sigma_A^2 + \sigma_B^2 - Cov(R_A, R_B)}$$

$$W_B = 1 - W_A$$

Table results for variance minimization:

	W_A		W_B
JNJ & SP 500	0.188354	JNJ & SP 500	0.811646
IBM & SP 500	−0.09488	IBM & SP 500	1.094882
AAPL & SP 500	−0.06217	AAPL & SP 500	1.062174
KO & SP 500	0.223491	KO & SP 500	0.776509
CAT & SP 500	0.011466	CAT & SP 500	0.988534

Sharpe performance maximization formula:

$$W_1 = \frac{[E(R_1) - R_f]\sigma_2^2 - [E(R_2) - R_f]Cov(R_1, R_2)}{[E(R_1) - R_f]\sigma_2^2 + [E(R_2) - R_f]\sigma_1^2 - [E(R_1) - R_f + E(R_2) - R_f]Cov(R_1, R_2)}$$

$$W_2 = 1 - W_1$$

Table results for Sharpe performance maximization:

	W_1		W_2
JNJ & SP 500	0.159602	JNJ & SP 500	0.840398
IBM & SP 500	−0.54313	IBM & SP 500	1.543127
AAPL & SP 500	0.902548	AAPL & SP 500	0.097452
KO & SP 500	−0.04864	KO & SP 500	1.048639
CAT & SP 500	0.528357	CAT & SP 500	0.471643

Bibliography

Bodie, Z., Kane, A., and Marcus, A., Investments (12th ed), McGraw-Hill Book Company, New York, 2021.

Bowman, R.G. "The Theoretical Relationship between Systematic Risk and Financial (Accounting) Variables," *Journal of Finance*, 34 (June 1979, pp. 617–30).

Chiou, W.J. Paul, Lee, Alice C., and Chang, Chiu-Chi A., "Do Investors Still Benefit from International Diversification with Investment Constraint?" The Quarterly Review of Economics and Finance, vol. 49, issue 2, 448–483, 2009.

Cochrane, John H., *Asset Pricing* (Revised Edition), Princeton University Press, 2005.

Elton, E.J., M.J. Gruber, Brown, S.J., and Goetzmann, W.N., *Modern Portfolio Theory and Investment Analysis* (Ninth ed), Wiley, 2013.

Francis. J.C. and D. Kim. *Modern Portfolio Theory: Foundations, Analysis, and New Developments*, Wiley, 2013.

Grabowski, R., C. Nunes, and J. P. Harrington. "2020 SBBI Yearbook: Stocks, Bonds, Bills, and Inflation."

Ibbotson, R., R. J. Grabowski, J. P. Harrington, and C. Nunes, *2016 Stocks, Bonds, Bills, and Inflation (SBBI) Yearbook* (First edition) Wiley, 2016.

Ibbotson, R. G. and R. A. Sinquefield. "Stocks, Bonds, Bills, and Inflation" (Chicago: Morningstar, 2015).

Lee, Cheng F., and Alice C. Lee, Encyclopedia of Finance, 3rd ed., Springer, New York, New York, 2022, forthcoming.

Lee, C. F, J. C. Lee, J. R. Chang, and T. Tai, Essentials of Excel, Excel VBA, SAS and Minitab for Statistical and Financial Analyses, Springer International Publishing, 2016.

Lee, C. F., J. C. Lee, and A. C. Lee, Statistics for Business and Financial Economics, Springer Academic Publishers, 3rd Edition, 2013a.

Lee, Cheng, F., Finnerty, J., Lee, J., Lee, A. C., and Wort. D, Security Analysis, Portfolio Management, and Financial Derivatives, World Scientific, 2013b.

Markowitz, H. M. *Portfolio Selection: Efficient Diversification of Investments* (New York: John Wiley & Sons, 1959).

Modigliani, F. and G.A. Pogue. "An Introduction to Risk and Return," *Financial Analysts Journal,* 30 (May-June 1974, pp. 69–86).

Lee, Cheng F., and Alice C. Lee, Encyclopedia of Finance, 2nd ed. Springer, New York, New York, 2013.

Asset Allocation and Markowitz Portfolio-Selection Model

23

23.1 Introduction

In Chap. 22, we have discussed risk estimation, risk diversification, and the procedure of obtaining optimal portfolio weights for two-security cases. In this chapter basic portfolio analysis concepts and techniques are discussed in the Markowitz portfolio-selection model and other related issues in portfolio analysis. Before Harry Markowitz (1952, 1959) developed his portfolio-selection technique into what came to be called modern portfolio theory (MPT), security-selection models focused primarily on the returns generated by investment opportunities. The Markowitz theory retained the emphasis on return, but it elevated risk to the same level of importance, and the concept of portfolio risk was born. Although risk has been considered an important factor and variance an accepted way of measuring risk, Markowitz was the first to clearly and rigorously show how the variance of a portfolio can be reduced through the impact of diversification. He demonstrated that by combining securities that are not perfectly positively correlated into a portfolio, the portfolio variance can be reduced.

The **Markowitz model** is based on several assumptions concerning the behavior of investors:

i. A probability distribution of possible returns over some holding period can be estimated by investors.
ii. Investors have single-period utility functions in which they maximize utility within the framework of diminishing marginal utility of wealth.
iii. Variability in the possible values of return is used by investors to measure risk.
iv. Investors use only expected return and risk to make investment decisions.
v. Expected return and risk as used by investors are measured by the first two moments of the probability distribution of returns-expected value and variance.
vi. Return is desirable; risk is to be avoided.
vii. It follows, then, that a security or portfolio is considered efficient if there exists no other investment opportunity with a higher level of return at a given level of risk and no other opportunity with a lower level of risk at a given level of return.

The structure of this chapter will now be discussed. In Sect. 23.2 we discuss utility theory, utility functions, and indifference curves. Efficient portfolios are examined in Sect. 23.3, and techniques for calculating the efficient frontier with short selling are discussed in Sect. 23.4. Finally, Sect. 23.5 summarizes the chapter. In Appendix 23A, Excel is used to calculate the optimal weight for Johnson & Johnson, IBM, and Caterpillar, and in Appendix 23B, the graphical analysis in Markowitz portfolio-selection model for three-security cases is presented.

23.2 Utility Theory, Utility Functions, and Indifference Curves

In this section, utility theory and functions, which are needed for portfolio analysis and capital asset models will be discussed in detail. In addition, indifference curves are also presented.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_23

23.2.1 Utility Theory

Utility theory is the foundation for the theory of choice under uncertainty. Following Henderson and Quandt (1980), cardinal and ordinal theories are the two major alternatives used by economists to determine how people and societies choose to allocate scarce resources and to distribute wealth among one another over time.[1]

23.2.2 Utility Functions

Economists define the relationships between psychological satisfaction and wealth as "utility." An upward sloping relationship, as shown in Fig. 23.1, identifies the phenomena of increasing wealth and increasing satisfaction as being directly related. These relationships can be classified into linear, concave, and convex utility functions.

In Fig. 23.1a, for each unit change in wealth, there is a linear utility function, and equal increase in satisfaction or utility. A doubling of wealth will double satisfaction, and so on. If an investor's utility function is a linear utility function, we call this kind of investor as risk-neutral investor. This is probably not very realistic: $1 increase in wealth from $1 to $2 is probably more important than an increase from $1 million to $2 million, because the marginal utility diminishes with increased wealth.

In Fig. 23.1b, the concave utility function shows the relationship of an increase in wealth and a less than proportional increase in utility. In other words, the marginal utility of wealth decreases as wealth increases. As mentioned above, the $1 increase from $1 to $2 of wealth is more important to the individual than the increase from $1,000,000 to $1,000,001. Each successive increase in wealth adds less satisfaction as the level of wealth rises. We call an investor with a concave utility function as risk-averse investor.

Finally, Fig. 23.1c is a convex utility function, which denotes a more than proportional increase in satisfaction for each increase in wealth. Behaviorally, the richer you are the more satisfaction you receive in getting an additional dollar of wealth. Investors with convex utility functions are called risk-seeking investors.

The utility theory, primarily used in finance, is developed by Von Neumann and Morgenstern (VNM, 1947). VNM define investor utility as a function of rates of return or wealth. Mao (1969) points out that the VNM utility theory is really somewhere between the cardinal and ordinal utility theories. The function associated with the VNM's utility theory in terms of wealth can be defined as

$$U = f(\mu_w, \sigma_w)$$

where μ_w indicates expected future wealth and σ_w represents the predicted standard deviation of the possible divergence of actual future wealth from μ_w.

Investors are expected to prefer a higher expected future wealth to a lower value. Moreover, they are generally risk-averse as well. That is, they prefer a lower value of σ_w to a higher value, given the level of μ_w.[2] These assumptions imply that the **indifference curves** relating to μ_w and σ_w will be upward sloping, as indicated in Fig. 23.2. In Fig. 23.2, each indifference curve is an expected utility isoquant showing all the various combinations of risk and return that provide an equal amount of expected utility for the investor.

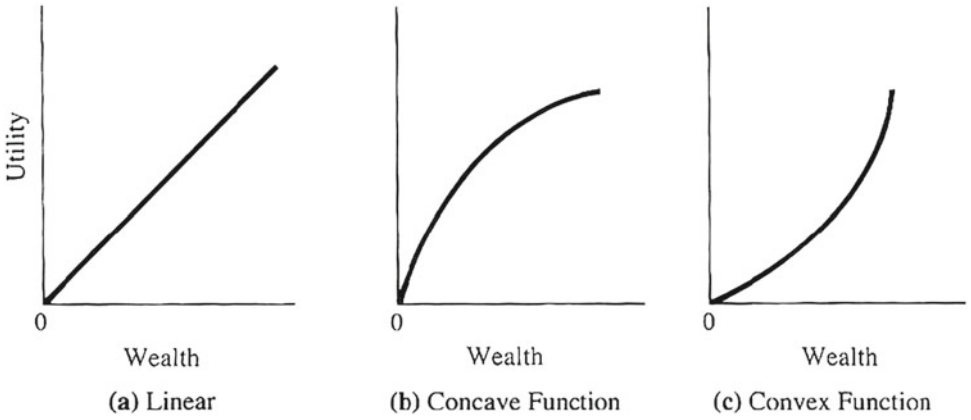
In explaining how investment decisions or portfolio choices are made, utility theory is used here not to imply that individuals actually make decisions using a utility curve, but rather as an expository vehicle that helps explain how investors presumably act. In general, humans behave as if more is better than less (the utility curve is upward sloping) and marginal utility is decreasing (the utility curve is concave).

In an uncertain environment it becomes necessary to ascertain how different individuals will react to risky situations. The risk is defined as the probability of success or failure. Alternatively, risk could be described as variability of outcomes, payoffs, or returns. This implies that there is a distribution of outcomes associated with each investment decisions. What is needed is a linkage between utility or expected utility and risk. Expected utility has been defined as the numerical value

[1] A *cardinal utility* implies that a consumer is capable of assigning to every commodity or combination of commodities a number representing the amount or degree of utility associated with it. An *ordinary utility* implies that a consumer will not be able to assign numbers that represent (in arbitrary unit) the degree or amount of utility associated with commodity or combination of commodity. The consumer can only rank and order the amount or degree of utility associated with commodity.

[2] Technically, these conditions can be represented mathematically by $\partial U/\partial \mu_w > 0$ and $\partial U/\partial \sigma_w < 0$.

Fig. 23.1 Utility functions



assigned to the probability distribution associated with a particular portfolio's return. This numerical value is calculated by taking a weighted average of the utilities of the various possible returns. The weights are the probabilities of occurrence associated with each of the possible returns. It is calculated by the following formula:

$$E(U) = \sum_{i}^{n} U(w_i)P_i, \tag{23.1}$$

where

$E(U)$ = expected utility;
$U(w_i)$ = the utility of the ith outcome w_i; and.
P_i = the probability of the ith outcome.

Example 23.1 Given investments A and B as shown in Table 23.1, determine the utilities of A and B for the given utility functions.

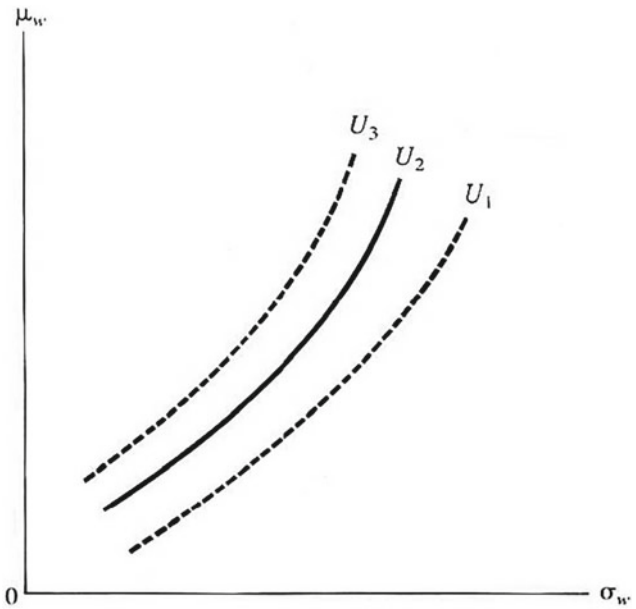


Fig. 23.2 Indifference curves of utility functions

Table 23.1 Possible outcomes of investments A and B

A		B	
Outcome w_i	Probability	**Outcome** w_i	Probability
10	2/5	9	2/3
5	2/5	3	1/3
1	1/5		

$$U(w) = w_i$$
$$U(w) = w_i^2$$
$$U(w) = w_i^2 - w_i$$

Solution

For $U(w) = w_i$

$$\text{Expected Utility of A} = \frac{2}{5}(10) + \frac{2}{5}(5) + \frac{1}{5}(1) = 6\frac{1}{5}$$
$$\text{Expected Utility of B} = \frac{2}{3}(9) + \frac{1}{3}(3) = 7$$

For $U(w) = w_i^2$

$$\text{Expected Utility of A} = \frac{2}{5}(100) + \frac{2}{5}(25) + \frac{1}{5}(1) = 50\frac{1}{5}$$
$$\text{Expected Utility of B} = \frac{2}{3}(81) + \frac{1}{3}(9) = 57$$

$U(w) = w_i^2 - w_i$

$$\text{Expected Utility of A} = 50\frac{1}{5} - 6\frac{1}{5} = 44$$
$$\text{Expected Utility of B} = 57 - 7 = 50$$

In all three cases, B has the higher degree of utility because it has a higher expected value as well as a smaller dispersion than A.

23.2.2.1 Linear Utility Function and Risk

It is useful to consider how the shape of an individual's utility function affects his or her reaction to risk. Assume that an individual who has \$5,000, and whose behavior is a linear utility function (Fig. 23.1a), is offered a chance to gain \$10,000 with a probability of 1/2 or to lose \$10,000 with a probability of 1/2. What should he or she pay for such an opportunity? The answer is nothing, for as can be seen in Fig. 23.3, this individual would be no better or worse off accepting or rejecting this opportunity. If he rejected the offer, his wealth would be \$5,000 with utility U_1; if he paid nothing for the opportunity, his wealth would remain as \$5,000 with utility U_1. Any payment for this chance would reduce his wealth and therefore be undesirable. This is so because the expected value of the fair game is zero:

$$\frac{1}{2}(10,000) + \frac{1}{2}(-10,000) = 0.$$

Figure 23.3 illustrates this linear utility function concept. In the following section, we will analyze the implication of concave utility function.

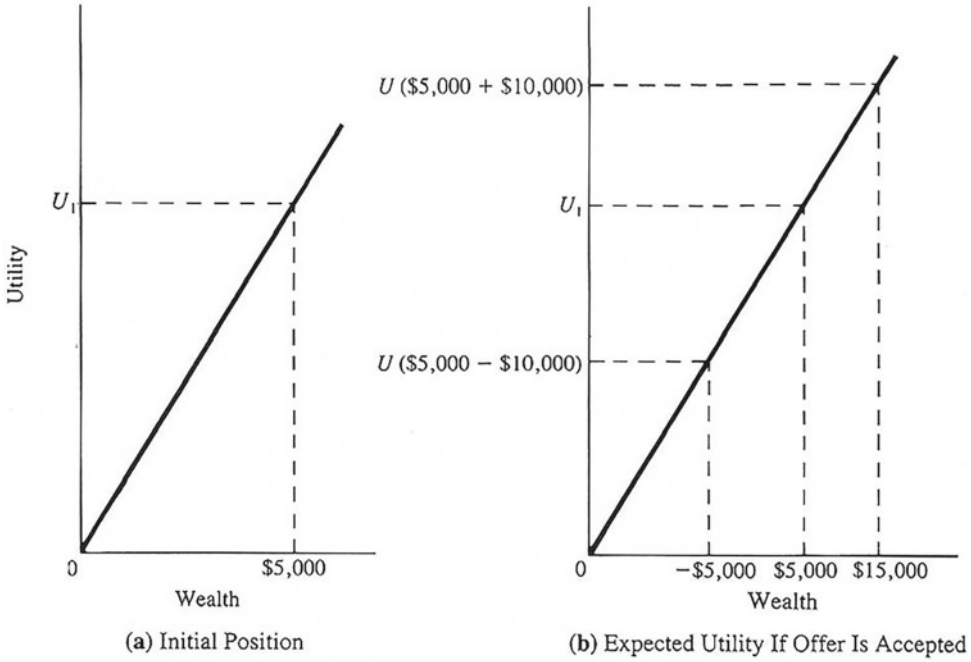


Fig. 23.3 Risk-neutral investors and fair games

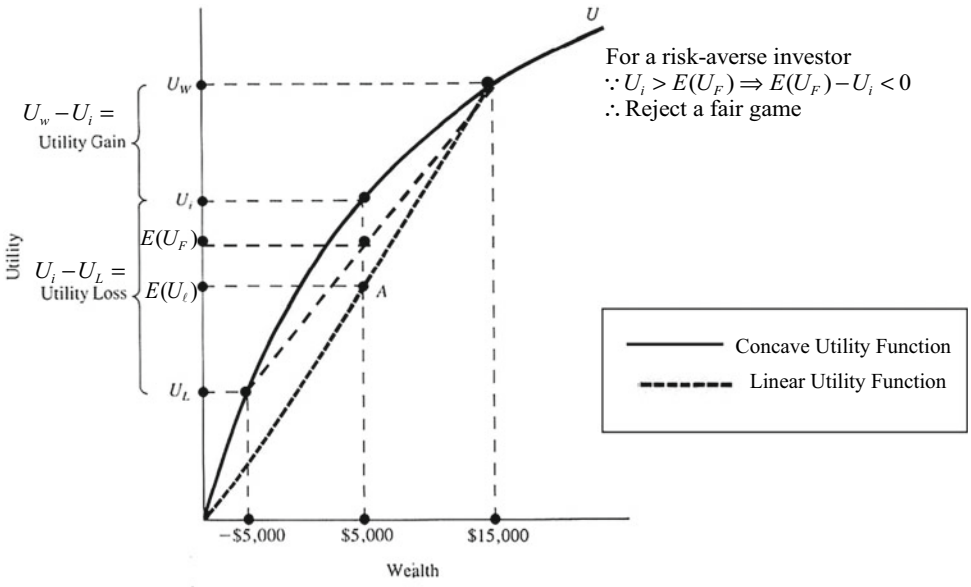


Fig. 23.4 Utility function for risk-averse investor

23.2.2.2 Concave Utility Function and Risk

Now consider an individual whose behavior is a concave utility function. If this individual participates and wins, his or her utility is shown by point U_W in Fig. 23.4. But if the individual loses, his or her position is shown by U_L. The expected value of this fair game, having a 50% chance of winning and a 50% chance of losing, is shown by point A.[3] The utility of the fair game is U_F. A comparison of U_F with the initial position, U_i, shows that the investor should not accept this fair game.

As shown in Fig. 23.4 the utility of winning $(U_W - U_i)$ is less than the utility of losing $(U_i - U_L)$. Therefore, the utility of doing nothing is greater than the expected utility of accepting the fair game. In fact, the individual should be willing to pay up to the difference between the utility of winning and the utility of losing $(U_i - U_L) - (U_W - U_i)$ to avoid being involved in

[3] Following Fama (1970) and Alexander and Francis (1986, p.177) a fair game means that the expected returns, given information set θ, equal the expected returns without the information set. Note that this does not mean the expected returns are zero or positive—they could be negative.

this situation. Alexander and Francis (1986) theoretically analyze this issue in more detail. Hence investors with concave utility functions are said to be risk-averse. That is, they would reject a fair game because the utility derived from winning is less than the utility lost should they lose the game. In other words, the expected utility of participating in a fair game is negative. Example 23.2 will provide illustration of linear, concave, and convex utility function.

$E(U_F) = \frac{1}{2}(U_w + U_L)$ is the expected utility of the fair game under a concave utility function.

$E(U_\ell)$ is the expected utility of the fair game under a linear utility function.

Example 23.2 Given the following utility functions for four investors, what can you conclude about their reaction toward a fair game?

(1) $u(w) = w + 4$
(2) $u(w) = w - \frac{1}{2}w^2$ (quadratic utility function)
(3) $u(w) = -e^{-2w}$ (negative exponential utility function), and
(4) $u(w) = w^2 - 4w$.

Evaluate the second derivative of the utility functions according to the following rules.

$$u''(w) < 0 \text{ implies risk averse}$$
$$u''(w) = 0 \text{ implies risk neutral}$$
$$u''(w) > 0 \text{ implies risk seeker}$$

Solution

$$u(w) = w + 4$$
$$u'(w) = 1$$
$$u''(w) = 0$$

This implies risk neutrality; it is indifference for the investor to accept or reject a fair gamble.

$$u(w) = w - \frac{1}{2}w^2$$
$$u'(w) = -w^2$$
$$u''(w) = -2w < 0 \text{ (We assume wealth is non} - \text{negative.)}$$

This implies the investor is risk-averse and would reject a fair gamble.

$$u(w) = -e^{-2w}$$
$$u'(w) = 2e^{-2w}$$
$$u''(w) = -4\,e^{-2w} < 0$$

This implies risk adversity; the investor would reject a fair gamble.

$$u(w) = w^2 - 4w$$
$$u'(w) = 2w - 4$$
$$u''(w) = 2 > 0$$

This implies risk preference; the investor would seek a fair gamble.

The convex utility function is not realistic in real-world decisions; therefore, it is not further explored at this point. The following section discusses the implications of alternative utility functions in terms of indifference curves. In their paper, "Alternative utility functions: review, analysis, and comparison," Harel et al. (2018) reviewed alternative utility functions in detail.

23.2.3 Risk Aversion and Asset Allocation

But when risk increases along with return, the most attractive portfolio is not obvious. How can investors quantify the rate at which they are willing to trade-off return against risk? By assuming that each investor can assign a welfare, or utility, score to competing investment portfolios based on the expected return and risk of those portfolios. Higher utility values are assigned to portfolios with more attractive risk–return profiles. Portfolios receive higher utility scores for higher expected returns and lower scores for higher volatility.

Assigning a portfolio with expected return and variance of returns, the following utility score:

$$U = E(r) - \frac{1}{2} \times A \times s^2, \tag{23.2}$$

where

U = utility;
$E(r)$ = expected return on the asset or portfolio;
A = coefficient of risk aversion; and.
s^2 = variance of returns.

Example 23.3 Consider three investors with different degrees of risk aversion: A1 = 2.0, A2 = 3.0, and A3 = 4.0, all of whom are evaluating the three portfolios in Table 23.2. Because the risk-free rate is assumed to be 5%, utility score implies that all three investors would assign a utility score of 0.05 to the risk-free alternative. Table 23.2 presents the utility scores that would be assigned by each investor to each portfolio. The portfolio with the highest utility score for each investor appears in bold. Notice that the high-risk portfolio, H, would be chosen only by the investor with the lowest degree of risk aversion, A1 = 2.0, while the low-risk portfolio, L would be passed over even by the most risk-averse of our three investors. All three portfolios beat the risk-free alternative for the investors with levels of risk aversion given in Table 23.2.

Example 23.4 Assuming an investor who faces a risk-free rate and a risky portfolio with expected return and standard deviation will find that for any choice of W, the expected return of the complete portfolio is given by.

$$E(r_C) = r_f + W\left[E\left(r_p\right) - r_f\right]. \tag{23.3}$$

The variance of the complete portfolio is

$$\sigma_C^2 = W^2\sigma_p^2. \tag{23.4}$$

The investor attempts to maximize utility by choosing the best allocation to the risk asset, W. To solve the utility maximization problem, we write the problem as follows:

Table 23.2 Utility scores assigned by each investor to each portfolio

Investor: risk-aversion (A)	Utility score portfolio (L) $E(r) = 0.07;\sigma = 0.05$	Utility score portfolio (M) $E(r) = 0.09;\sigma = 0.1$	Utility score portfolio (H) $E(r) = 0.13;\sigma = 0.2$
2.0	$0.07 - \frac{1}{2} \times 2 \times 0.05^2 = 0.0675$	$0.09 - \frac{1}{2} \times 2 \times 0.1^2 = 0.0800$	$0.13 - \frac{1}{2} \times 2 \times 0.2^2 = 0.0900$
3.0	$0.07 - \frac{1}{2} \times 3 \times 0.05^2 = 0.0663$	$0.09 - \frac{1}{2} \times 3 \times 1^2 = 0.0750$	$0.13 - \frac{1}{2} \times 3 \times 0.2^2 = 0.0700$
4.0	$0.07 - \frac{1}{2} \times 4 \times 0.05^2 = 0.0650$	$0.09 - \frac{1}{2} \times 4 \times 0.1^2 = 0.0700$	$0.13 - \frac{1}{2} \times 4 \times 0.2^2 = 0.0500$

$$Max\ U = E(r_C) - \frac{1}{2}A\sigma_C^2 = r_f + W\left[E\left(r_p\right) - r_f\right] - \frac{1}{2}AW^2\sigma_p^2. \tag{23.5}$$

Taking derivative of U with respective to W and set the derivative of this expression to zero, W yields the optimal position for risk-averse investors in the risk asset, W*, as follows:

$$W^* = \frac{E\left(r_p\right) - r_f}{A\sigma_p^2}. \tag{23.6}$$

In the numerical example, we assume that the expected rate of return on risky asset is $E\left(r_p\right) = 16\%$ and $\sigma_P = 20\%$, and the risk-free rate is $r_p = 6\%$. The optimal solution for an investor with a coefficient of risk aversion $A = 3$ is

$$W^* = \frac{0.16 - 0.06}{3 \times (0.20)^2} = 0.83.$$

Therefore, the investor will invest 83% of his investment budget in the risky asset and 17% of his investment budget in the risk-free asset.

23.2.4 Indifference Curves

Indifference (utility function) curves are abstract theoretical concepts. They cannot, as a practical matter, be used to actually measure how individuals make investment decisions—or any other decisions. They are, however, useful tools for building models that illustrate the relationship between risk and return. An investor's utility function can be utilized conceptually to derive an indifference curve, which shows individual preference for risk and return.[4] An indifference curve can be plotted in the risk–return space such that the investor's utility is equal all along its length. The investor is indifferent to various combinations of risk and return; hence, the name indifference curve. Various types of investor's indifference curves are shown in Fig. 23.5.

In the same level of satisfaction, the risk-averse investor requires more return for an extra unit of risk than the return the investor asks for previous one unit increase of risk. Therefore, the indifference curve is convex. Figure 23.5a presents two indifference curves which are risk-averse, and $U_1 > U_2$ (higher return and lower risk). For a risk-neutral investor, the indifference curve is a straight line. In the same level of satisfaction, the investor requires the same return for an extra unit of risk as the return he or she asks for previous one unit increase of risk. Figure 23.5b presents two indifference curves which are risk-neutral, and $U_1 > U_2$. For a risk-seeking investor, the indifference curve is a concave function. In the same level of satisfaction, the investor requires less return for an extra unit of risk than the return he asks for previous one unit increase of risk. Figure 23.5c presents two indifference curves which are risk-seeking, and $U_1 > U_2$.

The more risk-averse individuals will claim more premiums when they face uncertainty (risk). In Fig. 23.5d, when facing σ_0, investor 1 and investor 2 have the same expected return, $E(R_0)$. However, when the risk adds to σ_1, the expected return $E(R_1)$ investor 1 asks is higher than the expected return $E(R_2)$ investor 2 asks. Thus, investor 1 is more risk-averse than investor 2. Therefore, in return-risk plane, the larger slope of indifference curve which the investor has, the higher risk-averse level the investor is.

Later in this chapter different levels of risk and return are evaluated for securities and portfolios when a decision must be made concerning which security or portfolio is better than another. It is at that point in the analysis that indifference curves of hypothetical individuals are employed to help determine which securities or portfolios are desirable and which ones are not. Basically, values for return and risk will be plotted for a number of portfolios as well as indifference curves for different types of investors. The investor's optimal portfolio will then be the one identified with the highest level of utility for the various indifference curves.

Investors generally hold more than one type of investment asset in their portfolio. Besides securities, an investor may hold real estate, gold, art, and so on. Thus, as the measures of risk and return for individual securities developed, the measures of risk and return may be used for portfolios of risky assets. Risk-averse investors hold portfolios rather than individual securities as a means of eliminating unsystematic risk; hence the examination of risk and return will continue in terms of

[4] By definition, an indifference curve shows all combinations of products (investments) A and B that will yield same level of satisfaction or utility to consume. This kind of analysis is based upon ordinal rather than cardinal utility theory.

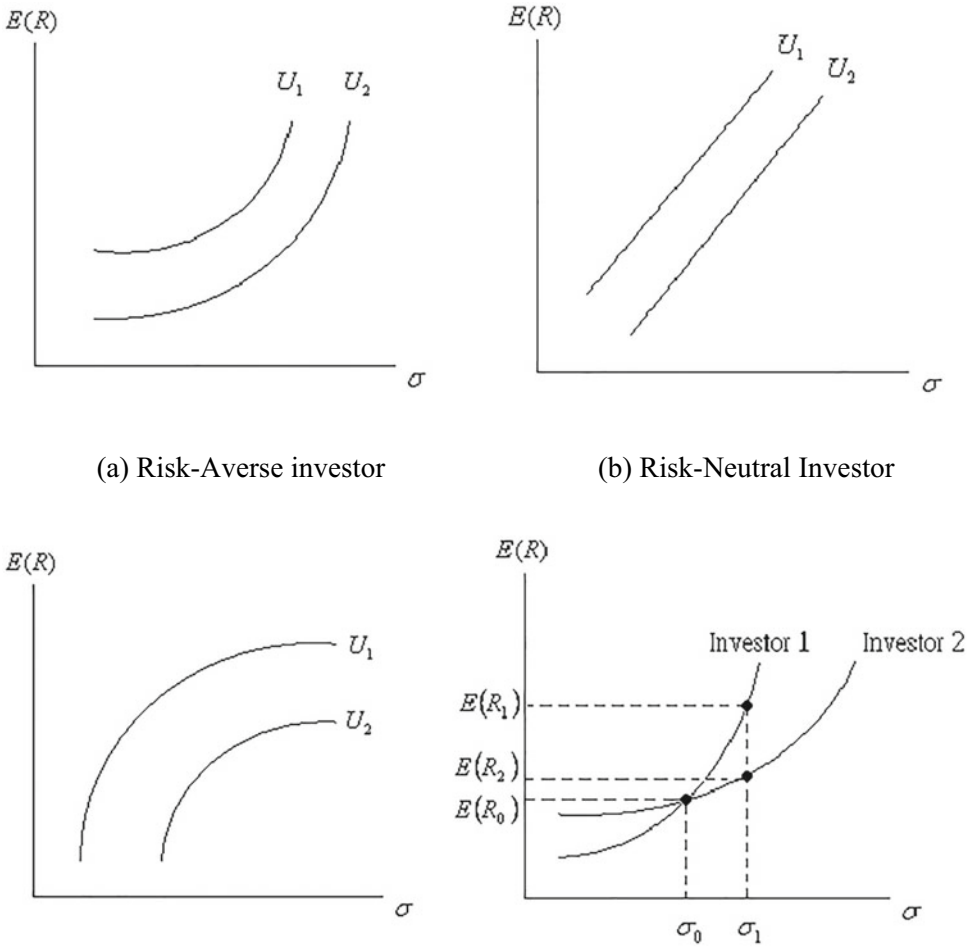


Fig. 23.5 Indifference curves for various types of individuals

portfolios rather than individual securities. Indifference curves can be used to indicate investors' *willingness* to trade risk for return; now investors' ability to trade risk for return needs to be represented in terms of indifference curves and efficient portfolios, as discussed in the next section.

23.3 Efficient Portfolios

It is well known that **efficient portfolios** may contain any number of asset combinations. The asset combinations can be illustrated in Fig. 23.6. In Fig. 23.6, the area within curve *XVYZ* is the feasible opportunity set representing all possible portfolio combinations. The curve *YV* represents all possible efficient portfolios and is the efficient frontier. The line segment *VX* is on the feasible opportunity set, but not on the efficient frontier; all points on *VX* represent inefficient portfolios.

The portfolios and securities that lie on the frontier *VX* in Fig. 23.6 would not be likely candidates for investors to hold. This is so because they do not meet the criteria of maximizing expected return for a given level of frisk or minimizing risk for a given level of return. This is easily seen by comparing the portfolio represented by points *X* and *X'*. Since investors always prefer more expected return than less for a given level of risk, *X'* is always better than *X*. Using similar reasoning, investors would always prefer *V* to *X* because it has both a higher return and a lower level of risk. In fact, the portfolio at point *V* is identified as the **minimum-variance portfolio**, since no other portfolio exists that has a lower variance.

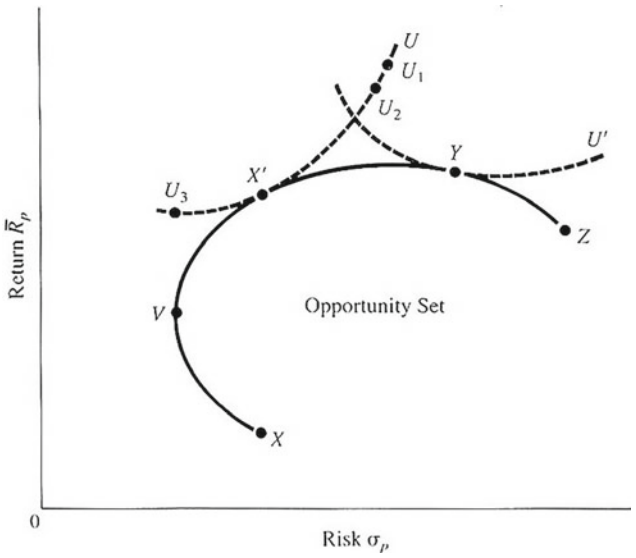


Fig. 23.6 Indifference Curves and the Minimum-Variance Set

23.3.1 Portfolio Combinations

For each of the combinations of individual securities and inefficient portfolios in Fig. 23.6, there is a corresponding portfolio along the efficient frontier that either has a higher return given the same risk or a lower risk given the same return. However, points on the efficient frontier do not dominate one another. While point *Y* has considerably higher return than point *V*, it also has considerably higher risk. The optimal portfolio along the efficient frontier is not unique with this model and depends upon the risk/return tradeoff utility function of each investor. Portfolio selection, then, is determined by plotting investors' utility functions together with the efficient frontier set of available investment opportunities. No two investors will select the same portfolio except by chance or if their utility curves are identical. In Fig. 23.6, two sets of indifference curves labeled *U* and *U′* are shown together with the efficient frontier. The *U* curves have a higher slope, indicating a greater level of risk aversion. The investor is indifferent to any combination of R_p and σ_p along a given curve, for example, U_1, U_2, or U_3. The U' curve would be appropriate for a less risk-averse investor—that is, one who would be willing to accept relatively higher risk to obtain higher levels of return. The optimal portfolio would be the one that provides the highest utility—a point in the northwest direction (higher return and lower risk). This point will be at the tangent of a utility curve and the efficient frontier. The tangency point investor in Fig. 23.6 is point *X'*; for the risk-averse it is point *Y*. Each investor is logically selecting the optimal portfolio given his or her risk–return preference, and neither is more correct than the other.

Individual investors sometimes find it necessary to restrict their portfolios to include a relatively small number of securities. Mutual-fund portfolios, on the other hand, often contain securities from more than 500 different companies. To give some idea of the number of securities that is necessary to achieve a high level of diversification, Levy and Sarnat (1971) considered a naïve strategy of equally weighted portfolios—that is, of dividing the total investment into equal proportions among component securities. Plotting the variance of the portfolios developed in this manner against the number of securities making up the portfolios will result in a function similar to that illustrated in Fig. 23.7. As can be seen, the additional reduction in portfolio variance rapidly levels off as the number of securities is increased beyond five. An earlier study by Evans and Archer (1968) shows that the percentage of diversification that can be achieved with randomly selected, equally weighted portfolios levels off rapidly beyond a portfolio size of about 15.

23.3.2 Short Selling

Short selling (or "going short") is a much regulated type of market transaction. It involves selling shares of a stock that are borrowed in expectation of a fall in the security's price. When and if the price declines, the investor buys an equivalent number of shares of the same stock at the new lower price and returns to the lender the stock that was borrowed. The Federal

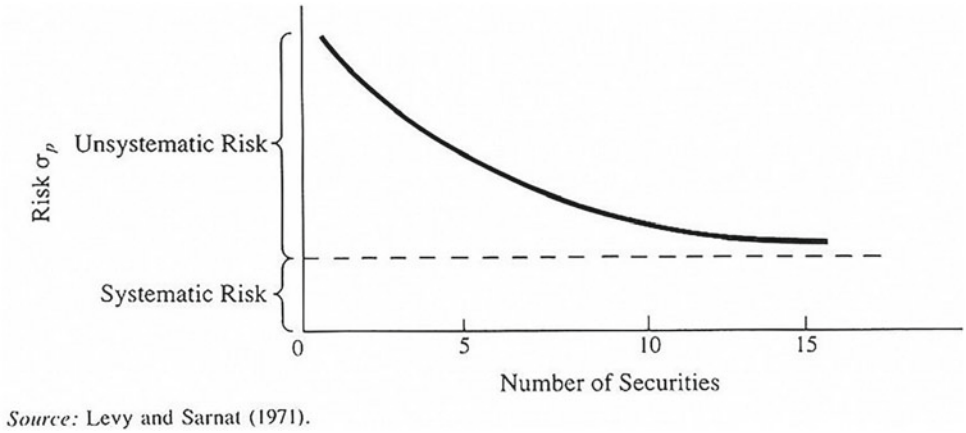


Fig. 23.7 Naive diversification reduces risk to the systematic level in a randomly selected portfolio

Source: Levy and Sarnat (1971).

Reserve Board requires short selling customers to deposit 50% of the net proceeds of such short sales with the brokerage firm carrying out the transaction. Another key requirement of a short sale, set by the Securities and Exchange Act of 1934, is that the short sale must occur at a price higher than the preceding sale—or at the same price as the preceding sale, if that took place at a higher price than a preceding price. This is the so-called uptick or zero-tick rule. It prevents the price of a security from successively falling because of continued short selling.

Relaxing the assumption of no short selling in this development of the efficient frontier involves a modification of the analysis of the previous section. The efficient frontier analyzed in the previous sections was bounded on both ends by Y and the minimum-variance portfolio V, respectively, as shown in Fig. 23.6. Point Y is called the **maximum return portfolio**, since there is no other portfolio with a higher return. This point is normally an efficient security or portfolio with the greatest level of risk and return. It could also be a portfolio of securities, all having the same highest levels of risk and return. Point Z is normally a single security with the lowest level of return, although it could be a portfolio of securities, all having the same low level of return.

The Black (1972) model is identical to the Markowitz model except that it allows for short selling.[5] That is, the non-negativity constraint on the amount that can be invested in each security is relaxed, $W_A \gtrless 0$. A negative value for the weight invested in a security is allowed, tantamount to allowing a short sale of the security. The new efficient frontier that can be derived with short selling is shown in Fig. 23.8a.

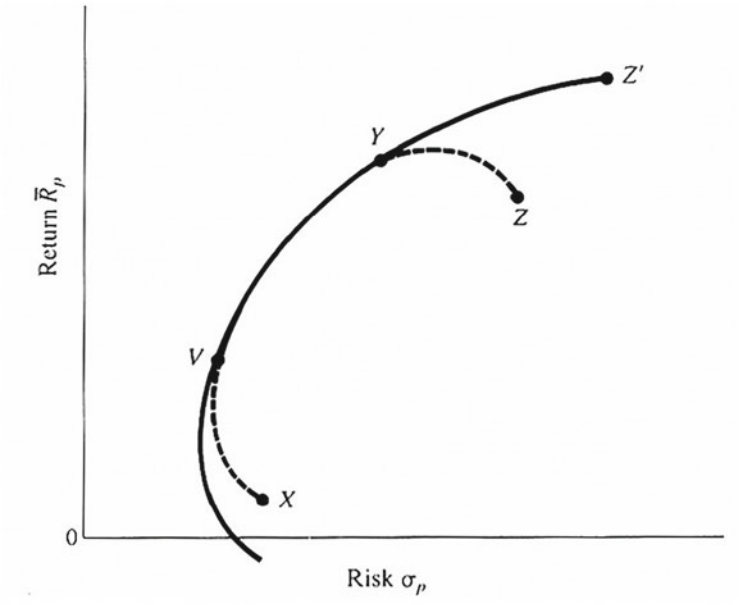
The major difference between the frontier in Fig. 23.8a (short selling) and Fig. 23.8b (no short selling) is the disappearance of the end points Y and Z. An investor could sell the lowest return security (Y). If the number of short sales is unrestricted, then by a continuous short selling of X and reinvesting in Y the investor could generate an infinite expected return. Hence the upper bound of the highest return portfolio would no longer be Y but infinity (shown by the arrow on the top of the efficient frontier). Likewise, the investor could short sell the highest return security U and reinvest the proceeds into the lowest yield security X, thereby generating a return less than the return on the lowest return security. Given no restriction on the amount of short selling, an infinitely negative return can be achieved, thereby removing the lower bound of X on the efficient frontier. But rational investors will not short sell a high-return stock and buy a low-return stock.

The portfolios on VZ' always dominate those of VX, as shown in Fig. 23.8a.

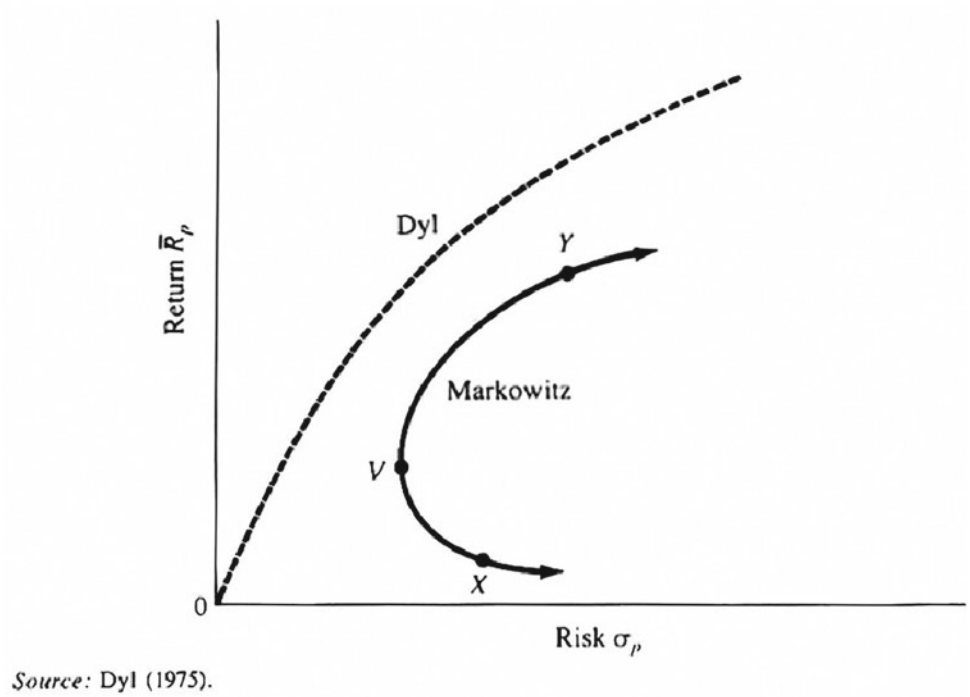
Whether an investor engages in any of this short selling activity depends on the investor's own unique set of indifference curves. Hence, short selling generally will increase the range of alternative investments from the minimum-variance portfolio to plus or minus infinity. However, in the Black model with short selling, no provision was made for the Securities and Exchange Commission (SEC) margin requirement. Dyl (1975) imposed the margin requirement on short selling and added it to the Markowitz development of the efficient frontier.

The Dyl Model. Dyl introduced short selling with margin requirements by creating a new set of risky securities, the ones sold short, which are negatively correlated with the existing set of risky securities. These new securities greatly enhance the diversification effect when they are placed in portfolios. The **Dyl model** affects the efficient frontier in two ways: (1) If the investor were to combine in equal weight any long position in a security or portfolio with a short position in a security or a

[5] Most texts do not identify the Markowitz model with restrictions on short sale. Markowitz (1952) in fact, excluded short sales.



(a) The Efficient Frontier with Short Selling



Source: Dyl (1975).

(b) Efficient Frontiers with and without Short Selling and Margin Requirements

Fig. 23.8 **a** The efficient frontier with short selling. **b** Efficient frontiers with and without short selling and margin requirements

portfolio, the resulting portfolio would yield zero return and zero variance and (2) any combination of unequal weighted long or short positions would yield portfolios with higher returns and lower risk levels. Overall, these two effects will yield an efficient frontier that dominates the Markowitz efficient frontier. Figure 80.8b compares the Dyl and Markowitz efficient frontiers.

Even though the inclusion of short selling changes the location and boundaries of the efficient frontier, the concavity of the curve is still intact. This is important in that it preserves the efficient frontier as the locus of optimal portfolios for risk-average investors. As long as the efficient frontier remains concave, by using indifference curves it will be possible to locate the optimal portfolio for each investor.

23.4 Techniques for Calculating the Efficient Frontier with Short Selling

Since there are thousands of securities from which the investor can choose for portfolio formation, it can be very costly and time consuming to calculate the efficient frontier set. One way of determining the optimal investment proportions in a portfolio is to hold the return constant and solve for the weighting factors ($W_1 \ldots, W_n$ for n securities) that minimize the variance, given the constraint that all weights sum to one and that the constant return equals the expected return developed from the portfolio. The optimal weights can then be obtained by minimizing the Lagrange function C for portfolio variance.

$$C = \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j r_{ij}\sigma_i\sigma_j + \lambda_1\left(1 - \sum_{i=1}^{n} W_i\right) + \lambda_2\left[E^* - \sum_{i=1}^{n} W_i E(R_i)\right] \tag{23.8}$$

in which λ_1 and λ_2 are the Lagrange multipliers, E^* and $E(R_i)$ are targeted rate of return and expected rate of return for security i, respectively, ρ_{ij} is the correlation coefficient between R_i and R_j, and other variables are as previously defined.

Function C has $n + 2$ unknowns: $W_1, \ldots, W_n, \lambda_1$, and λ_2. By differentiating C with respect to W_i, λ_1, and λ_2 and equating the first derivatives to zero, $n + 2$ equations can be derived to solve for $n + 2$ unknowns. As shown in the empirical example later in this chapter, matrix algebra is best suited to this solution process. By using this approach, the minimum-variance can be computed for any given level of expected portfolio return (subject to the other constraint that the weights sum to one). In practice it is best to use a computer because of the explosive increase in the number of calculations as the number of securities considered grows. The efficient set that is generated by the aforementioned approach (Eq. 23.1) is sometimes called the **minimum-variance set** because of the minimizing nature of the Lagrangian solution.

Thus far no specific distribution for measuring returns has been assumed. In most cases, specific distribution of returns will be needed when applying the portfolio model. The normal and log normal are two of the most commonly used distribution.

23.4.1 The Normal Distribution

As you undoubtedly are aware, the **normal (probability) distribution** is a bell-shaped curve centered on the mean of a given population or sample distribution. The area under the curve is an accumulation of probabilities that sum to one. With half of the area lying to the left and half to the right of the mean, probability statements may be made about underlying sample that makes up the distribution.

Within the scope of the Markowitz model, the normal distribution may be applied because of the use of the mean–variance assumptions. The formation of probability statements from the underlying sample is a result of the standard deviation (square root of the variance) quantifying the spread of the distribution. If the sample is normally distributed, approximately 68% of the observations will lie within one standard deviation on either side of the mean, approximately 95% will lie within two standard deviations, and approximately 99% will lie within three standard deviations. This ability to ascertain intervals of confidence for the observations allows the utilization of the normal distribution to predict ranges within which the portfolio returns will lie.

Utilizing the standard normal distribution (mean of zero, standard deviation of one), any set of mean–variance data can be standardized to develop probability statements about returns on a given portfolio. Standardization is an algebraic operation in which the mean is subtracted from a given return and this difference is divided by the standard deviation. The resultant metric is then standard normal and can be compared with tabulated values to calculate probabilistic quantities of occurrence. Within this framework, any hypothesized level of return can be assigned a probability of occurrence given a probabilistic value of occurrence that is based upon the nature of the sample.

For expository purposes, suppose the mean return on a particular investment is 10% for a given period, and that historically these returns have a variance of 0.16%, it is of interest to know what the probabilities are of obtaining a 15% or greater return, or a return less than or equal to 8%. These probabilities may be evaluated by using the normal distribution, as stated in notation below:

$$P\left(X \geq 0.15 \middle| \overline{R}_p = 0.1, \sigma_p^2 = 0.0016\right)$$
$$P\left(X \leq 0.18 \middle| \overline{R}_p = 0.1, \sigma_p^2 = 0.0016\right)$$

by standardizing the probability statements to z values:

$$P\left(z \geq \frac{0.15 - 0.1}{0.04} = 1.25\right)$$
$$P\left(z \leq \frac{0.08 - 0.1}{0.04} = -0.5\right) \tag{23.9}$$

These standardized values can then be compared with the tabulated z values found in tables of the standardized normal distribution. For $z \geq 1.25$ the probability is 10.56% (50%-39.44%), and for $z = -0.05$ the probability is 0.3085. Therefore this investment has an 11.51% chance of obtaining a 15% or more return, and a 30.85% chance of obtaining 8% or less.

It is extremely difficult to make probabilistic statements when using the normal distribution. Given that the return data utilized are *ex post*, the predictive ability of the normal distribution based on historical data is limited. The past does not always predict the future; hence, this problem is somewhat alleviated by an assumption that the security under consideration is in a static state. Nevertheless, as shown by research and common sense, this assumption is rather bold, and the use of the normal distribution should be limited to comparisons of various past portfolio returns.

Another difficulty with the normal distribution is that if the distribution of the sample is skewed in any way, the standard deviation will not properly reflect the equivalent areas under the curve on either side of the mean. This problem will be discussed and solved in the next section through the application of the **log-normal distribution.** A variable is log-normally distributed if a logarithm of this variable is normally distributed. In addition, Simkowitz and Beedles (1978) have discussed the implications of skewness on the diversification of the portfolio.

23.4.2 The Log-Normal Distribution

The approach outlined in the last section for delineating the efficient frontier assumes that the security-return data are normally distributed. In reality, most financial researcher would agree that security-return data tend to be positively skewed. This skewness can be a serious problem in accurately developing the efficient frontier with the Markowitz model because of the assumption that only the first two moments of the return distribution, mean and variance, are important.

One reason for returns being positively skewed is the inability of the investor to lose more than 100% of his or her investment, effectively creating a lower bound to portfolio returns. This is called the limited-liability constraint. But since capital gains and dividends could conceivably be infinite, the upper tail of the distribution of returns has no upper bounds. The range of probable returns is spread toward the positive side and therefore contains a potential bias when it is utilized in developing statistical estimates.

If the return distributions are significantly skewed, the efficient frontier can be more accurately determined with the use of logarithmically transformed holding-period returns. That is, each holding-period return for security i with holding period $T(1 + R_{iT})$ is transformed by computing its natural logarithm, $\ln(1 + R_{iT})$. The logarithmic transformation converts a data set of discretely compounded returns into continuously compounded returns. The larger the differencing interval used to measure the returns, the more positively skewed the distribution of discrete time returns; that is, if skewness exists in the return distribution, annual data will be more positively skewed than monthly data, which will be more skewed than weekly data.

The continuous compounding implied in the logarithmically transformed data will virtually eliminate any positive skewness existing in the raw return data when rates of return are log-normally distributed. It is practically possible to directly derive the efficient frontier using means and variances of logarithmically transformed data. Under this circumstance, assume that utility functions use continuous returns and variances, and then use the $\ln(1 + R_p)$ transformation on discrete data. To deal with the untransformed log-normally distributed data, Elton et al. (1976) use the mean and variance of log-normal distributed to derive the efficient frontier. In other words, they first delineate the efficient frontier by using the means and variances of the untransformed data and then determine the subset of the $[E(1 + R_p), \sigma(1 + R_p)]$ efficient frontier in terms of log-transformed data, $\ln(1 + R_p)$.

Use of the logarithmic transformation can also be shown to eliminate less desirable portfolios in the lowest return segment of the efficient frontier computer using the raw data.[6] Before moving on to an example, however, it will be useful to briefly summarize the portfolio-selection process.

23.4.3 Mathematical Method to Calculate Efficient Frontier[7]

The Markowitz model of portfolio selection is a mathematical approach for deriving optimal portfolios, that is, portfolios that satisfy the following conditions.

The least risk for a given level of expected return (minimum-variance portfolios).

The greatest expected return for a given level of risk (efficient portfolios).

How does a portfolio manager apply these techniques in the real world?

The process would normally begin with a universe of securities available to the fund manager. These securities would be determined by the goals and objectives of the mutual fund. For example, a portfolio manager who runs a mutual fund specializing in health-care stocks would be required to select securities from the universe of health-care stocks. This would greatly reduce the analysis of the fund manager by limiting the number of securities available.

The next step in the process would be to determine the proportions of each security to be included in the portfolio. To do this, the fund manager would begin by setting a target rate of return for the portfolio. After determining the target rate of return, the fund manager can determine the different proportions of each security that will allow the portfolio to reach this target rate of return.

The final step in the process would be for the fund manager to find the portfolio with the lowest variance given the target rate of return.

Following Elton et al. (2014) and Francis and Kim (2013), the optimal portfolio can be obtained mathematically through the use of the Lagrangian multipliers. The Lagrangian method allows the minimization or maximization of an objective function when the objective function is subject to some constraints. To do mathematical analysis, we need to calculate the expected return and variance of the portfolio.

One of the goals of portfolio analysis is minimizing the risk or variance of the portfolio, subject to the portfolio's attaining some target expected rate of return, and also subject to the portfolio weights' summing to one. The problem can be stated mathematically:

$$\text{Min } \sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma_{ij} \tag{23.10}$$

Subject to

(i) $\sum_{i=1}^{n} W_i E(R_i) = E^*$,

where E^* is the target expected return and

(ii) $\sum_{i=1}^{n} W_i = 1.0$.

The first constraint simply says that the expected return on the portfolio should equal the target return determined by the portfolio manager. The second constraint says that the weights of the securities invested in the portfolio must sum to one.

The Lagrangian objective function can be written:

[6] See Baumol, W. J. "An Expected Gain-Confidence Limits Criterion for Portfolio Selection." *Management Science* (October 1963).

[7] We also provide graphical method to calculate efficient frontier. Please see Appendix 80A for detailed graphical method to calculate efficient frontier.

$$C = \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j \mathrm{Cov}(R_i R_j) + \lambda_1\left(1 - \sum_{i=1}^{n} W_i\right) + \lambda_2\left[E^* - \sum_{i=1}^{n} W_i E(R_i)\right]. \tag{23.11}$$

For three securities case, the Lagrangian objective function is as follow:

$$\begin{aligned} C = {} & W_1^2\sigma_1^2 + W_2^2\sigma_2^2 + W_3^2\sigma_3^2 + 2W_1W_2\sigma_{12} + 2W_1W_3\sigma_{13} + 2W_2W_3\sigma_{23} \\ & + \lambda_1(1 - W_1 - W_2 - W_3) + \lambda_2 E^* - W_1E(R_1) - W_2E(R_2) - W_3E(R_3) \end{aligned}$$

Taking the partial derivatives of (80.12) with respect to each of the variables, $W_1, W_2, W_3, \lambda_1, \lambda_2$ and setting the resulting five equations equal to zero yields the minimization of risk subject to the Lagrangian constraints. We can obtain the following equations.

$$\begin{aligned} \frac{\partial C}{\partial W_1} &= 2W_1\sigma_1^2 + 2W_2\sigma_{12} + 2W_3\sigma_{13} - \lambda_1 - \lambda_2 E(R_1) = 0 \\ \frac{\partial C}{\partial W_2} &= 2W_2\sigma_2^2 + 2W_1\sigma_{12} + 2W_3\sigma_{23} - \lambda_1 - \lambda_2 E(R_2) = 0 \\ \frac{\partial C}{\partial W_3} &= 2W_3\sigma_3^2 + 2W_1\sigma_{13} + 2W_2\sigma_{23} - \lambda_1 - \lambda_2 E(R_3) = 0 \end{aligned} \tag{23.12}$$

$$\frac{\partial C}{\partial \lambda_1} = 1 - W_1 - W_2 - W_3 = 0$$

$$\frac{\partial C}{\partial \lambda_2} = E^* - W_1E(R_1) - W_2E(R_2) - W_3E(R_3) = 0$$

This system of five equations and five unknowns can be solved by the use of matrix algebra. Briefly, the Jacobian matrix of these equations is

$$\begin{bmatrix} 2\sigma_{11} & 2\sigma_{12} & 2\sigma_{13} & -1 & -E(R_1) \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & -1 & -E(R_2) \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & -1 & -E(R_3) \\ 1 & 1 & 1 & 0 & 0 \\ E(R_1) & E(R_2) & E(R_3) & 0 & 0 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ E^* \end{bmatrix} \tag{23.13}$$

Therefore, it is possible to premultiply both sides of the matrix equation of (23.13), AW = K, by the inverse of A (denoted A^{-1}) and solve for the W column. This is possible because all values in the A and K matrices are known or arbitrarily set.

The first problem is the inversion of the matrix of coefficients. First, derive the determinant of matrix A. Next, after developing the signed, inverted matrix of cofactors for A, divide the cofactors by the determinant, resulting in the inversion of the original matrix. Finally, premultiply the column vector for the investment weights.

Example 23.4 This example focuses on the returns and risk of the first three industrial companies, International Business Machines Corp. (IBM), Johnson & Johnson (JNJ), and JP Morgan (JPM), for the period July 2013 to June 2018. The data used are tabulated in Table 23.3. The monthly price per share data used can be found in Appendix 23B.

Plugging the data listed in Table 23.3 and $E^* = 0.00193$ into the matrix above yields

$$\begin{bmatrix} 0.0451 & 0.0006 & 0.0014 & -1 & 0.0013 \\ 0.0019 & 0.0808 & 0.0019 & -1 & -0.0079 \\ 0.0014 & 0.0019 & 0.1163 & -1 & -0.0137 \\ 1 & 1 & 1 & 0 & 0 \\ -0.0013 & 0.0079 & 0.0137 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0.00193 \end{bmatrix} \tag{23.14}$$

When matrix A is properly inverted and post-multiplied by K, the solution vector $A^{-1}K$ is derived:

Table 23.3 Data for three securities

Company	$E(r_i)$	σ_i^2	$\text{Cov}(R_i, R_j)$
IBM	−0.0013	0.0005	$\sigma_{12} = 0.0003$
JNJ	0.0079	0.0016	$\sigma_{23} = 0.0009$
JPM	0.0137	0.0034	$\sigma_{13} = 0.0007$

$$\begin{matrix} W \\ \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \end{matrix} = \begin{matrix} A^{-1}K \\ \begin{bmatrix} 0.7014 \\ 0.2168 \\ 0.0818 \\ 0.0301 \\ -1.4005 \end{bmatrix} \end{matrix} \tag{23.15}$$

The solutions presented in Eq. 23.15 are obtained by Excel and are presented in Appendix 23A.

With the knowledge of the efficient portfolio weights given that $E(R_p)$ is equal to 0.7014, 0.2168, and 0.0818, the variances of the efficient portfolios may be derived from plugging the numbers into Eq. (23.2). Taking the square root of the variances to derive the standard deviation, the various risk–return combinations can be plotted and the efficient frontier graphed, as shown in Fig. 23.8 in the previous section. As can be seen from comparing this figure to the graphical derivation of the efficient frontier derived graphically, the results are almost the same.

Moreover, we can conduct the sensitive analysis by changing the value of our target rate of return. For example, if we set a higher target rate of return, we say, $E^* = 0.01$, then it can be calculated that the optimal weights would be as follows:

$$W_1 = 0.0759, W_2 = 0.4420, \text{ and } W_3 = 0.4820$$

When seeking a higher target rate of return, the weight on securities with higher expected returns would be higher. Furthermore, if we set the target rate of return even higher than that of the most profitable securities, for instance, $E^* = 0.015$, then it can be calculated that

$$W_1 = -0.3115, W_2 = 0.5816, \text{ and } W_3 = 0.7300$$

These results show that when the target rate of return is extremely high, we might even need to short the security with a very low expected rate of return.

In Appendix 23A, we use Johnson & Johnson (JNJ), IBM, and Caterpillar (CAT) data from Appendix 22D, we will show how the Excel program can be used to calculate optimal weights for these three companies. In addition, in this appendix, we also perform sensitivity analysis for optimal weights determination in terms of changing the target rate of return.

23.4.4 Portfolio Determination with Specific Adjustment for Short Selling

By using a definition of short sales developed by Lintner (1965), the computation procedure for the efficient frontier can be modified. Lintner defines short selling as putting up an amount of money equal to the value of the security sold short. Thus the short sale is a use rather than a source of funds to the short seller. The total funds the investor invests short, plus the funds invested long, must add up to the original investment. The proportion of funds invested short is $|X_i|$, since $X_i < 0$. The constraint in the minimization problem concerning the weights of the individual securities needs to be modified to incorporate this fact. Additionally, the final portfolio weight (output of the matrix inversion) must be rescaled so the sum of the absolute value of the weights equals one. By defining short sales in this manner, the efficient frontier does not extend to infinity (as shown in Fig. 23.8(a)) but resembles the efficient frontier in Fig. 23.8(b).

Monthly rates of return for JNJ, IBM, and BA are used to perform the analysis in this section. The sample period is from January 2001 to September 2007. The Markowitz model determines optimal asset allocation by minimizing portfolio variance using a constrained optimization procedure:

$$\text{Min Var}(R_p) = \sum_{i=1}^{3}\sum_{j=1}^{3} W_i W_j \sigma_{ij}. \tag{23.16}$$

Subject to

(i) $\sum_{i=1}^{3} W_i E(R_i) = E^*,$

(ii) $\sum_{i=1}^{3} |W_i| = 1.0,$

where the E^* is the investors' desired rate of return, and where the absolute value of the weights $|W_i|$ allows for a given W to be negative (sold short) but maintains the requirement that all funds are invested or their sum equals one.

The Lagrangian function is

$$\text{Min } L = \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j + \lambda_1 \sum_{i=1}^{n} [W_i E(R_i) - E^*] + \lambda_2 \left(\sum_{i=1}^{n} W_i - 1 \right). \tag{23.17}$$

By using IBM, JNJ, and JPM, a three-security portfolio will be formed and optimal weights will be solved for.

Company	$E(r_i)$	σ_i^2	$\text{Cov}(R_i, R_j)$
IBM	-0.0013	0.0005	σ_{12} = 0.0003
JNJ	0.0079	0.0016	σ_{23} = 0.0009
JPM	0.0137	0.0034	σ_{13} = 0.0007

Substituting these values into the matrix:

$$\begin{bmatrix} 0.0451 & 0.0006 & 0.0014 & 0.0013 & 1 \\ 0.0019 & 0.0808 & 0.0019 & -0.0079 & 1 \\ 0.0014 & 0.0019 & 0.1163 & -0.0137 & 1 \\ -0.0013 & 0.0079 & 0.0137 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E^* \\ 1 \end{bmatrix} \tag{23.18}$$

And we set $E^* = 0.022$.

By solving using the identity-matrix technique, the weights for the three securities can be obtained:

$$W_{IBM} = -0.8541$$
$$W_{JNJ} = 0.7769$$
$$W_{JPM} = 1.0772$$

By using the following relationship to rescale these weights so that the second constraint for the sum of the absolute values of the weights to equal one is satisfied:

$$W_{|A|} = \frac{|A|}{|A| + |B| + |C|} \tag{23.19}$$

The resealed absolute weights are

$$W_{IBM} = \frac{0.8541}{0.8541 + 0.7769 + 1.0772} = 0.3154$$
$$W_{JNJ} = \frac{0.7769}{0.8541 + 0.7769 + 1.0772} = 0.2869$$
$$W_{JPM} = \frac{1.0772}{0.8541 + 0.7769 + 1.0772} = 0.3978$$

and

$$W_{IBM} + W_{JNJ} + W_{JPM} = 1$$

The return on this portfolio is

$$\overline{R}_P = 0.3154 \times -0.0013 + 0.2869 \times 0.0079 + 0.3978 \times 0.0137 = 0.0073$$

The variance:

$$\begin{aligned}\sigma_P^2 &= (0.3154)^2 \times 0.0226 + (0.2869)^2 \times 0.0404 + (0.3978)^2 \times 0.0581 + 2 \times 0.3153 \times 0.2869 \\ &\quad \times 0.0003 + 2 \times 0.3154 \times 0.3978 \times 0.0007 + 2 \times 0.2869 \times 0.3978 \times 0.0009 \\ &= 0.0152\end{aligned}$$

23.4.5 Portfolio Determination Without Short Selling

The minimization problem under study can be modified to include the restriction of no short selling by adding a third constraint:

$$W_i \geq 0, \qquad i = 1, \ldots, N. \tag{23.20}$$

The addition of this non-negativity constraint precludes negative values for the weights (i.e., no short selling). The problem now is a quadratic programming problem similar to the ones solved so far, except that the optimal portfolio may fall in an unfeasible region. In this circumstance, the next best optimal portfolio is elected that meets all of the constraints. An example of this situation is shown in Fig. 23.9.[8]

In Fig. 23.9a the optimal portfolio is located in the region of positive values for weights W_i^*, and thereby satisfies the constraint $W_i^* \geq 0$. In Fig. 23.9b the optimal portfolio shown by W_i^* falls in a region where W_1 is negative, and so the constraint is not satisfied. The next best optimal portfolio is shown by B, and this is the solution of the problem that satisfies the non-negativity constraint. Most recently, Lewis (1988) has developed a simple algorithm for the portfolio-selection problem. His method is based on an interactive use of the Markowitz critical-line method for solving quadratic programs. In Fig. 23.9a, b the vertical axis is defined as $(\overline{R}_p - R_f)/\sigma_p$ which is a Sharpe performance mean can be regarded as the objective function for a portfolio selection.

23.5 Summary

This chapter has focused on the foundations of Markowitz's model and on derivation of efficient frontier through the creation of efficient portfolios of varying risk and return. It has been shown that an investor can increase expected utility through portfolio diversification as long as there is no perfect positive correlation among the component securities. The extent of the benefit increases as the correlation is lower, and also increases with the number of securities included.

The Markowitz model can be applied to develop the efficient frontier that delineates the optimal portfolios that match the greatest return with a given amount of risk. Also, this frontier shows the dominant portfolios as having the lowest risk given a stated return. This chapter has included methods of solving for the efficient frontier both graphically and through a combination of calculus and matrix algebra, with and without explicitly incorporating short selling.

Appendix 23A: Excel Program to Calculate the Optimal Weight For Johnson & Johnson, IBM, and Caterpillar

Step 1

Using Johnson & Johnson (JNJ), IBM, and Caterpillar (CAT) data from Appendix 22D, we have monthly expected returns, monthly standard deviation, and covariance between each company. There are 63 months from January 01, 2016 to March 01, 2021 in our sample.

[8] In Fig. 8.9(a) and (b), the vertical axis is defined as $(\overline{R}_p - R_f)/\sigma_p$, which is the Sharpe performance measure.

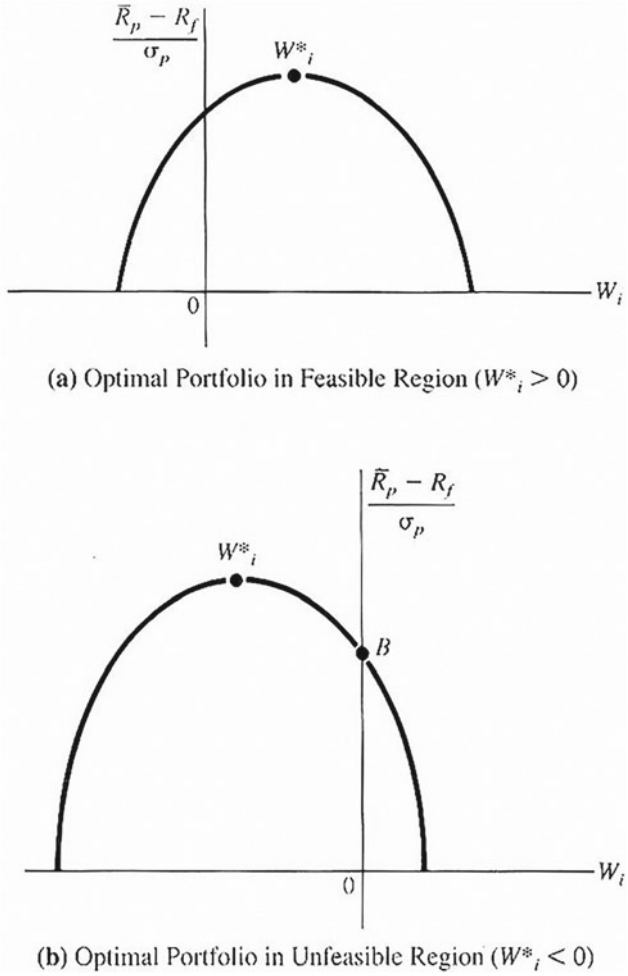


Fig. 23.9 Optimal portfolio in feasible and unfeasible regions

By using the information which was listed in Appendix 22D, we can get the table below.
Table. Data for Three Securities.

Company	$E(r_i)$	σ_i^2	$\mathrm{Cov}(R_i, R_j)$
IBM	0.0064	0.0051	$\sigma_{12} = 0.0015$
JNJ	0.0105	0.0024	$\sigma_{23} = 0.0009$
CAT	0.0236	0.0053	$\sigma_{13} = 0.0026$

Step 2

The minimum expected return within these three companies is IBM, and the value of the expected return for IBM company is 0.0064. The maximum expected return within these three companies is IBM, and the value of the expected return for CAT company is 0.0236. We can assume a desire expected return E* = 0.015.

Plugging the data listed in the above table and $E^* = 0.015$ into the matrix, we can get

$$\begin{bmatrix} 0.1427 & 0.0030 & 0.0052 & -1 & -0.0064 \\ 0.0030 & 0.0981 & 0.0018 & -1 & -0.0105 \\ 0.0052 & 0.0018 & 0.1456 & -1 & -0.0236 \\ 1 & 1 & 1 & 0 & 0 \\ 0.0064 & 0.0105 & 0.0236 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0.015 \end{bmatrix} \tag{23.21}$$

Step 3

From Eq. 23.13, we have

$$\begin{bmatrix} 2\sigma_{11} & 2\sigma_{12} & 2\sigma_{13} & -1 & -E(R_1) \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & -1 & -E(R_2) \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & -1 & -E(R_3) \\ 1 & 1 & 1 & 0 & 0 \\ E(R_1) & E(R_2) & E(R_3) & 0 & 0 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ E^* \end{bmatrix}$$

By using the matrix inversion of Eq. 23.13, we have

$$\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 2\sigma_{11} & 2\sigma_{12} & 2\sigma_{13} & -1 & -E(R_1) \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & -1 & -E(R_2) \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & -1 & -E(R_3) \\ 1 & 1 & 1 & 0 & 0 \\ E(R_1) & E(R_2) & E(R_3) & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ E^* \end{bmatrix}$$

Combine all information into excel calculation, we can get the efficient portfolio weights for each company.

	A	B	C	D	E	F	G	H	I
10									
11	Matrix A						Matrix K		Matrix W
12	0.14268	0.003075	0.0052	-1	-0.0064		0		W1
13	0.003075	0.098115	0.005107	-1	-0.01046		0		W2
14	0.005107	0.001757	0.145552	-1	-0.0236		0		W3
15	1	1	1	0	0		1		lambda 1
16	0.006401	0.010459	0.023596	0	0		0.015		lambda 2
17									
18									
19	Inverse A						Matrix W		
20	3.154249	-4.12871	0.974459	0.81632	-40.4087		0.21019		
21	-4.12871	5.404212	-1.2755	0.727683	-23.2315		0.37921		
22	0.974459	-1.2755	0.301045	-0.544	63.64017		0.4106		
23	-0.82124	-0.72125	0.542484	0.186466	-10.9118		0.022789		
24	41.19089	22.20762	-63.3985	-11.0274	844.5425		1.640729		

$$\begin{matrix} W \\ \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \end{matrix} = \begin{matrix} A^{-1}K \\ \begin{bmatrix} 0.2102 \\ 0.3792 \\ 0.4106 \\ 0.0228 \\ 1.9407 \end{bmatrix} \end{matrix} \tag{23.22}$$

From Eq. 23.22, we know the optimal weights for JNJ, IBM, and CAT are 0.2102, 0.3792, and 0.4106, respectively.

Moreover, we can conduct the sensitive analysis by changing the value of our target rate of return. For example, if we set a lower target rate of return, $E^* = 0.01$, then the optimal weights for W_1, W_2, and W_3 would be, 0.4122, 0.4954, *and* 0.0924, respectively. If we set the target rate of return higher, $E^* = 0.023$, then W_1, W_2, and W_3 will be −0.1131, 0.1934, and 0.9197, respectively.

The detailed Excel programs, which were used for this appendix, can be found in Lee et al. (2016). In the next appendix, we will show how graphical analysis can be used to calculate optimal weights for IBM, JNJ, and JPM.

Appendix 23B: Graphical Analysis in Markowitz Portfolio-Selection Model: Three-Security Empirical Solution

The process of finding the efficient frontier graphically described in this appendix is originally developed by Markowitz (1952). Alexander and Francis (1986) have discussed this subject in detail. To facilitate a realistic example, actual data have been taken from a set of monthly returns of IBM, JNJ, and JPM from July 2013 =to June 2018. The data used are tabulated in Table 23.4. Using historical data, we can obtain the expected monthly returns, variances, and covariances of IBM, JNJ, and JPM as indicated in Table 23.3. Both graphical and mathematical analyses are employed to obtain an empirical solution.

Table 23.4 Adjusted prices for IBM, JNJ, and JPM (July 2013–June 2018)

Date	IBM	JNJ	JPM	Date	IBM	JNJ	JPM
2013/7/1	158.0662	75.43977	45.59301	2016/2/1	111.0931	97.57764	55.08128
2013/8/1	161.8096	81.65747	49.46898	2016/3/1	120.8047	100.9686	55.39945
2013/9/3	152.7772	75.82388	44.73557	2016/4/1	137.1224	102.8238	56.02644
2013/10/1	154.787	76.74516	45.46177	2016/5/2	130.6044	106.1762	60.13674
2013/11/1	148.849	81.92174	46.28145	2016/6/1	139.7651	106.9627	61.92793
2013/12/2	148.1872	83.29762	50.22123	2016/7/1	139.6185	115.0338	58.2012
2014/1/2	154.9085	80.42622	51.6409	2016/8/1	147.958	118.9318	60.61437
2014/2/3	144.3631	76.67128	48.18102	2016/9/1	147.4704	113.6912	63.85411
2014/3/3	154.6915	81.47659	49.8666	2016/10/3	145.6864	113.4334	63.18907
2014/4/1	163.2883	87.15396	53.82328	2016/11/1	141.231	110.1204	66.00257
2014/5/1	162.474	89.45874	49.74346	2016/12/1	149.0694	107.0745	78.271
2014/6/2	156.7997	91.55177	49.41315	2017/1/3	155.9436	111.3621	83.47694
2014/7/1	157.3571	94.86723	51.7543	2017/2/1	162.5661	108.853	81.74485
2014/8/1	159.7214	89.51769	50.77442	2017/3/1	171.0527	119.8599	90.06849
2014/9/2	162.7131	93.24455	53.64217	2017/4/3	164.0488	120.6631	84.21787
2014/10/1	158.9841	94.09257	53.73206	2017/5/1	149.3268	119.3567	84.74188
2014/11/3	139.6091	96.94329	55.09868	2017/6/1	144.9284	125.4475	80.8484
2014/12/1	138.1529	98.09224	54.30224	2017/7/3	147.6908	129.4608	90.77698
2015/1/2	138.5976	94.90512	56.91961	2017/8/1	137.9321	129.0809	91.05102
2015/2/2	132.2689	91.55457	50.52538	2017/9/1	138.2191	128.4512	89.74931
2015/3/2	138.2091	94.3808	56.26378	2017/10/2	140.6941	128.6375	94.77997
2015/4/1	137.0895	90.65933	54.96896	2017/11/1	147.7643	137.225	99.35055
2015/5/1	149.5686	91.55539	58.32488	2017/12/1	149.9486	138.0654	103.1604
2015/6/1	148.8097	92.14014	60.58966	2018/1/2	149.4544	137.3256	106.2712
2015/7/1	143.8343	90.69412	62.82223	2018/2/1	157.351	138.1048	115.6519
2015/8/3	138.7801	92.12171	63.24679	2018/3/1	150.4954	126.343	112.2477
2015/9/1	125.7988	85.46677	56.71261	2018/4/2	146.836	123.643	106.7259
2015/10/1	126.6011	86.46911	56.68491	2018/5/1	141.8753	125.0823	108.1921
2015/11/2	123.7621	94.82182	60.92373	2018/6/1	140.4328	121.26	107.8141
2015/12/1	125.7328	95.69846	62.84793	2018/6/29	138.2068	121.34	103.6368
2016/1/4	120.9894	93.9408	59.53569				

Graphical Analysis. To begin to develop the efficient frontier graphically, it is necessary to move from the three dimensions necessitated by the three-security portfolio to a two-dimensional problem by transforming the third security into an implicit solution from the other two. To do this it must be noted that since the summation of the weights of the three securities is equal to unity, then implicitly:

$$W_3 = 1 - W_1 - W_2 \tag{23.23}$$

Additionally, the above relation may be substituted into Eq. (23.23):

$$\begin{aligned} E(R_p) &= W_1E(R_1) + W_2E(R_2) + W_3E(R_3) \\ &= W_1E(R_1) + W_2E(R_2) + (1 - W_1 - W_2)E(R_3) \\ &= W_1E(R_1) + W_2E(R_2) + E(R_3) - W_1E(R_3) - W_2E(R_3) \\ &= [E(R_1) - E(R_3)]W_1 + [E(R_2) - E(R_3)]W_2 + E(R_3) \end{aligned} \tag{23.24}$$

Finally, inserting the values for the first and second securities yields

$$\begin{aligned} E(R_p) &= (-0.0013 - 0.0147)W_1 + (0.0079 - 0.0147)W_2 + 0.0147 \\ &= -0.016W_1 - 0.0068W_2 + 0.0147 \end{aligned} \tag{23.25}$$

As can be seen, Eq. (23.25) is a linear function in two variables and as such is readily graphable. Since given a certain level of portfolio return the function will solve jointly for the weights of securities 1 and 2, it solves indirectly for the weight of security 3.

The variance formula shown in Eq. (23.2) is converted in a similar manner by substituting in Eq. (23.23) as follows:

$$\begin{aligned} \sigma_p^2 &= \sum_{i=1}^{3}\sum_{j=1}^{3} W_iW_jCov(R_i, R_j) = Var(R_p) \\ &= W_1^2\sigma_{11} + W_2^2\sigma_{22} + W_3^2\sigma_{33} + 2W_1W_2\sigma_{12} + 2W_1W_3\sigma_{13} \\ &+ 2W_2W_3\sigma_{23} \\ &= W_1^2\sigma_{11} + W_2^2\sigma_{22} + (1 - W_1 - W_2)^2\sigma_{33} + 2W_1W_2\sigma_{12} \\ &+ 2W_1(1 - W_1 - W_2)\sigma_{13} + 2W_2(1 - W_1 - W_2)\sigma_{23} \\ &= (\sigma_{11} + \sigma_{33} - 2\sigma_{13})W_1^2 + (2\sigma_{33} + 2\sigma_{12} - 2\sigma_{13} - 2\sigma_{23})W_1W_2 \\ &+ (\sigma_{22} + \sigma_{33} - 2\sigma_{23})W_2^2 + (-2\sigma_{33} + 2\sigma_{13})W_1 \\ &+ (-2\sigma_{33} + 2\sigma_{13})W_2 + \sigma_{33} \end{aligned} \tag{23.26}$$

Inserting the covariances and variances of the three securities from Table 23.3:

$$\begin{aligned} \sigma_P^2 &= [0.0226 + 0.0581 - 2(0.0007)]W_1^2 \\ &+ [2(0.0581) + 2(0.0003) - 2(0.0007) - 2(0.0009)]W_1W_2 \\ &+ [0.0404 + 0.0581 - 2(0.0009)]W_2^2 \\ &+ [-2(0.0581) + 2(0.0007)]W_1 \\ &+ [-2(0.0581) + 2(0.0007)]W_2 + 0.0581 \\ &= 0.0793W_1^2 + 0.1136W_1W_2 + 0.0967W_2^2 - 0.1148W_1 \\ &- 0.1148W_2 + 0.0581 \end{aligned} \tag{23.27}$$

Minimum-Risk Portfolio. Part of the graphical solution is the determination of the minimum-risk portfolio. Standard partial derivatives are taken from Eq. (23.26) with respect to the directly solved weight factors as follows:

$$\begin{aligned}\frac{\partial\sigma_p^2}{\partial W_1} &= 2(\sigma_{11}+\sigma_{33}-2\sigma_{13})W_1+(2\sigma_{33}+2\sigma_{12}-2\sigma_{13}-2\sigma_{23})W_2\\ &\quad +(-2\sigma_{33}+2\sigma_{13})=0\\ \frac{\partial\sigma_p^2}{\partial W_2} &= (2\sigma_{33}+2\sigma_{12}-2\sigma_{13}-2\sigma_{23})W_1+2(\sigma_{22}+\sigma_{33}-2\sigma_{23})W_2\\ &\quad +(2\sigma_{23}-2\sigma_{33})=0\end{aligned} \tag{23.28}$$

When these two partial derivatives are set equal to zero and the unknown weight factors are solved, the minimum-risk portfolio is derived. Using the numeric values from Table 23.3:

$$\begin{aligned}\frac{\partial\sigma_P^2}{\partial W_1} &= 2\times[0.0226+0.0581-2\times0.0007]W_1\\ &\quad +(2\times0.0581+2\times0.0003-2\times0.0007-2\times0.0009)W_2\\ &\quad +(-2\times0.0581+2\times0.0007)\\ &\quad =0.1586W_1+0.1136W_2-0.1148=0\\ \frac{\partial\sigma_P^2}{\partial W_2} &= (2\times0.0581+2\times0.0003-2\times0.0007-2\times0.0009)W_1\\ &\quad +2(0.0404+0.0581-2\times0.0009)W_2\\ &\quad +(2\times0.0009-2\times0.0581)\\ &\quad =0.1136W_1+0.1934W_2-0.1144=0\end{aligned} \tag{23.29}$$

By solving these two equations simultaneously the weights of the minimum-risk portfolio are derived. This variance represents the lowest possible portfolio variance level achievable, given variance and covariance data for these stocks. This can be represented by the point V of Fig. 23.6. This solution is an algebraic exercise that yields $W_1 = 1.2176$ and $W_2 = 0.6894$ and therefore, through Eq. (23.23), $W_3 = -0.907$.

The Iso-Expected Return Line. The variance and return equations have been derived and it is now time to complete the graphing procedure. To begin the graphing, the iso-expected return function lines must be delineated given various levels of expected return. The **iso-expected return line** is a line that has the same expected return on every point of the line. Utilizing Eq. (23.25), three arbitrary returns are specified: 0.0082, 0.01008, 0.01134. These three monthly returns are then set equal to Eq. (23.25), and graphing is now possible by setting W_1 equal to zero or other values to solve for W_2. For example, if $W_2 = 0$ and $E(R_p) = 0.0080$, then W_1 can be solved as follows (see Table 23.5 for final results).

$$\begin{aligned}0.008 &= -0.0033W_1-0.0063(0)+0.0113\\ 0.0033W_1 &= 0.0113-0.008=0.0033\\ W_1 &= 1.0000\end{aligned}$$

In a similar fashion, other points are calculated and are listed in Table 23.5. A line may be drawn between the points to develop the various iso-expected return function lines. There is a multitude of possible return functions, but only these three lines (IR_1, IR_2, and IR_3) in terms of the data listed in Table 23.5 are shown in Fig. 23.10. In Fig. 23.10, vertical axis and horizontal axis represent W_1 and W_2, respectively.

Table 23.5 Iso-return lines

W_2	Target Return		
	0.008 W_1	0.010 W_1	0.012 W_1
–1.0	2.9091	2.3030	1.6970
–0.5	1.9545	1.3485	0.7424
0.0	1.0000	0.3939	−0.2121
0.5	0.0455	−0.5606	−1.1667
1.0	−0.9091	−1.5152	−2.1212

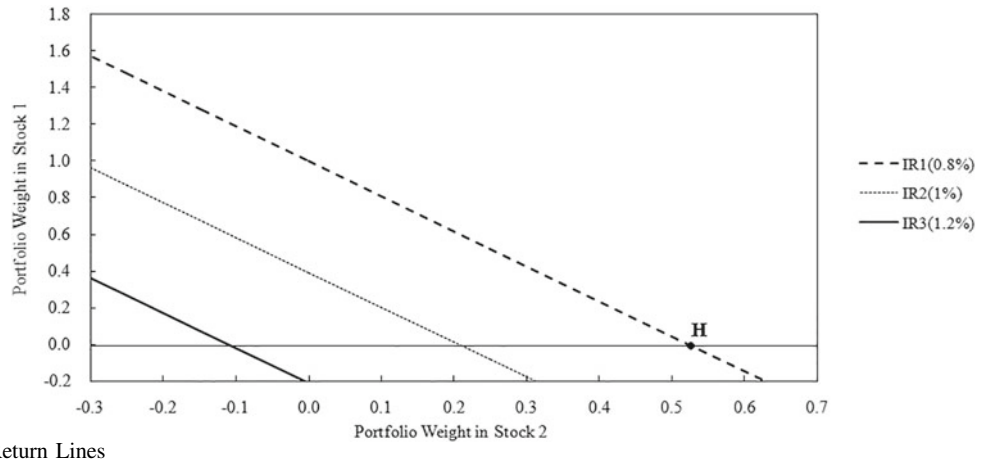


Fig. 23.10 Iso-Return Lines

Table 23.6 Portfolio variance along the iso-return line

W_2	0.0082		0.0100		0.0120	
	W_1	Var	W_1	Var	W_1	Var
−1.0000	2.9091	0.0284	2.3030	0.0173	1.6970	0.0130
−0.7500	2.4318	0.0184	1.8258	0.0103	1.2197	0.0092
−0.5000	1.9545	0.0107	1.3485	0.0058	0.7424	0.0077
−0.4713	1.8998	0.0100	1.2937	0.0054	0.6876	0.0077*
−0.2500	1.4773	0.0054	0.8712	0.0036	0.2652	0.0086
−0.1455	1.2778	0.0039	0.6717	0.0034*	0.0657	0.0097
0.0000	1.0000	0.0025	0.3939	0.0038	−0.2121	0.0119
0.1803	0.6558	0.0019*	0.0497	0.0054	−0.5563	0.0158
0.2500	0.5227	0.0020	−0.0833	0.0063	−0.6894	0.0176
0.5000	0.0455	0.0038	−0.5606	0.0113	−1.1667	0.0257
0.7500	−0.4318	0.0081	−1.0379	0.0187	−1.6439	0.0361
1.0000	−0.9091	0.0147	−1.5152	0.0284	−2.1212	0.0490

* *Note* Underlined variances indicate minimum-variance portfolios

Each point on the iso-expected return line of Fig. 23.10 represents a different combination of weights placed in the three securities. The issue now is to determine the portfolio that lies on the iso-expected return line with the lowest variance. Table 23.6 shows the portfolio variances as we move along the iso-expected return line. For example, moving along the iso-expected return line associated with an expected return of 0.80%, note that the minimum-variance portfolio is associated with weights of 0.1803 in security 2 and 0.6558 in security 1.

Iso-Variance Ellipses. As is seen from Table 23.6 the minimum-variance portfolio can be found by moving along the iso-expected return line until the minimum-variance portfolio is reached. To better visualize this, examine the variance of a portfolio as the proportions in security 1 and security 2 are changed. A plot of these points will allow the family of iso-variance ellipses to be traced. The **iso-variance ellipses** are the ellipses that have the same variance on every point of this ellipse curve. It should be noted that an ellipse is an egg-shaped circle of points with a common center and orientation. The minimum-risk portfolio variance is the center of all possible ellipses as it has the least risk. It will be desirable, then, to find this minimum-risk value, as no other weighting scheme of these three securities will develop a lesser risk. The solutions of Eq. (23.29). have determined that the weights associated with the minimum-risk portfolio are $W_1 = 0.3857$, $W_2 = 0.2810$, and $W_3 = 0.3333$. Substituting this information listed in Table 23.4 into Eq. (23.2), the variance of the minimum-risk portfolio is W_1.

$$\begin{aligned}\text{Var}(R_p) &= \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j \text{Cov}(R_i R_j) \\ &= W_1^2\sigma_{11} + W_2^2\sigma_{22} + W_3^2\sigma_{33} + 2W_1W_2\sigma_{12} + 2W_2W_3\sigma_{23} \\ &+ 2W_1W_3\sigma_{13} \\ &= (0.6558)^2(0.0025) + (0.1803)^2(0.0071) + (0.1639)^2(0.0083) \\ &+ 2(0.6558)(0.1803)(0.0007) + 2(0.1803)(0.1639)(0.0006) \\ &+ 2(0.6558)(0.1639)(0.0007) \\ &= 0.0019\end{aligned}$$

Note that the variance of the minimum-risk portfolio can be used as a base for graphing the iso-variance ellipses. It can be completed by taking Eq. (23.25) and holding one of the weights, say W_2 portfolio variance $\text{Var}(R_p)$, constant. Bring the $\text{Var}(R_p)$ to the right-hand side of the equation, and notice that the equation is of a quadratic form and can be solved using the quadratic formula:

$$W_1 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{23.30}$$

where

$$\begin{aligned}a &= \text{all coefficients of } W_1^2; \\ b &= \text{all coefficients of } W_1; \text{and} \\ c &= \text{all coefficients that are not multiplied by } W_1, \text{ or } W_1^2 : \text{or} \\ a &= \sigma_{11} + \sigma_{33} - 2\sigma_{13}; \\ b &= (2\sigma_{33} + 2\sigma_{12} - 2\sigma_{13} - 2\sigma_{23})W_2 - 2\sigma_{33} + 2\sigma_{13}; \text{and} \\ c &= (\sigma_{22} + \sigma_{33} - 2\sigma_{23})W_2^2 + (-2\sigma_{33} + 2\sigma_{23})W_2 + \sigma_{33} - \text{Var}(R_p).\end{aligned}$$

Substituting the numbers from the data of Table 23.4 into Eq. (23.26) yields

$$\begin{aligned}\text{Var}\ (R_p) &= 0.0094W_1^2 + 0.0154W_1W_2 + 0.0142W_2^2 - 0.0152W_1 \\ &\quad - 0.0152W_2 + 0.0083 \\ 0 &= 0.0094W_1^2 + 0.0154W_1W_2 + 0.0142W_2^2 - 0.0152W_1 \\ &\quad - 0.0152W_2 + 0.0083 - \text{Var}(R_p)\end{aligned}$$

where

$$\begin{aligned}a &= 0.0094; \\ b &= 0.0154W_2 - 0.0152; \text{and} \\ c &= 0.0142W_2^2 - 0.0152W_2 + 0.0083 - \text{Var}\ (R_p).\end{aligned}$$

When these expressions are plugged into the quadratic formula:

$$W_1 = \frac{-(0.0154W_2 - 0.0152) \pm \sqrt{b^2 - 4ac}}{2(0.0094)} \tag{23.31}$$

where

$$\begin{aligned}b^2 &= (0.0154W_2 - 0.0152)^2 \\ 4ac &= 4\{(0.0094)[0.0142W_2^2 - 0.0152W_2 + 0.0083 - \text{Var}(R_p)]\}\end{aligned}$$

This is a solution for the two points of W_1(W_{11} and W_{12}) for a given portfolio variance and weight of the second security (W_2). When selecting the level of $\text{Var}(R_p)$ *it is best to choose a value that is slightly larger than the minimum-risk value.*

This assures the calculation of a possible portfolio, since no portfolio of these three securities may have less risk. Additionally, an initial selection of a value for W_2 close to the minimum-risk portfolio W_2 will be desirable.

In sum, Eq. (23.31) can be used to construct the iso-variance ellipse for given $\text{Var}(R_p)$ in terms of arbitrary W_2. By jointly considering iso-expected return lines and iso-variance ellipses, efficient portfolios can be identified that are defined as the portfolios with the minimum-variance given the expected rate of return. This task can be done by computer, as mentioned in the last section. To obtain the weights of an efficient portfolio, the following set of instructions must be entered into the computer.

Find the portfolio weights that minimize portfolio variance, subject to the target expected rate of return constraint. For this case, target rates of return are 0.82, 1.008, or 1.1134%. The sum of the portfolio weights for all stocks in the portfolio must be equal to one. Mathematically, this instruction is defined in Eq. (23.24).

The expected return and variance for a three-security portfolio is defined in Eqs. (23.25) and (23.27), respectively.

The estimated expected return, variance, and covariance as defined in Table 23.6 should be entered into the computer for estimation.

Using $E(R_p) = 0.80\%$ as an example, $W_1 = 0$ and $W_2 = 0.5238$. From the relationship, $W_3 = 1 - W_1 - W_2 = 1 - 0 - 0.5238 = 0.4762$. In other words, the point H represents three portfolio weights for the three stocks. The computer substitutes weights, variances, and covariances (listed in Table 23.4) into Eq. (23.27) to obtain portfolio variance. The portfolio variance consistent with the weights of point H is computed. The computer now moves by some predetermined distance either northeast or southwest along the 0.80% iso-expected return line. From this kind of search, the minimum-variance associated with this 0.82% iso-expected return line is identified to be 0.0019, as indicated in Table 23.6. By a similar procedure, the minimum variances associated with 1 and 1.20% iso-expected return lines are identified to be 0.0034 and 0.0077, respectively. When minimum variances associated with predefined iso-expected return are identified, the optimal weights associated with expected returns equal to 1% and 1.20% in terms of data indicated in Table 23.4 are also calculated. These results are indicated in Table 23.6.

Using the minimum variances associated with iso-expected returns (0.80%, 1%, and 1.20%), three iso-variance ellipses can be drawn as $IV_1, IV_2, and\ IV_3$.

When $\text{Var}(R_p) = 0.0182$ *and* $W_2 = 0.3$, the quadratic formula from Eq. (23.31) above yields two roots.

$$W_{11} = \frac{-(0.0730) + \sqrt{(0.0730)^2 - (-0.0052)}}{2(0.0972)}$$
$$= 0.4247$$
$$W_{12} = \frac{-(0.0730) - \sqrt{(0.0730)^2 - (-0.0052)}}{2(0.0972)}$$
$$= 0.3263$$

To solve for W_{11} and W_{12}, we need $b^2 > 4ac$. Solving for the variance:

$$(0.104W_2 - 0.1042)^2 - 4\{(0.0972) > [0.1119W_2^2 - 0.103W_2 + 0.0525 - Var(R_p)]\}.$$

If W_2 and $\text{Var}(R_p)$ are not selected so that $b^2 > 4ac$, Eq. (23.31) will require taking the square root of a negative number, and mathematically the solution will be an imaginary number. Solving W_2 in terms of an imaginary number is not meaningful; therefore, we only consider the case when $b^2 > 4ac$.

The variance chosen was the minimum-variance of the portfolio associated with $E(R_p) = 0.0082$. Actually, any variance might have realistically been chosen as long as it exceeded the minimum-risk portfolio variance. Table 23.7 lists various W_{11} and W_{12} values for given levels of W_2 and $\text{Var}(R_p)$. In addition, Table 23.7 presents the value of b^2 and $4ac$ to check whether or not the root is a real number. It should be noticed that all possible variances are higher than the minimum-risk portfolio variance. Data from Table 23.7 are used to draw three iso-variance ellipses, as indicated in Figs. 23.11 and 23.12.

Table 23.7 Various W_1 given W_2 and Var(R_p)

W_2	Var(R_p)	b	b^2	C	4ac	W_{11}	W_{12}
0.14	0.0019	− 0.013044	0.000170	0.004522	0.000170	0.711205	0.676454
0.16	0.0019	− 0.012736	0.000162	0.004300	0.000162	0.716669	0.638223
0.18	0.0019	− 0.012428	0.000154	0.004088	0.000154	0.706925	0.615203
0.20	0.0019	− 0.012120	0.000147	0.003888	0.000146	0.689362	0.600000
0.22	0.0019	− 0.011812	0.000140	0.003699	0.000139	0.663195	0.593401
− 0.20	0.0034	− 0.018280	0.000334	0.008548	0.000321	1.162299	0.782382
0.00	0.0034	− 0.015200	0.000231	0.004900	0.000184	1.172396	0.444625
0.20	0.0034	− 0.012120	0.000147	0.002388	0.000090	1.046640	0.242722
0.40	0.0034	− 0.009040	0.000082	0.001012	0.000038	0.832359	0.129343
0.60	0.0034	− 0.005960	0.000036	0.000772	0.000029	0.452575	0.181467
− 0.6	0.0077	− 0.024440	0.000597	0.014952	0.000562	1.615217	0.984783
− 0.2	0.0077	− 0.018280	0.000334	0.004248	0.000160	1.674858	0.269823
0.2	0.0077	− 0.012120	0.000147	− 0.001912	− 0.000072	1.431458	− 0.142096
0.6	0.0077	− 0.005960	0.000036	− 0.003528	− 0.000133	1.006820	− 0.372777
1.0	0.0077	0.000200	0.000000	− 0.000600	− 0.000023	0.242231	− 0.263508

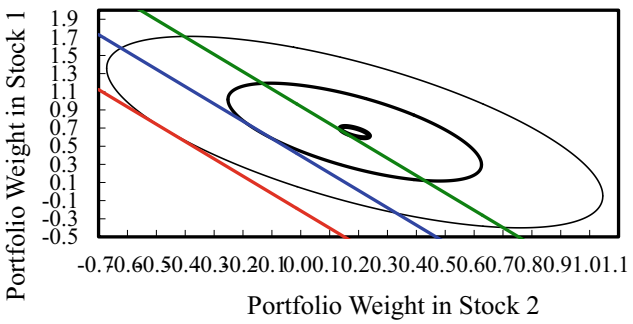


Fig. 23.11 Iso-variance ellipses

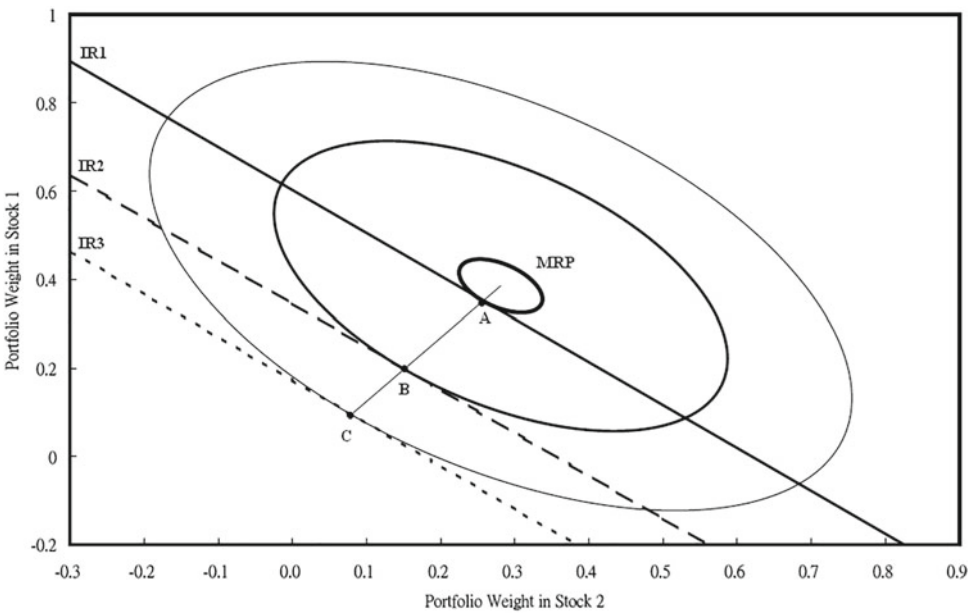


Fig. 23.12 Iso-variance ellipses and iso-return lines

Table 23.8 Weights, $\overline{R}_p$, and σ_p for efficient points

Portfolio*	W_1	W_2	W_3	$E(R_p)$	$Var(R_p)$	$\sigma(R_p)$
A	0.3521	0.2577	0.3902	0.0082	0.0182	0.1349
B	0.1983	0.1510	0.0795	0.01008	0.0258	0.1606
C	0.3902	0.6507	0.8252	0.01134	0.0368	0.1918

*Portfolios A, B, and C represent expected returns of 0.82%, 1.008%, and 1.134%, respectively

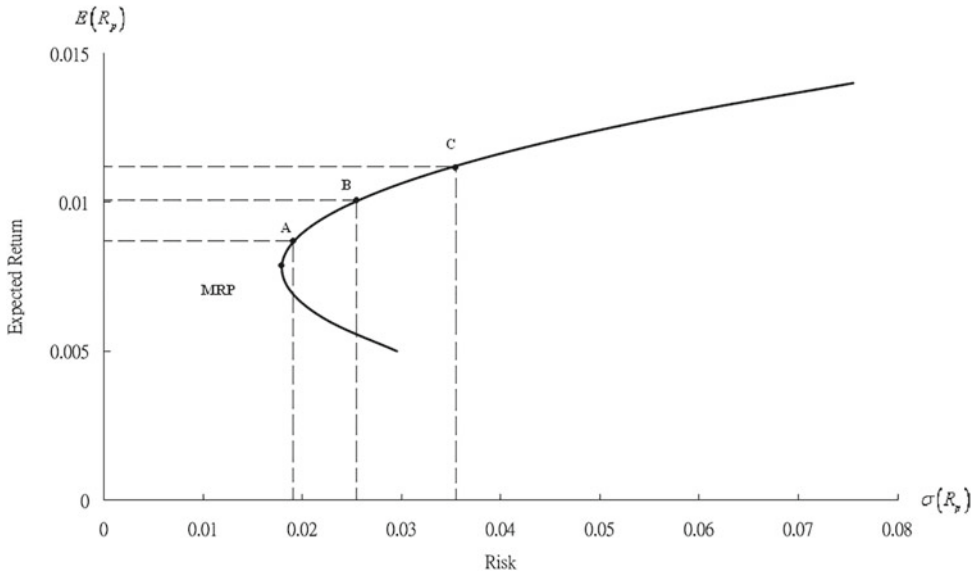


Fig. 23.13 The efficient frontier for the data of Table 23.8

The Critical Line and Efficient Frontier. After the iso-expected return functions and iso-variance ellipses have been plotted, it is an easy task to delineate the efficient frontier. By definition, the efficient portfolio is the portfolio with the highest return for any given risk. In Fig. 23.12, the efficient portfolios are those where a given iso-expected return line is just tangent to the variance ellipse. *MRPABC* is denoted as the **critical line**; all portfolios that lie between points *MRP* and *C* are said to be efficient, and the weights of these portfolios may be read directly from the graph. From the graph, portfolio weights for the portfolios that minimize variances, given a 0.80, 1, or 1.20% expected rate of return are listed in Table 23.6. It is possible, given these various weights, to calculate the $E(R_p)$ and the variances of these portfolios, as indicated in Table 23.8. The efficient frontier is then developed by plotting each risk–return combination, as shown in Fig. 23.13.

In this section, a step-wise graphical approach has been employed to obtain optimal weights for a three-security portfolio. In the next section, a mathematical optimization approach is used to calculate the optimal weight for a three-security portfolio.

Bibliogrphy

Alexander, G. J. and J. C. Francis. *Portfolio Analysis.* New York: Prentice-Hall, Inc., 1986.
Baumol, W. J. "An Expected Gain-Confidence Limit Criterion for Portfolio Selection." *Management Science*, v. 10 (October 1963), pp. 171–182.
Bertsekas, D. "Necessary and Sufficient Conditions for Existence of an Optimal Portfolio." *Journal of Economic Theory*, v. 8 (June 1974), pp. 235–247.
Black, F. "Capital Market Equilibrium with Restricted Borrowing." *Journal of Business*, v. 45 (July 1972), pp. 444–455.
Blume, M. "Portfolio Theory: A Step toward Its Practical Application." *Journal of Business*, v. 43 (April 1970), pp. 152–173.
Bodie, Z., A. Kane and A. Marcus. *Investments*, 12th ed. New York: McGraw-Hill Book Company, 2021.
Brealey, R. A. and S. D. Hodges. "Playing with Portfolios." *Journal of Finance*, v. 30 (March 1975), pp. 125–134.
Breen, W. and R. Jackson. "An Efficient Algorithm for Solving Large-Scale Portfolio Problems." *Journal of Financial and Quantitative Analysis*, v. 6 (January 1971), pp. 627–637.
Brennan, M. J. "The Optimal Number of Securities in a Risky Asset Portfolio Where There are Fixed Costs of Transaction: Theory and Some Empirical Results." *Journal of Financial and Quantitative Analysis*, v. 10 (September 1975), pp. 483–496.

Cohen, K. and J. Pogue. "An Empirical Evaluation of Alter native Portfolio-Selection Models." *Journal off Business*, v. 46 (April 1967), pp. 166–193.
Dyl, E. A. "Negative Betas: The Attractions of Selling Short." *Journal of Portfolio Management*, v. I (Spring 1975), pp. 74–76.
Elton, E. J. and Martin Gruber. "Portfolio Theory When Investment Relatives are Log Normally Distributed." *Journal of Finance*, v. 29 (September 1974), pp. 1265–1273.
Elton, E. J. and M. E. Padberg. "Simple Criteria for Optimal Portfolio Selection." *Journal of Finance*, v.11 (December 1976), pp. 1341–1357.
Elton, E. J. and M. E. Padberg. "Simple Criteria for Optimal Portfolio Selection: Tracing Out the Efficient Frontier." *Journal of Finance*, v. 13 (March 1978), pp. 296–302.
Elton, E. J., M. J. Gruber, S. J. Brown and W. N. Goetzmann. *Modern Portfolio Theory and Investment Analysis*, 9th ed. New York: John Wiley & Sons, 2014.
Evans, J. and S. Archer. "Diversification and the Reduction of Dispersion: An Empirical Analysis." *Journal of Finance*, v. 3 (December 1968), pp. 761–767.
Fama, E. F. "Efficient Capital Markets: A Review of Theory and Empirical Work." *Journal of Finance*, v. 25 (May 1970), pp. 383–417.
Feller, W. *An Introduction to Probability Theory and Its Application*, Vol. 1. New York: John Wiley and Sons, Inc., 1968.
Francis, J. C. and S. H. Archer. *Portfolio Analysis.* New York: Prentice-Hall, Inc., 1979.
Francis, J.C. and D. Kim. *Modern Portfolio Theory: Foundations, Analysis, and New Developments*, 1st Edition. New York: John Wiley and Sons, Inc., 2013.
Gressis, N., G. Philiippatos and J. Hayya. "Multiperiod Portfolio Analysis and the Inefficiencies of the Market Portfolio." *Journal of Finance*, v. 31 (September 1976), pp. 1115–1126.
Harel, A., J.C. Francis, and G. Harpaz. `Alternative utility functions: review, analysis and comparison.' *Review of Quantitative Finance and Accounting*, v. 51 (October 2018), pp. 1-27.
Henderson, J. and R. Quandt. *Microeconomic Theory: A Mathematical Approach,* 3rd ed. New York: McGraw-Hill, 1980.
Lee, C. F. and A. C. Lee. *Encyclopedia of Finance*. 2nd ed. New York: Springer, 2013.
Lee, C. F., J. C. Lee and A. C. Lee. *Statistics for Business and Financial Economics*. Singapore: World Scientific Publishing Co., 2000.
Lee, C. F., A. C. Lee and J. C. Lee . *Handbook of Quantitative Finance and Risk Management*. New York: Springer, 2010.
Lee, C.F., J. Finnerty, J. Lee, A.C. Lee, and D. Wort. *Security Analysis, Portfolio Management, and Financial Derivatives*. Singapore: World Scientific, 2013.
Lee, C. F, J. C. Lee, J. R. Chang, and T. Tai, *Essentials of Excel, Excel VBA, SAS and Minitab for Statistical and Financial Analyses*, Springer International Publishing, 2016.
Levy, H. and M. Sarnat. "A Note on Portfolio Selection and Investors' Wealth." *Journal of Financial and Quantitative Analysis*, v. 6 (January 1971), pp. 639–642.
Lewis, A. L. "A Simple Algorithm for the Portfolio Selection Problem." *Journal of Finance*, v. 43 (March 1988), pp. 71–82.
Lintner, J. "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolio and Capital Budgets." *Review of Economics and Statistics*, v. 47 (February 1965), pp. 13–27.
Maginn, J. L., D. L. Tuttle, J. E. Pinto and D. W. McLeavey. *Managing Investment Portfolios: A Dynamic Process*, CFA Institute Investment Series, 3rd ed. New York: John Wiley & Sons, 2007.
Mao, J. C. F. *Quantitative Analysis of Financial Decisions.* New York: Macmillan, 1969.
Markowitz, H. M. "Markowitz Revisited." *Financial Analysts Journal*, v. 32 (September/October 1976), pp. 47–52.
Markowitz, H. M. Mean-Variance Analysis in Portfolio Choice and Capital Markets. New York: Blackwell, 1987.
Markowitz, H. M. *Portfolio Selection.* Cowles Foundation Monograph 16. New York: John Wiley and Sons, Inc., 1959.
Markowitz, H. M. "Portfolio Selection." *Journal of Finance*, v. 1 (December 1952), pp. 77–91.
Martin, A. D., Jr. "Mathematical Programming of Portfolio Selections." Management Science, v. 1 (1955), pp. 152–166.
Merton, R. "An Analytical Derivation of Efficient Portfolio Frontier." *Journal of Financial and Quantitative Analysis*, v. 7 (September 1972), pp. 1851–1872.
Mossin, J. "Optimal Multiperiod Portfolio Policies." *Journal of Business*, v.41 (April 1968), pp. 215–229.
Ross, S. A. "On the General Validity of the Mean-Variance Approach in Large Markets," in W. F. Sharpe and C. M. Cootner, *Financial Economics: Essays in Honor of Paul Cootner*, pp. 52–84. New York: PrenticeHall, Inc., 1982.
Sharpe, W. F. *Portfolio Theory and Capital Markets.* New York: McGraw-Hill, 1970.
Simkowitz, M. A. and W. L. Beedles. "Diversitifcation in a Three-Moment World." *Journal of Finance and Quantitative Analysis*, v. 13 (1978), pp. 927–941.
Von Neumann, J. and O. Morgenstern. *Theory of Games and Economic Behavior*, 2nd ed. Princeton, NJ: Princeton University Press, 1947.
Wackerly, D., W. Mendenhall and R. L. Scheaffer. *Mathematical Statistics with Applications*, 7th ed. California: Duxbury Press, 2007.

Check for updates

Capm, Beta Estimation, and Forecasting

24

24.1 Introduction

This chapter is broken down into the following sections. Section 24.2 looks at the graphical approach to the derivation of the CAPM, and Sect. 24.3 looks at the mathematical approach to the derivation of the CAPM. Section 24.4 discusses the market model and risk decomposition, Sect. 24.5 discusses growth rates, accounting betas, and variance in EBIT, and Sect. 24.6 discusses some applications and implications of the CAPM. Finally, Sect. 24.7 summarizes the chapter.

24.2 A Graphical Approach to the Derivation of the Capm

Following the risk–return tradeoff principle and the portfolio diversification process, Sharpe (1964), Lintner (1965), and Mossin (1966) have developed an asset pricing model that can determine both the market price of a portfolio and the price of an individual security. They focus upon the pricing determination of those parts of security risk that can be eliminated through diversification as well as those that cannot.

Systematic risk is that part of total risk that results from the common variability of stock prices and the subsequent tendency of stock prices to move together with the general market. The other portion of total risk is **unsystematic risk**, the result of variables peculiar to the firm or industry—for example, a labor strike or resource shortage.

Systematic risk, also referred to as market risk, reflects the swings of the general market. Some stocks and portfolios can be very sensitive to movements in the market, while others show more independence and stability. The universally accepted notion for the measure of a stock's or a portfolio's relative sensitivity to the market based upon its past record is the Greek letter **beta (β)**. The estimation, application, and forecasting of beta are discussed in the following sections.

24.2.1 The Lending, Borrowing, and Market Portfolios

If there existed both risk-free and risky assets, investors would then have the choice of investing in one or the other, or in some combination of the two.

The concept of the risk-free asset is generally proxied by a government security, such as the Treasury bill (T-bill). T-bills are backed by the Federal government and are default free, hence considered riskless. An investor's portfolio can be composed of different combinations of riskless and risky assets. Figure 24.1 is a graph of different sets of portfolio opportunities; it includes the risk-free asset with a return of R_f. Since the riskless asset has zero risk, it is represented by a point on the vertical axis.

With the additional alternative to invest in risk-free assets that yield a return of R_f, the investor is able to create new combinations of portfolios that combine the risk-free assets with the risky assets. The investor is thus able to achieve any combination of risk and return that lies along the line connecting R_f and a tangent point M_p, the market portfolio. All portfolios along the line $R_f M_p C$ are preferred to the risky portfolio opportunities on the curve AM_pB. Therefore, the points on the line $R_f M_p C$ represent the best attainable combinations of risk and return.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_24

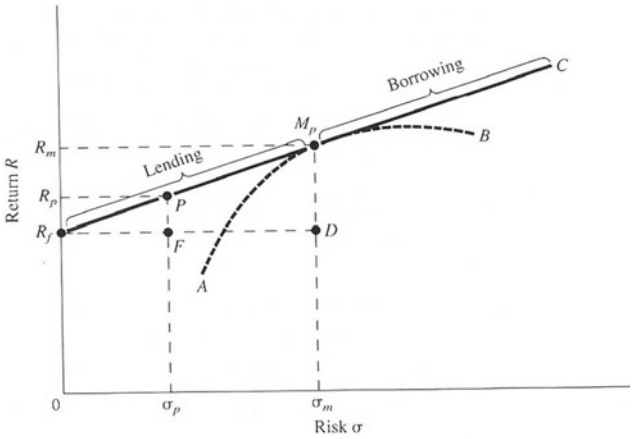


Fig. 24.1 The capital market line

At point R_f investor has all available funds invested in the riskless asset and expects to receive the return of R_f. The portfolios along the line $R_f M_p$ are lending portfolios and contain combinations of investments in the risk-free asset and investments in a portfolio of risky assets M_p. These are called *lending portfolios* because investors are, in effect, lending money to the government at the risk-free rate, when they invest in the risk-free asset.

At point M_p the investor wants only risky assets and has put his wealth into the risky asset portfolio, which is called the **market portfolio**. The investor at this point is neither lending nor borrowing. An alternative view is that at this point the investor may be lending and borrowing equal amounts that just offset each other. At M_p investors receive a rate of return R_m and undertake risk σ_m.

If it is assumed that the investor can borrow money at the risk-free rate R_f and invest this money in the risky portfolio M_p, he or she will be able to derive portfolios with higher rates of return but with higher risks along the line extending beyond $M_p C$. The portfolios along line $M_p C$ are **borrowing portfolios**. They are so called because they contain a negative amount of the risk-free asset. The negative amount invested in the risk-free asset can be viewed as borrowing funds at the risk-free rate and investing in risky assets. The borrowing for investment is called *margin* and is controlled by the government. Therefore, the new efficient frontier becomes $R_f M_p C$ and is referred to as the **capital market line (CML)**. The CML describes the relationship between expected return and total risk.

24.2.2 The Capital Market Line

An illustration and explanation of the CML have already been provided; the equation for the CML is

$$E(R_p) = R_f + \left[E(R_m) - R_f\right] \frac{\sigma_p}{\sigma_m}, \tag{24.1}$$

where

$R_f =$ the risk-free rate;
$R_m =$ return on market portfolio M_p;
$R_p =$ return on the portfolio consisting of the risk-free asset and portfolio M_p;and.
$\sigma_p \sigma_m =$ the standard deviations of the portfolio and the market, respectively.

The implementation of Eq. (24.1) can be explained graphically. An investor has three choices in terms of investments. He or she may invest in the riskless asset R_f in the market portfolio M_p or in any other efficient portfolio along the CML, such as portfolio P of Fig. 24.1.

If the investor puts his or her money into the riskless asset, he or she can receive a certain return of R_f; if his or her investments are put into the market portfolio he or she can expect an average return of R_m and risk of σ_m; if he or she invests

in portfolio P, he or she can expect an average return of R_p with risk of σ_p. The difference between R_m and R_f $\left(R_m - R_f\right)$ is called the **market risk premium**.

The investor of portfolio P only needs to take on risk of σ_p so his or her risk premium is $\left(R_p - R_f\right)$. This is less than the market risk premium because the investor is taking on a smaller amount of risk, $\sigma_p < \sigma_m$.

By geometric theory, triangles $R_f PF$ and $R_f M_p D$ are similar. Consequently, they are directly proportional. Thus:

$$E\left(R_p\right) - R_f = \left[E(R_m) - R_f\right] \frac{\sigma_p}{\sigma_m}.$$

At equilibrium, all investors will want to hold a combination of the risk-free asset and the tangency portfolio $\left(M_p\right)$. Since the market is cleared at equilibrium—that is, prices are such that the demand for all marketable assets equals their supply—the tangency portfolio M_p must represent the market portfolio. An individual security's proportional makeup (X_i) in the market portfolio will be the ratio of its market value (its equilibrium price times the total number of shares outstanding) to the total market value of all securities in the market, or:

$$X_i = \frac{\text{Market value of individual asset}}{\text{Market value of all assets}}. \tag{24.2}$$

Thus, at equilibrium, prices are such that supply is equated to demand, and all securities in the market will be represented in the market portfolio according to their market value.

Given the CML, assuming that investors are interested only in mean and variance of return, the CML will dominate all other attainable portfolios and securities because of the dominance principle. This implies that they will attempt to put some portion of their wealth into the market portfolio of risky assets, depending on their individual risk preference.

Since it has been established that the market portfolio is the only relevant portfolio of risky assets, the relevant risk measure of any individual security is its contribution to the risk of the market portfolio. The beta coefficient relates the covariance between security and market to the market's total variance, and it is the relevant market risk measure. This will be explored in the next two sections. Notationally, beta is

$$\beta_i = \frac{\sigma_{im}}{\sigma_m^2} = \frac{\text{Cov}(R_i, R_m)}{\text{Var}(R_m)}.$$

Example 24.1 provides further illustration of risk/return tradeoffs.

Example 24.1 Describe the kinds of assets that could characterize the risk/return tradeoffs depicted by A, B, and C in Fig. 24.2.

Solution

A is a correctly priced asset, perhaps a stock or a portfolio. B is an overpriced asset, perhaps a bond selling at too large a premium. C is an asset with expected negative return—for example, a lottery ticket. The price of the lottery ticket is too high to be justified by the expected value of winning the jackpot. Even though the jackpot may be large, the probability of winning it is very small, hence its expected value is small. Therefore, the cost of the ticket is greater than the expected value of winning, which yields a negative return.

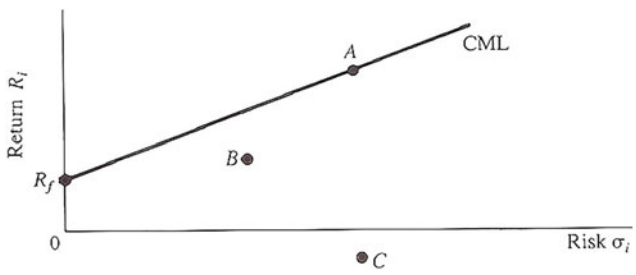


Fig. 24.2 The capital market line

24.2.3 The Security Market Line—The Capital Asset Pricing Model

An asset's systematic risk with the market, β, is the only relevant risk measure of capital asset pricing for the individual asset and the portfolio. Consequently, a derivation of the relationship between systematic risk and return can be made where the expected linear relationship between these two variables is referred to as the **security market line (SML)**, illustrated in Fig. 24.3.

To derive the graphical picture of the SML, it is necessary to list a few assumptions concerning the investors and the securities market.

(1) Investors are risk-averse.
(2) The CAPM is a one-period model because it is assumed that investors maximize the utility of their end-of-period wealth.
(3) All investors have the same efficient frontier—that is, they have homogeneous expectations concerning asset returns and risk.
(4) Portfolios can be characterized by their means and variances.
(5) There exists a risk-free asset with a return R_f, the rate at which all investors borrow or lend. The borrowing rate is equal to the lending rate.
(6) All assets are marketable and perfectly divisible, and their supplies are fixed.
(7) There are no transaction costs.
(8) Investors have all information available to them at no cost.
(9) There are no taxes or regulations associated with trading.

All individual assets or portfolios will fall along the SML. The position of an asset will depend on its systematic risk or beta. The required risk premium for the market is $R_m - R_f$, and β_m is equal to one. Given the SML, the return on a risky asset is equal to

$$E(R_i) = R_f + \beta_i\left[E(R_m) - R_f\right], \tag{24.3}$$

where

$E(R_i)$ = the expected rate of return for asset i;
R_f = the expected risk-free rate;
β_i = the measure of normalized systematic risk (beta) of asset i, and.
$E(R_m)$ = the expected return on the market portfolio.

(*Note*: This SML is also generally called the CAPM. The mathematical derivation of this model will be shown in the next section.)

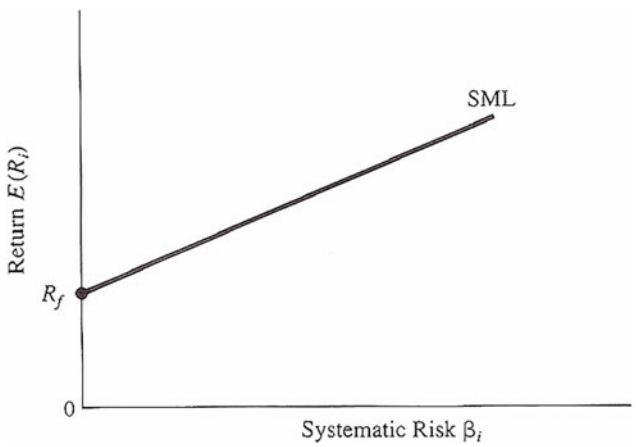


Fig. 24.3 The CAPM showing the SML

The relationship between the CML and the SML can be seen by rearranging the definition of the beta coefficient:

$$\beta_i = \frac{\text{Cov}(R_i, R_m)}{\text{Var}(R_m)} = \frac{\sigma_{im}}{\sigma_m^2} = \frac{\rho_{i,m}\sigma_i\sigma_m}{\sigma_m^2} = \frac{\sigma_{i,m}\sigma_i}{\sigma_m}, \tag{24.4}$$

where

σ_i= standard deviation of a security's rate of return;
σ_m= standard deviation of the market rate of return;
$\sigma_{i,m}$= the correlation coefficient of R_i and R_m; and

$$\sigma_{im} = \frac{\sigma_{im}}{\sigma_i\sigma_m}.$$

If $\sigma_{im} = 1$, then Eq. (24.3) reduces to

$$E(R_i) = R_f + \frac{\sigma_i}{\sigma_m}\left[E(R_m) - R_f\right]. \tag{24.5}$$

If $\sigma_{i,m} = 1$, this implies that this portfolio is an efficient portfolio. If i is an individual security, it implies that the returns and risks associated with the asset are perfectly correlated with the market as a whole. There are several implications:

(1) Eq. (24.3) is a generalized case of Eq. (24.3).
(2) The SML instead of the CML should be used to price an individual security or an inefficient portfolio.
(3) The CML prices the risk premium in terms of total risk, and the SML prices the risk premium in terms of systematic risk.

Example 24.2 provides further illustration.

Example 24.2 Suppose the expected return on the market portfolio is 10% and that $R_f = 6\%$. Further, suppose that you were confronted with an investment opportunity to buy a security with a return expected to be 12% and with $\beta = 1.2$. Should you undertake this investment?

Solution

$$R_i = a + b\beta_i$$

for the market portfolio with $\beta = 1$ we have

$$0.10 = a + (b \times 1) = a + b,$$

and for the risk-free rate with $\beta = 0$, we have

$$0.06 = a + (b \times 0) = a.$$

Solving these equations for a and b yields

$$a = 0.06$$
$$b = 0.04.$$

For the security with $\beta = 1.2$ the expected return given the SML is

$$R_i = 0.06 + 0.04(1.2) = 0.108.$$

Since 12 > 10.8%, the security is undervalued—that is, the security should be purchased since its return is above the equilibrium return of 10.8% for that level of the risk.

24.3 Mathematical Approach to the Derivation of the Capm

Using the assumptions about efficient and perfect markets stated earlier in the chapter, we can show how Sharpe derived the CAPM.

Sharpe (1964) used a general risky asset that did not lie along the CML and dubbed it *i*. Risk and return for the possible combinations of security *i* with the market portfolio M are shown in Fig. 24.4. The average return and standard deviation for any I–M combination can be approached in the Markowitz fashion for a two-asset case:

$$\begin{aligned} &\text{(i)} \qquad E(R_p) = w_i E(R_i) + (1 - w_i) E(R_m) \\ &\text{(ii)} \qquad \sigma(R_p) = \left(w_i^2 \sigma_i^2 + (1 - w_i)^2 \sigma_m^2 + 2(1 - w_i) w_i \sigma_{im} \right)^{1/2} \end{aligned} \tag{24.6}$$

in which w_i represents excess demand for *i* or demand greater than its equilibrium weight in portfolio M.

The changes in mean and standard deviation as the proportion w_i changes are represented by

$$\frac{\partial E(R_p)}{\partial w_i} = E(R_i) - E(R_m)$$

$$\begin{aligned} \frac{\partial \sigma(R_p)}{\partial w_i} = {} & 1/2 \left[w_i^2 \sigma_i^2 + (1 - w_i)^2 \sigma_m^2 + 2 w_i (1 - w_i) \sigma_{im} \right]^{-1/2} \\ & \times \left[\left(2 w_i \sigma_i^2 - 2 \sigma_m^2 + 2 w_i \sigma_m^2 + 2 \sigma_{im} - 4 w_i \sigma_{im} \right) \right]. \end{aligned} \tag{24.7}$$

When $w_i = 0$, the *i*th security is held in proportion to its total market value, and there is no excess demand for security *i*. This is the key insight to Sharpe's argument, for when $w_i = 0$, it is possible to equate the slope of the curve iMi' with the CML, thus obtaining an expression for the return on any risky security *i*. At equilibrium when $w_i = 0$, the slope along the iMi' curve will be equal to

$$\frac{\partial E(R_p)}{\partial \sigma(R_p)} = \frac{\frac{\partial E(R_p)}{\partial w_i}}{\frac{\partial \sigma(R_p)}{\partial w_i}} = \frac{E(R_i) - E(R_m)}{\frac{\sigma_{im} - \sigma_m^2}{\sigma_m}}. \tag{24.8}$$

The slope of the CML at point *M* is

$$\frac{E(R_m) - R_f}{\sigma_m}. \tag{24.9}$$

Setting Eq. (24.7) equal to Eq. (24.8) and rearranging the terms to solve for $E(R_i)$ gives the equation for the SML or CAPM:

$$E(R_i) = R_f + \left[E(R_m) - R_f \right] \frac{\sigma_{im}}{\sigma_m^2}. \tag{24.10}$$

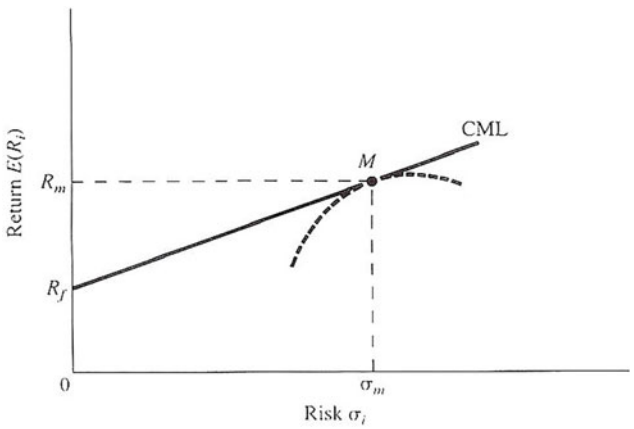


Fig. 24.4 The opportunity set provided by combinations of risky asset *i* and the market portfolio M

This represents the return on any risky asset i. At equilibrium, every risky asset will be priced so that it lies along the SML. It should be noted that the term σ_{im}/σ_m^2 represents the beta coefficient for the regression of R_i versus R_m so that Eq. (9.9) can be rewritten as

$$E(R_i) = R_f + \left[E(R_m) - R_f\right]\beta_i, \tag{24.11}$$

which is the formula for the CAPM.

24.4 The Market Model and Risk Decomposition

To use the CAPM, the market model must be employed to estimate the beta (systematic risk). In addition, the market model can be used to do risk decomposition. Both the market model and risk decomposition are discussed in this section.

24.4.1 The Market Model

As noted previously, the total risk of a portfolio or security can be conceived as the sum of its systematic and unsystematic risks. The equation that expresses these concepts states that the return on any asset at time t can be expressed as a linear function of the market return at time t plus a random error component. Thus, the market model is expressed as

$$R_{i,t} = \alpha_i + \beta_i R_{m,t} + e_{i,t}, \tag{24.12}$$

where

$R_{i,t} =$ the return of the ith security in time t;
$\alpha_i =$ the intercept of the regression;
$\beta_i =$ the slope;
$R_{m,t} =$ the market return at time t; and.
$e_{i,t} =$ random error term.

24.4.2 Risk Decomposition

The regression model of Eq. (24.11), which describes the risky asset's characteristics relative to the market portfolio, is also called the *characteristic line.* Since beta is the slope coefficient for this regression (market model), it demonstrates how responsive returns for the individual earning assets are to the market portfolio. By using the market model, the total variance for security i (σ_i^2) can be represented by and decomposed into

$$\sigma_i^2 = \beta_i^2\sigma_m^2 + \sigma_{ei}^2, \tag{24.13}$$

in which $\beta_i^2\sigma_m^2$ is the systematic risk component of total risk and σ_{ei}^2 is the unsystematic component.

The CAPM as developed here is expressed in terms of expected values. Since expected values are not directly measured, we must transform the CAPM into an expression that uses observable variables. Assuming that on average expected returns $E\left(R_{i,t}\right)$ for a security equal realized returns, returns can be expressed as

$$R_{i,t} = E\left(R_{i,t}\right) + \beta_i\left[R_{m,t} - E\left(R_{m,t}\right)\right] + e_{i,t}, \tag{24.14}$$

in which $e_{i,t}$ is a random error term.

Assuming that the expected value of the error term is 0, that it is uncorrelated with the term $\left[R_{m,t} - E\left(R_{m,t}\right)\right]$ and that Cov $(e_{i,t}, e_{i,t-1}) = 0$, the expression for $E\left(R_{i,t}\right)$ can be substituted from the CAPM Eq. (24.10) into (24.13):

$$R_{i,t} = R_{f,t} + \left[E\left(R_{m,t}\right) - R_{f,t}\right]\beta_i + \left[R_{m,t} - E\left(R_{m,t}\right)\right]\beta_i + e_{i,t}.$$

Table 24.1 Average return, beta coefficient, total variance, and residual variance of J&J and IBM

	Avg. return	Beta	Residual variance	Total variance
Johnson & Johnson	0.008	0.3726	0.0022	0.0025
IBM	0.005	1.1964	0.0041	0.007

Simplifying this equation:

$$R_{i,t} - R_{f,t} = \alpha_i + \beta_i\left(R_{m,t} - R_{f,t}\right) + e_{i,t}, \tag{24.15}$$

where α_i is the intercept. All variables of Eq. (24.14) can be estimated from observed data. Equation (24.14) is called the **risk premium version** of the market model. It is similar to the market model indicated in Eq. (24.11), except that instead of using the total returns R_i and R_m, it uses the risk premium portion of the returns, or $R_i - R_f$ and $R_m - R_f$. Example 24.3 provides further illustration.

Example 24.3 To show how Eqs. (24.11) and (24.12) can be used, we will use monthly return data from Johnson & Johnson (J&J) and IBM. The time period covers from March 2000 to April 2010. Their average return, beta coefficient, total variance, and residual variance are as indicated in Table 24.1 below. We can see that IBM with $\beta = 1.1964$ is more sensitive to the fluctuations of the market than J&J with $\beta = 0.3726$.

Both the magnitude of total variance and beta of IBM were larger than for J&J, although the average rates of return for J&J were higher for IBM. Hence J&J was a more desirable security, if that was the only choice the investor had.

In Appendix 22D, we use Excel program to calculate market model for Johnson & Johnson, IBM, Apple, Coca-Cola, and Caterpillar from January 2016 to March 2021. The beta coefficient estimates for Johnson & Johnson are 0.7032, for IBM is 1.2442, for Apple is 1.2345, for Coca-Cola is 0.6406, and for Caterpillar is 0.9651.

Example 24.4 .

24.4.3 Why Beta is Important for Security Analysis

Implications and applications of beta coefficients in security analysis will be discussed in this section. Beta is used as a measurement of risk: It gauges the sensitivity of a stock or portfolio relative to the market, as indicated in Eq. (24.11). A beta of 1.0 is characteristic of a broad market index such as the NYSE index or the S&P 500. A beta of 2.0 indicates that the stock will swing twice as far in either direction than a fall or rise in the market average. If the market gains 15%, a security of $\beta = 2$ is expected to gain 30%. Conversely, should the market fall 15%, the stock is expected to fall 30%.

A beta of 0.50 indicates that the stock is more stable than the market and will move only half as much as the market. For example, if the market loses 20%, a stock with $\beta = 0.50$ will lose only 10% of the market, and if the market should gain 20%, the stock will gain only 10%. High-beta stocks have been classified as aggressive while low-beta stocks are referred to as defensive. From Table 24.1 it is clear that Pennzoil is an offensive stock while Coca-Cola is a defensive stock.

The CAPM has illustrated the concept that risks are associated with portfolios where the relevant (systematic) risk of an individual security is dependent upon the security's effect on portfolio risk. Therefore, the CAPM equation $E(R_i) = R_f + \beta_i[E(R_m) - R_f]$ represents a description of how rates of return are established in the marketplace assuming investors behave according to the assumptions of the model. Thus a stock's beta measures its contributions to the risk of the portfolio and is therefore a measure of the stock's riskiness relative to the market.

In the previous section, it was found that the rates of return of stock i are presumed to bear a linear relationship with the market rate of return as defined in Eq. (24.11). Equation (24.11) is a fixed-coefficient market model. Following Fabozzi and Francis (1978), Sunder (1980), and Lee and Chen (1980), a random-coefficient market model can be defined as Eq. (24.15a) or Eq. (24.15b):

$$R_{it} = \alpha_i + \beta_{it}R_{mt} + e_{it}, \tag{24.15a}$$

$$R_{it} - R_{ft} = \alpha_i' + \beta_{ii}'\left(R_{mt} - R_{ft}\right) + e_{it}, \tag{24.15b}$$

in which $\beta_{it} = \beta_i + \eta_{it}.\eta_{it}$ represents the random fluctuation associated with the beta coefficient. Using the random-coefficient market model, the total risk can be decomposed into three components, as defined in Eq. (24.16):

$$\sigma_i^2 = \beta_j^2 \sigma_m^2 + \sigma_{\varepsilon i}^2 + \sigma_\eta^2 \sigma_m^2, \tag{24.16}$$

in which $\sigma_\eta^2 \sigma_m^2$ represents an interaction risk between the market and the random fluctuation of the beta. Equation (24.16) is a generalized case of Eq. (24.12). It contains systematic risk, unsystematic risk, and a risk term that reflects any interaction between the systematic and unsystematic risk.

Both fixed-coefficient and random-coefficient market models can be used to do security analysis and portfolio selection as discussed later in this text.

The relationship between total risk, market risk, and firm-specific risk can be shown as

$$\text{Total risk} = \text{Market risk} + \text{Firm} - \text{specific risk}.$$

Since firm-specific risk can be eliminated by diversification:

$$\text{Relevant risk} = \text{Market risk}.$$

Beta is important for the investment manager because it can be used (1) to select individual stocks for investment; (2) to construct portfolios of financial assets with desired levels of risk and return; and (3) to evaluate the performance of portfolio managers. Details of the various uses of beta and the CAPM are provided later in this text. Example 24.4 provides a further illustration.

Example 24.4 Given the SML $\overline{R}_i = 0.06 + 0.08\beta_i$ what should the expected return of a security be if it has a beta twice as great as a similar security returning 18%?

Solution

$$\overline{R}_i = 0.06 + 0.08\beta = 0.18.$$

Solving for β yields

$$\beta = 1.5.$$

Therefore, the security's β is $2 \times 1.5 = 3.0$ and the required return is

$$\begin{aligned} \overline{R}_i &= 0.06 + 0.08(3.0) \\ &= 0.30. \end{aligned}$$

A 30% return is required.

24.4.4 Determination of Systematic Risk

As mentioned above, systematic risk is non-diversifiable because it runs across industries and companies and affects all securities. At this point, we are interested only in those management decisions that can be related to the degree of systematic risk a firm exhibits—namely, the operating decisions and the financing decisions of management.

There are two dimensions of risk that affect a firm's systematic risk. The first is **financial risk**, the additional risk placed on the firm and its stock holders due to the firm's decision to be leveraged—that is, to take on additional debt. The second, **business risk**, is the riskiness involved with a firm's operations, if it takes on no debt.

A particular firm's capital structure affects the riskiness inherent in the company's common stock and thus affects its required rate of return and the price of the stock. A company's capital structure policy requires choosing between risk and return. Taking on increasing levels of debt increases the riskiness of the firm's earning stream, but it usually also results in a higher expected rate of return. High levels of risk tend to lower a stock's price, but a high level of expected rates of return tends to raise it. Therefore, striking a balance with the optimal capital structure maximizes the price of the stock.

Business risk is the risk inherent in a firm's operations. It can also be defined as the uncertainty inherent in projection of future operating income or **earnings before interest and taxes (EBIT)**. Fluctuations in EBIT can result from a number of factors. On the national level these can be economic factors such as inflationary or recessionary times. At the industry level some factors may be the level of competition between similar industries, natural or man-made catastrophes, labor strikes, price controls, and so on. There are a host of possibilities that affect EBIT by raising or lowering its level.

Uncertainty regarding future income flows is a function of the company's business risk. It may fluctuate among industries, among firms, and cross time. The extent of business risk is dependent upon the firm and the Industry. Cyclical industries such as steel production or automobile manufacture have especially high business risks because they are dependent upon the strength of the economy. The retail food industry is considered to be quite stable because food is a necessary good that will be purchased regardless of the state of the economy.

Business risk is dependent upon several factors, the most important of which are listed here:

(1) *Demand variability*: Stability in the levels of demand for the firm's product results in a reduction *of* business risk.
(2) *Sales price variability*: Highly volatile prices result in high-risk, volatile markets; therefore, stability of prices results in reduction *of* business risk.
(3) *Suppliers' price variability*: Firms whose input prices are highly variable are exposed to higher levels of risk.
(4) *Output price flexibility relative to input prices*: As a result of inflation, a firm that is able to raise its output prices with increasing input costs minimizes its business risk.

When a firm uses debt or financial leverage, business risk and financial risk are concentrated on the stockholders. For example, if a firm is capitalized only with common equity, then the investors all share the business risk in proportion to their ownership of stock. If, however, a firm is 50% levered (50% of the corporation is financed by debt, the other half by common equity), the investors who put up the equity will then have to bear all business risk and some financial risk.

The effect of leverage upon return on assets (ROA) and return on equity (ROE) and its effect upon the stockholders can be generalized as follows:

(1) The use of leverage or debt generally increases ROE.
(2) The standard deviation of ROA (σ_{ROA}) is a measure of business risk, whereas the standard deviation of ROE (σ_{ROE}) is a measure of the risk borne by stockholders; $\sigma_{\text{ROA}} = \sigma_{\text{ROE}}$ if the firm is not levered; otherwise with the use of debt $\sigma_{\text{ROE}} > \sigma_{\text{ROA}}$ an indication that business risk is being borne by stockholders.
(3) The difference between σ_{ROE} and σ_{ROA} is the actual risk stockholders face and a measure of the increased risk resulting from financial leverage. Thus

$$\text{Risk of financial leverage} = \sigma^2_{\text{ROE}} - \sigma^2_{\text{ROA}}.$$

24.5 Growth Rates, Accounting Betas, and Variance in Ebit

Besides leverage, other financial variables associated with the firm can affect the beta coefficient. These are the growth rate, accounting beta, and variance in EBIT.

24.5.1 Sustainable Growth Rates

There are two methods for estimating the growth rate. Alternatively, the **growth rate** can be measured in terms of the growth in total assets or the growth in sales. It is determined by the percentage change between the two periods.

$$\frac{\text{sales}_t - \text{sales}_{t-1}}{\text{sales}_{t-1}} \times 100\% \quad \text{or} \quad \frac{\text{total assets}_t - \text{total assets}_{t-1}}{\text{total assets}_{t-1}} \times 100\%.$$

Based upon this definition, Higgins (1977, 1981, and 2008) has developed a sustainable growth rate. Growth and its management present special problems in financial planning. From a financial perspective, growth is not always a blessing. Rapid growth can put considerable strain on a company's resources, and unless management is aware of this effect and takes active steps to control it, rapid growth can lead to bankruptcy. It becomes necessary, therefore, to define a company's sustainable growth rate:

$$\frac{\Delta S}{S} = g^* = \frac{P(1-D)(1+L)}{T - P(1-D)(1+L)} = \frac{(1-D)ROE}{1-(1-D)ROE}, \tag{24.17}$$

where

P = the profit margin on all sales;
D = the target dividend payout ratio;
L = the target debt to equity ratio;
T = the ratio of total assets to sales;
S = annual sales;and
ΔS = the increase in sales during the year.

How is Eq. (24.17) derived? Assuming a company is not raising new equity, the cash to finance growth must come from retained profits and new borrowings:

$$\begin{aligned}\text{Retained profits} &= \text{Profits} - \text{Dividends}\\ &= \text{Profit margin} \times \text{Total sales} - \text{Dividends}\\ &= \text{P}(S+\Delta S)(1-D).\end{aligned}$$

Further, because the company wants to maintain a target debt-to-equity ratio equal to L, each dollar added to the owners' equity enables it to increase its indebtedness by $\$L$. Since the owners' equity will rise by an amount equal to retained profits:

$$\begin{aligned}\text{New borrowings} &= \text{Retained profit} \times \text{Target debt} - \text{to} - \text{equity ratio}\\ &= P(S+\Delta S)(1-D)L.\end{aligned}$$

The use of cash represented by the increase in assets must equal the two sources of cash (retained profits and new borrowings):

$$\begin{aligned}\text{Uses of cash} &= \text{Sources of cash}\\ \text{Increases in assets} &= \text{Retained profits} + \text{New borrowings}\\ \Delta ST &= P(S+\Delta S)(1-D) + P(S+\Delta S)(1-D)L\\ \Delta ST &= P(1-D)(1+L)S + P(1-D)(1+L)\Delta S\\ \Delta S[T - P(1-D)(1+L)] &= P(1-D)(1+L)S\\ \frac{\Delta S}{S} &= \frac{P(1-D)(1+L)}{T - P(1-D)(1+L)}\end{aligned}$$

In Eq. (24.17), the $\Delta S/S$ or g^* is the firm's sustainable growth rate assuming no infusion of new equity. Therefore, a company's growth rate in sales must equal the indicated combination of four ratios, P, D, L, and T. In addition, if the company's growth rate differs from g^*, one or more of the ratios must change. For example, suppose a company grows at a rate in excess of g^*, then it must either use its assets more efficiently or it must alter its financial policies. Efficiency is represented by the profit margin and asset-to-sales ratio. It therefore would need to increase its profit margin (P) or decrease its asset-to-sales ratio (T) in order to increase efficiency. Financial policies are represented by payout or leverage ratios. In this case, a decrease in its payout ratio (D) or an increase in its leverage (L) would be necessary to alter its financial policies to accommodate a different growth rate. It should be noted that increasing efficiency is not always possible and altering financial policies not always wise. (For other available methods to alter growth the reader is referred to Chap. 11 of Lee et al. (2009).

If we divide both numerator and denominator of Eq. (24.17) by T and make some arrangement, then we can show that the sustainable growth rate can be shown as

$$\frac{\Delta S}{S} = \frac{P(1-D)(1+L)}{T - P(1-D)(1+L)} = \frac{(1-D)ROE}{1-(1-D)ROE}, \tag{24.17a}$$

where ROE = the rate of return on equity.

Since sustainable growth rate does not allow the company to use external equity, by allowing the company to use both external debt and equity, Lee et al. (2011) have derived a generalized sustainable growth rate as

$$g(t) = \frac{(1-D)ROE}{1-(1-D)ROE} + \frac{\lambda \cdot \Delta n \cdot p/E}{1-(1-D)ROE}, \tag{24.17b}$$

where

λ= degree of market imperfection.
Δn= number of shares of new equity issued.
p= price per share of new equity.
E= total equity.

24.5.2 Accounting Beta

The **accounting beta** can be calculated from earnings-per-share (EPS) data. Using EPS as an example, beta can be computed as follows:

$$\text{EPS}_{i,t} = \alpha_i + \beta_i \text{EPS}_{m,t} + e_{i,t},$$

where

$\text{EPS}_{i,t}$ = earnings-per-share of firm i at time t;
$\text{EPS}_{m,t}$ = earnings-per-share of market average at time t; and.
$e_{i,t}$ = error term.

The estimate of β_i is the EPS type of accounting beta.

24.5.3 Variance in EBIT

The variance in EBIT (X) can be defined as

$$\frac{\sum_{t=1}^{n} \left(X_t - \overline{X}\right)^2}{n-1},$$

in which X_t = earnings before interest and taxes in period t, and $\overline{X}$ = average EBIT.

The total variance of EBIT can be used to measure the overall fluctuation of accounting earnings for a firm.

24.5.4 Capital–Labor Ratio

The **capital–labor ratio** has an impact upon the magnitude of the beta coefficient. In order to examine this impact, it is necessary to examine the capital–labor ratio.

A production function is a function that can be seen as a function of labor and capital:

$$Q = f(K, L), \tag{24.18}$$

where K = capital and L = labor. K/L (the capital–labor ratio) is generally used to measure a firm's degree of capital intensity. Corporations often choose between increasing their capital intensity through the installation of computers, use of robotics in place of labor, and increase in labor inputs. Small industries that specialize in hand-crafted or -tooled goods will have to increase their capital ratio in order to increase production. However, it has been the trend in recent years for many growth-oriented firms, whether manufacturers or other members of the business sector, to increase their efficiencies through increased investments in capital. Auto manufacturers are finding that robots can assemble cars of high and consistent quality at only a fraction of the cost of workers.

Capital intensity results in increased total risks and generally results in an increase in beta. Large investments are often needed to fully automate a plant or to computerize a bank completely. Taking on debt or issuing securities is normally how these capitalization increases are financed. If the capital–labor ratio is greater than one—that is, if K is greater than L—a firm is capital intensive. If the capital–labor ratio is less than one—that is, if K is less than L—then there is a reduction in capital intensity and a shift toward human-resource investment.

24.5.5 Fixed Costs and Variable Costs

Business risk is dependent upon the extent a firm builds **fixed costs** into its operations. If a large percentage of a firm's costs are fixed, costs cannot decline proportionally when demand falls off. If fixed costs are high, then a slight drop in sales can lead to large declines in EBIT. Therefore, the higher a firm's fixed costs are, the greater its business risk and, generally, the higher its beta.

A firm with a large amount of fixed costs is said to have a *large degree of operating leverage.* An example of these costs may be the highly skilled workers of an engineering firm. The firm cannot hire and fire experienced and highly skilled workers easily; therefore, the workers must be retained and paid during a period of slack demand. Similarly, a firm that is highly leveraged will be characterized by the scenario that small changes in sales will result in large changes in operating income.

Variable costs have the opposite effect, because they are adjustable to the firm's needs. Should a drop in sales occur, variable costs can be lowered to meet the lowered output.

The extent to which firms can control their operating leverage is dependent upon their technological needs. Companies that require large investments in fixed assets—such as steel mills, auto manufacturers, and airlines—will have large fixed costs and operating leverages. Therefore, how much a company is willing to undertake operating leverages must come into play during capital-budgeting decisions. If the company is risk-averse, it may opt for alternatives with smaller investments and fixed costs.

A company having a larger percentage of fixed costs generally implies that they use more capital-intensive types of technology in production. For example, an auto manufacturer such as General Motors (GM) has a higher percentage of fixed costs than a food manufacturer such as General Foods. Therefore, GM's capital–labor ratio can be expected to be higher than that of General Foods. An implication of this phenomenon is that General Foods has a lower beta than GM.

24.5.6 Beta Forecasting

Analysts and investment managers who use the CAPM and beta are interested in whether the beta coefficient and the standard deviation statistics for different securities and portfolios are stable through time or whether they change in a predictable fashion. If the beta coefficient and standard deviation are stable, then using a beta derived from current and historical price data is fine, because the beta today is the same as the beta in the future. However, if the beta coefficient and standard deviation are unstable or vary through time, the analyst or manager must forecast a beta's future value before employing it.

The available evidence on the stability of beta indicates that the beta on an individual security is generally not stable, whereas portfolios have stable betas. Hence we are faced with the problem of forecasting future betas in order to use the CAPM for individual securities. **Beta forecasting** refers to using the historical beta estimates or other historical financial information to forecast future betas.

Beaver et al. (1970) argue that if it is possible to find the underlying determinants of beta, such knowledge can be used to help forecast future betas. Beaver et al. used accounting information from the financial statements of firms to help in the estimation of beta. They concluded that using both price and accounting information shows promise for better beta estimates.

Rosenberg and Marathe (1975) used 54 factors in six categories to estimate betas. Their factors included historical price information, price/earnings (P/E) ratios, financial ratios, and statistical measures associated with the market model. They support the notion that betas are determined by fundamental factors related to firms in addition to market pricing data.

Given that the most useful beta is one that can be forecasted correctly, and that the beta is related to the financial and business risk of the firm as well as the degree to which a firm's businesses co-vary with the total economy, it becomes necessary to determine the best way to forecast beta. The next section addresses this issue.

24.5.7 Market-Based Versus Accounting-Based Beta Forecasting

Market-based beta forecasts are based upon market information alone. Historical betas of firms are used as a proxy for their future betas. This implies that the unadjusted sample beta, $\hat{\beta}_t$, is equal to the population value of future beta:

$$\beta_{t+1} = \hat{\beta}_t. \tag{24.19a}$$

Alternatively, there may be a systematic relationship between the estimated betas for the first period and those of the second period, as shown by Blume (1971):

$$\hat{\beta}_{i,t+1} = a_0 + a_1\hat{\beta}_{i,t} \tag{24.19b}$$

in which $\hat{\beta}_{i,t+1}$ and $\hat{\beta}_{i,t}$ estimated beta for the ith firm in period $t + 1$ and t, respectively. Example 24.5 provides a further illustration.

Example 24.5 If $\hat{a}_0 = 0.35, \hat{a}_1 = 0.80$, and $\hat{\beta}_{i,t}$ = 0.12, then the future beta can be either.

$$\beta_{t+1} = 1.2 \qquad \text{or}$$
$$\hat{\beta}_{t+1} = 0.35 + (0.80)(1.2) = 1.31.$$

It is worthwhile to note that Value Line uses Eq. (24.19b) to estimate the future beta.

Accounting-based beta forecasts rely upon the relationships of accounting information such as the growth rate of the firm, EBIT, leverage, and the accounting beta as a basis for forecasting beta. To use accounting information in beta forecasts, the historical beta estimates are first cross-sectionally related to accounting information such as growth rate, variance of EBIT, leverage, accounting beta, and so on:

$$\beta_i = a_0 + a_1X_{1i} + a_2X_{2i} + a_jX_{ji} + \ldots + a_mX_{mi}, \tag{24.20}$$

where X_{ji} is the jth accounting variables for ith firm, and a_j is the regression coefficient.

Some researchers have found that the historically based beta is the best forecast, whereas others have found that the accounting-based beta is the best. Lee et al. (1986) used composite concepts to show that both accounting and market information are useful for beta forecasting. They found that neither beta forecasts based upon market information nor those based on accounting information are conditionally more efficient with respect to each other. It can be inferred, then, that each set of forecasts contains useful information for the prediction of systematic risk. The composite method can be found in Appendix 24B.

The statistical procedure used by Lee et al. (1986) is the ordinary least-squares method. Ordinary least squares is a statistical procedure for finding the best fitting straight line for a set of points; it seems in many respects a formalization of the procedure employed when fitting a line by eye. For instance, when visually fitting a line to a set of data, the ruler is moved until it appears that the deviations of the points from the prospective line have been minimized. If the predicted value of y_i (dependent variable) obtained from the fitted line is denoted as $\hat{y}_i$ the prediction equation becomes

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i,$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ represent estimates of the true β_0 and β_1; x_i is an independent variable.

Graphically, the vertical lines drawn from the prediction line to each point represent the deviations of the points from the predicted value of y. Thus the deviation of the ith point is $y_i - \hat{y}_i$.

In order to find the best fit, it is necessary to minimize the deviations of the points. A criterion of best fit that is often employed is known as the *principle of least squares.* It may be stated as follows: Choose the best fitting line as the one that minimizes the sum of squares of the errors (SSE) of the observed values of Y from those predicted. Expressed mathematically

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Further, the mean sum of squares of the error (MSSE) of the observed values of Y could be minimized from the predicted mean. Expressed mathematically as

$$\text{MSSE} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}.$$

Mincer and Zarnowitz (1969) suggest a decomposition of the mean squared error term into three components representing bias, inefficiency, and random error. Mathematically, this is represented as

$$\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} = (y_i - \hat{y}_i)^2 + \left(S_y - \rho S_y\right)^2 + \left(1 - \rho^2\right) S_y^2,$$

where $\overline{y}$, and S_y represent the mean, standard deviation, and correlation coefficient of y, respectively.

Using the ordinary least-squares method, Lee et al. (1986) showed that it is possible to achieve better forecasts by combining two types of betas. Using an ordinary least-squares estimate of β_{OLS} and a Bayesian adjustment procedure of Vasicek (1973) as defined in Eq. (24.21):

$$\beta_V = \frac{\frac{\overline{\beta}}{V(\hat{\beta})} + \frac{\hat{\beta}_{\text{OLS}}}{V(\hat{\beta}_{\text{OLS}})}}{\frac{1}{V(\hat{\beta})} + \frac{1}{V(\hat{\beta}_{\text{OLS}})}}, \tag{24.21}$$

where

β_{OLS} = the least-squares estimate of a first-period individual beta;
$V\left(\hat{\beta}_{\text{OLS}}\right)$ = variance estimate of β_{OLS};
$\overline{\beta}$ = the cross-sectional mean value of estimated β_{OLS};
$V\left(\hat{\beta}\right)$ = the cross-sectional variance of estimated β_{OLS}; and.
β_V = the Bayesian-adjusted beta.

Equation (24.21) indicates that the Vasicek type of Bayesian adjustment beta is a weighted average of $\overline{\beta}$ and β_{OLS}. The weights are

$$w_1 = \frac{\frac{1}{V(\hat{\beta})}}{\frac{1}{V(\hat{\beta})} + \frac{1}{V(\hat{\beta}_{\text{OLS}})}} \text{ and } w_2 = \frac{\frac{1}{V(\hat{\beta}_{\text{OLS}})}}{\frac{1}{V(\hat{\beta})} + \frac{1}{V(\hat{\beta}_{\text{OLS}})}}$$

The authors first use first-period regressions projected forward to obtain forecasts of the second-period betas. The summary of the step-wise regression for the first-period data is given in Table 24.2. Table 24.3 summarizes the overall mean squared errors together with the mean squared error decomposition.

Table 24.2 Summary of stepwise regression results for first-period data

Dependent variable Adjusted R^2 Independent variables	$\widehat{\beta}_{OLS}$ 0.245	Coefficients (t-values)	$\widehat{\beta}_{V}$ 0.236
Intercept	0.911		1.037
	(11.92)		(12.71)
Financial leverage	0.704		0.599
	(5.31)		(5.81)
Dividend payout	–0.175		–0.108
	(–3.50)		(–2.85)
Sales	0.030		0.018
	(3.02)		(2.28)
Operating income	0.011		0.026
	(2.18)		(2.28)
Assets	—		–0.026
	—		(–2.41)

Source Lee, C. F.. P. Newbold, J. E. Finnerty, and C. C. Chu. *The Financial Review,* v. 21 (February 1986). p. 59

Table 24.3 Mean squared error decompositions for forecasts of Second-Period

	Market-based forecasts		Accounting-based forecasts	
	Without Bayesian Adjustment	With Bayesian Adjustment	Without Bayesian Adjustment	With Bayesian Adjustment
Bias	0.0003	0.0000	0.0003	0.0000
Inefficiency	0.0309	0.0090	0.0015	0.0001
Random error	0.0581	0.0577	0.0659	0.0638
Total mean				
squared error	0.0893	0.0667	0.0677	0.0640

Source Lee et al. *The Financial Review*. v. 21 (February 1986). p. 600

Lee et al. (1986) note that for forecasts developed without the Bayesian adjustment, the accounting-based forecast is better, in terms of overall mean squared error. This advantage results entirely from inefficiency in the Mincer–Zarnowitz sense of the market-based forecast. By contrast, the Bayesian-adjusted market-based forecasts suffer far less from this problem and, as a result, the mean squared prediction error for Bayesian-adjusted market-based forecasts are only marginally larger than the accounting-based forecasts—0.0667 compared to 0.0640.

Lee et al. (1986) tested a composite predictor for beta consisting of both a market beta and an accounting beta. Table 24.4 shows the results of using this composite method to forecast beta for both the Bayesian-adjusted and non-adjusted model. In Table 24.4, β_{OLS} (2) represents the OLS-estimated beta in the second period; β^{A}_{OLS} and $\hat{\beta}^{A}_{V}$ represent accounting-based beta without and with Bayesian adjustment, respectively. By comparing both MSE (market) of 0.893 and 0.0667 and MSE (accounting) of 0.0677 and 0.0640 with MSE (composite) of 0.0545 and 0.0530, respectively, it can be concluded that on the basis of mean squared error, the composite predictor outperforms either the market- or accounting information predictor. Of the models used for the composite predictor, it appears that the Bayesian adjustment case is superior.

From research it appears, then, that accounting-based and market-based forecasts can be combined to produce a superior composite forecast of beta.

In this section, we have discussed three alternative methods to forecast beta. The importance of beta forecasting in portfolio management will be discussed in Chap. 25. In Appendix 25C, we have used monthly data of Johnson & Johnson, IBM, Caterpillar, and the S&P 500 index to show how the single-index model can be used to estimate the optimal weights of three securities portfolios.

Table 24.4 Results for the estimation of market- and Accounting-based composite with and without bayesian adjustment

	Without bayesian adjustment $\widehat{\beta}_{OLS}(2) = a + b_1\,\widehat{\beta}^{A}_{OLS} + b_2\,\widehat{\beta}_{OLS} + w$		
	Intercept	$\widehat{\beta}^{A}_{OLS}$	$\widehat{\beta}_{OLS}$
Coefficients Standard errors	0.251 0.104	0.399 0.102 MSE = 0.0545	0.370 0.052
	With bayesian adjustment $\widehat{\beta}_{OLS}(2) = a + b_1\,\widehat{\beta}^{A}_{V} + b_2\,\widehat{\beta}_{V} + w$		
	Intercept	$\widehat{\beta}^{A}_{V}$	$\widehat{\beta}_{V}$
Coefficients Standard errors	–0.0% 0.134	0.606 0.133 MSE = 0.0530	0.475 0.067

Source Lee et al. *The Financial Review,* v. 21 (February 1986). p. 61

24.6 Some Applications and Implications of the Capm

Since its development, the uses of the CAPM have extended into all areas of corporate finance and investments. The CAPM can be applied to two aspects of the capital-budgeting problem: (1) determining the cost of capital and (2) assessing the riskiness of a project under consideration. The CAPM can also be useful in a real-estate problem, deciding to lease or buy.

The use of the CAPM can be extended into valuation of the entire firm. Because of its impact upon firm valuation, the CAPM has been of great use in the merger-analysis area of financial analysis. The CAPM has also been used to test various financial theories. By including a dividend term and considering its effects, the CAPM can be used to test the effects of the firm's dividend policy. An area that has received a great deal of attention is the use of the CAPM in testing the efficient-market hypothesis.

An application of the CAPM to the capital-budgeting process concerns the valuation of risky projects. If accurate estimates can be made about the systematic risk of a project, then the CAPM can be used to determine the return necessary to compensate the firm for the project's risk. If the sum of the estimated cash flows discounted by the CAPM-calculated required rate of return is positive, then the firm should undertake the project.

Rubinstein (1973) demonstrates how the CAPM can be used to value securities and calculate their risk-adjusted equilibrium price. First, the CAPM must be converted using price variables instead of expected return. It may be rewritten

$$E(R_i) = \frac{P_1 - P_0}{P_0},$$

where

R_i = the expected returns for the ith firms;
P_1 = the price of stock in time l; and.
P_0 = the price of stock in the previous period.

Thus, the CAPM is redefined:

$$\frac{E(P_1) - P_0}{P_0} = R_{f.t} + \left[E(R_m) - R_f\right]\frac{\sigma_{im}}{\sigma_m^2}, \tag{24.22}$$

or, rearranging Eq. (24.22):

$$P_0 = \frac{E(P_1)}{1 + R_f + \left[E(R_m) - R_f\right]\frac{\sigma_{im}}{\sigma_m^2}}.$$

Thus the rate of return used to discount the expected end-of-period price contains a risk premium dependent upon the security's systematic risk.

CAPM has also been applied in the analysis of mergers. It has been shown that the risks of a portfolio can be substantially reduced with the inclusion of securities that are not perfectly correlated in terms of returns. This principle also applies with respect to the mergers between firms. The merging of two firms with different product lines, called a *conglomerate merger*, creates diversification, considered of great benefit. Suppose that one firm sells a product that is recession resistant, then a decrease in earnings of one division of the conglomerate will be offset by the steady earnings of another division. The overall result will be a relatively stable income stream despite shifting trends in the economy.

24.7 Summary

This chapter has discussed the basic concepts of risk and diversification and how they pertain to the CAPM. The procedures for deriving the CAPM itself were presented, and the CAPM was shown to be an extension of the CML theory. The possible uses of the CAPM in financial management were also indicated.

The concept of beta and its importance to the financial manager was introduced. Beta represents the systematic risk of a firm and is a comparison measure between a particular firm's security or portfolio risk and the market average. Systematic risk was further discussed through an investigation of the beta coefficient and the impact of other important financial variables on the magnitude of the beta coefficient.

The statistical method of least squares and its application was introduced, and beta forecasts based on the least-squares method were compared with those based on market information, accounting information, and a composite predictor beta forecast composed of both accounting and market information. The composite predictor appears to yield a better forecast than either the market information or accounting information forecasts separately.

Appendix 24A: Empirical Evidence for the Risk–Return Relationship

The validity of the CAPM can be borne out partly through observations of actual portfolios held in the marketplace. There exist two primary relationships between risk and return. First, the rates of return of efficient portfolios are linear functions of their riskiness as measured by their standard deviation. This is illustrated by the CML. Second, the rate of return of an individual asset is determined by its contribution of risk to the portfolio, and this is measured by beta, where beta has a linear relationship with the security's expected rate of return. This is illustrated by the SML.

The performance of mutual funds can be employed to test the explanatory powers of the linear relationship between risk and return of the CML. Mutual funds are professionally managed and therefore the most visible type of portfolio, easily used for comparison testing. One study was performed by Sharpe (1966) to test the performance of a fund and the relationship between its rate of return and risk over time. Sharpe computed average annual returns and the standard deviations of these returns for thirty-four mutual funds from 1954 to 1963. His model implies that portfolios with higher risks will receive higher returns. This Sharpe was found to be true for all 34 funds. He calculated the correlation between average returns and their standard deviations to be 0.836, indicating that more than 80% of the difference in returns was due to differences in risk.

Sharpe also found that there was a linear relationship between returns and risks, except in the region of very high risks. Sharpe's study provides basic support to the contention that the CML explains the relationship between risk and return, both in portfolio theory and in the marketplace.

Another study was performed by Jensen (1969). He studied the correlation of beta coefficients (market sensitivity) and the expected return of mutual funds. On the basis of analysis of 115 mutual funds over a 9-year period he was ab1e to conclude that high returns were associated with high volatility or high systematic risks. He also found evidence that beta coefficients are a valid and accurate measure of risk.

Both the Sharpe and Jensen studies on the risk and return of mutual funds show that an empirical risk–return relationship does exist among mutual funds. (However, Sharpe used the capita1 market line to perform his empirical tests, while Jensen used the SML, derived by Sharpe (1964).)

The second implication of the risk–return relationship is that the risk premium on individual assets depends on the contribution each makes to the riskiness of the entire portfolio.

The CAPM is a simple linear model expressed in terms of expected returns and expected risk. In its *ex-ante* form:

$$E(R_i) = R_f + \left[E(R_m) - R_f\right]\beta_i. \tag{24.23}$$

Although many of the aforementioned extensions of the model support this simple linear form, others suggest that it may not be linear, that factors other than beta are needed to explain $E(R_i)$, or that the R_f is not the appropriate riskless rate.

The first step necessary to empirically test the theoretical CAPM is to transform it from expectations (*ex-ante*) form into a form that uses observed data. On average, the expected rate of return on an asset is equal to the realized rate of return. This can be written as

$$R_{it} = E(R_{it}) + \beta_i \delta_{mt} + e_{it},$$

where

$$\begin{aligned} \delta_{mt} &= R_{mt} - E(R_{mt}); \\ E(\delta_{mt}) &= 0; \\ e_{it} &= \text{a random error term}; \\ \text{Cov}(e_{it}, \delta_{mt}) &= 0; \\ \text{Cov}(e_{it}, e_{it-1}) &= 0; \text{and} \\ \beta_{it} &= \text{Cov}(R_{it}R_{mt})\text{Var}(R_{mt}). \end{aligned}$$

When CAPM is empirically tested it is usually written in the following form:

$$R'_{pt} = \gamma_0 + \gamma_1 \beta_p + e_{pt}, \tag{24.24}$$

where

$$\begin{aligned} \gamma_1 &= \overline{R}_{mt} - \overline{R}_{ft} \\ R'_{pt} &= R_{pt} - R_{ft}. \end{aligned}$$

These relationships can be stated as follows.

(1) The intercept term γ_0 should not be significantly different from zero.
(2) Beta should be the only factor that explains the rate of return on a risky asset. If other terms, such as residual variance, dividend yields, P/E ratios, firm size, or beta squared are included in an attempt to explain return, they should have no explanatory power.
(3) The relationship should be linear in beta.
(4) The coefficient of beta, "II" should be equal to $\overline{R}_{mt} - \overline{R}_{ft}$.
(5) When the equation is estimated over very long periods of time, the rate of return on the market portfolio should be greater than the risk-free rate.

The work of Jensen (1972) provides a comprehensive and unifying review of the theoretical developments and the empirical work done in the field until that year. In his paper he points out that.

The main result of the original papers in this area is the demonstration that one can derive the individual's demand function for assets, aggregate these demands to obtain equilibrium prices (or expected returns) solely as a function of potentially measurable market parameters. Thus the model becomes testable.

Let us now summarize the empirical work of Douglas (1969), Black et al. (1972), and Fama and MacBeth (1973). The first published test of the CAPM was by Douglas (1969), who regressed the returns of a large cross-sectional sample of common stocks on their own variances and on their beta coefficients β_i obtained by market models. His results are in variance with the Sharpe–Lintner–Mossin (S–L–M) model, for he found that the return was positively related to the variance of the security but not the covariance with the index of returns.

Douglas also summarizes some of the work of Lintner (1965), who estimates beta from a typical market model. Douglas then adds a term for the standard deviation of error (proxy for unsystematic risk). He finds that the coefficient for the unsystematic risk is both positive and significant, the intercept term is higher than the appropriate risk-free rate, and the coefficient for the market risk premium is too low.

Black et al. observed that cross-sectional tests may not provide direct validation of the CAPM, and they proceeded to construct a time-series test, which they considered more powerful. Their results lead them to assert that the usual form of the

CAPM does not provide an accurate description of the structure of security returns. Their results indicate that β s are nonzero and are directly related to the risk level. Low-beta securities earn significantly more on average than predicted by the model, and high-risk securities earn significantly less on average than predicted by the model. They go on to argue for a two-factor model:

$$R_{it} = (1 - \beta_i)R_{zt} + \beta_i(R_{mt}) + e_{it}. \tag{24.25}$$

If $E(R_z) = 0$ then the S–L–M CAPM would be consistent with this model. However, the cross-sectional term for the intercept is a constant and not equal to zero. They then proceed to look for a rationale for this finding in Black's zero-beta model.

Fama and MacBeth (1973) tested (1) a linear relationship between return on the portfolio and the portfolio's beta and (2) whether the unsystematic risk has an effect between portfolio return and a risk measure in addition to beta. Their basic estimation equation is

$$\tilde{R}_{it} = \tilde{\gamma}_{0t} + \tilde{\gamma}_{1t}\overline{\beta}_i + \tilde{\gamma}_{2t}\overline{\beta}_i^2 + \tilde{\gamma}_{3t}\overline{\sigma}_i^2(\mu) + \tilde{e}_{it}, \tag{24.26}$$

in which $\tilde{\gamma}_{0t}$ is the intercept term, $\overline{\beta}_i$ is the average of the β_i for all individual securities in portfolio i, and $\overline{\sigma}_i(\mu)$ is the average of the residual standard deviations from all securities in portfolio j.

Although they found that there are variables in addition to the portfolio beta that systematically affects period-by-period returns (which are apparently related to the average squared beta of the portfolio and the risk factor other than beta), they dismissed the latter as "almost surely proxies," since "there is no rationale for their presence in our stochastic risk–return model." Their results seem to suggest that the S–L–M model does not hold. The intercept factor, $\tilde{\gamma}_{0t}$ is generally greater than R_F, and $\tilde{\gamma}_{1t}$ is substantially less than $\overline{R}_m - \overline{R}_f$. This seems to indicate that the zero-beta model is more consistent with the data.

Blume and Husick (1973) found empirical evidence to indicate that historical rates of return may sometimes foreshadow changes in future betas and that stocks with higher transaction costs should yield somewhat higher gross expected returns. Their data indicate that beta is not stationary over time and that it does change over time as a function of price. Transaction-cost effects appear less important than the informational effects of price in explaining future returns or future betas. Therefore, the return-generating process may be more complex than what has been assumed.

Blume and Friend (1973) examine both theoretically and empirically the reasons why the CAPM does not adequately explain differential returns on financial assets. Empirically the risk–return tradeoffs implied by stocks on the New York Stock Exchange for three different periods after World War II casted doubt on the validity of the CAPM either in its S–L–M form or zero-beta form. However, they do confirm the linearity of the relationship for NYSE stocks.

Rosenberg and McKibben (1973) observed that predictions of the riskiness of returns on common stocks can be based on fundamental accounting data for the firm and also on the previous history of stock prices. This paper tries to combine both sources of information to provide efficient predictions. A stochastic model of the parameters is built and the mean squared error is used as a criterion for the evaluation of the forecasting performance of estimators. They concluded that the results "strongly confirm the usefulness of the specific risk predictions based on the accounting descriptors."

Merton (1980) is concerned with the estimation of the expected return on the market. He notes that the current practice for estimating the expected market return adds the historical average realized excess market returns to the current observed interest rate. However, while this model explicitly reflects the dependence of the market return on the interest rate, it fails to account for the effects of changes in the level of market risk. Three models of equilibrium-expected market returns are elaborated, and estimation procedures that incorporate the prior restriction that equilibrium-expected excess returns on the market must be positive are derived and applied to return data for the period 1926–1978. The following are the principal conclusions of the study.

(1) The non-negativity restriction above should be explicitly included as part of the specifications.
(2) Estimators that use realized returns should be adjusted for heteroskedasticity.

Roll (1977) directs an attack on the empirical tests of the CAPM. Although recognizing that the theory is testable in principle, he asserts that "no correct and unambiguous test to the theory [has] appeared in the literature and there is practically no possibility that such a test can be accomplished in the future." This conclusion is derived from the mathematical equivalence between the individual return beta linearity and the market portfolio's mean–variance efficiency.

Therefore, any valid test presupposes complete knowledge of the market portfolio's composition. The major results reported by Roll from his theoretical inquiry include:

(1) The only testable hypothesis is that the market portfolio is mean–variance efficient.
(2) All other so-called implications of the CAPM are not independently testable.
(3) In any sample, there will always be an infinite number of *ex post* mean–variance efficient portfolios; betas calculated will satisfy the linearity relation exactly, whether or not the true market portfolio is mean–variance efficient.
(4) The theory is not testable unless the exact composition of the true market portfolio is known and used in the tests.
(5) Using a proxy for the market portfolio does not solve the problem, for the proxy itself might be mean–variance efficient even when the true market portfolio is not, and conversely.
(6) Empirical tests that reject the S–L–M model have results fully compatible with the S–L–M model and a specification error in the measured market portfolio.
(7) If the selected index is mean–variance efficient, then the betas of all assets are related to their mean returns by the same linear function (all assets and portfolios fall exactly on the SML).
(8) For every ranking of performances obtained with a mean–variance inefficient index, there exists another nonefficient index that reverses the ranking.

Roll's critique is a broad indictment of most of the accepted empirical evidence concerning the CAPM theory.

Anomalies in the Semi-Strong Efficient-Market Hypothesis

Three anomalies in the semi-strong efficient-market hypothesis are noteworthy. Four authors—Basu, Banz, Reinganum, and Keirn—deal with these three anomalies: (1) P/E ratios; (2) size effects; and (3) the January effect.

Basu (1977) empirically notes that a firm with a low P/E ratio, when adjusted for risk, has an excess return over firms that have a high P/E ratio. If this is true, then there are implications for the market's efficiency, the validity of the CAPM, or both. However, Basu found that the excess returns, when adjusted for transaction costs, taxes, and so forth, were so much smaller as to be insignificant. Therefore, CAPM and market efficiency were supported. In conjecturing why a difference in P/E ratios could affect the returns of the firm, we believe that the relationship may involve the firm's ability to raise debt. A low P/E ratio may indicate more difficulty in raising capital than a high P/E ratio. This difficulty in raising capital could result in different lending and borrowing rates for different firms. Therefore, a Brennan version of the standard CAPM pricing model may be more applicable. Or the APT model may be more appropriate in pricing the assets where one of the factors involved captures the P/E effect.

Reinganum (1981) also empirically tests the P/E effect and finds the same results as Basu. In addition, Reinganum was concerned with the efficiency of the market. In order to see if the market was informationally efficient, Reinganum also looked at returns of firms with neither high nor low P/E ratios. He found these firms to be correctly priced. From these results, he conjectures that the market was informationally efficient, but CAPM did not allow for the P/E effect on returns. Therefore an APT model with the P/E effect as one of its factors would be preferable to the CAPM model.

Banz (1981) empirically tests the effect of firm size, finding that small-company stock returns were higher than large-company stock returns. Banz argues that the P/E ratio serves as a proxy for the size of a firm and not vice versa. His conjecture about why this anomaly exists centers around informational distribution. A small firm's information distribution is somewhat limited, which causes investors to be wary of buying the stock and depresses the price. Banz suggests that the APT valuation model may be more robust than the CAPM in that the APT would be able to capture the size effect by using a P/E ratio as a proxy for the size effect as a factor in the model. Since the distribution of information affects a firm's ability to raise capital (less information on a firm may cause a firm to pay a premium for capital), the premium would indicate that the lending and borrowing rates of different firms are not the same. A valuation model capturing different lending and borrowing rates was provided by Brennan (1971). The utilization of this model could be implemented where rates differed for large and small firms. If small firms do not borrow in the capital market, then Black's no-borrowing CAPM would be more suitable than the standard CAPM.

Keim (1983) empirically tests one of the most baffling anomalies, the January effect. He found that for stocks with excess returns, over 50% of these excess returns were realized in January. In addition, 50% of the January excess return occurs in the first week of January. This phenomenon of excess returns occurring in the month following the tax-year end has been found empirically in Great Britain also. Although Keim offers no rationale for this phenomenon, others have tried to find tax

reasons for this anomaly. The selling of assets in January rather than December to postpone capital gains taxes to the next taxable year has been one suggested rationale. Unlike the P/E and size effects, this anomaly does not have a clear proxy that could be utilized as a factor in an APT model. CAPM does not capture the effect well, and without any theoretical or economical rationale for this effect, any valuation model would be hard pressed to account for it.

Many of the authors seem to conclude that the standard CAPM is not working well and that alternative valuation models should be considered to capture these anomalies. This might imply that security analysis and portfolio management techniques can be used to beat the market.

Appendix 24B: Composite Forecasting Method

Most forecasts contain some information that is independent of that contained in other forecasts; thus a combination of the forecasts will, quite often, outperform any of the individual forecasts.

Nelson (1973) has shown that a composite forecast of unbiased forecasts is unbiased. For n individual unbiased forecasts $X_i (i = 1, 2, \ldots, n)$ with n weights a_i, each greater than or equal to zero and all weights summing to one, and a composite forecast X, the value of X is then given as

$$X = \sum_{i=1}^{n} a_i X_i \quad \sum_{i=1}^{n} a_i = 1,\ a_i \geq 0.$$

The expected value of X is

$$E(X) = E\left(\sum_{i=1}^{n} a_i X_i\right) = \sum_{i=1}^{n} a_i E(X_i) = \sum_{i=1}^{n} a_i(\mu_x) = \mu_x$$

in which μ_x is the expected value of X_i. Therefore, the expected value of combination of n unbiased forecasts is itself unbiased.

If, however, a combination of n forecasts is formed, m of which are biased, the result is generally a biased composite forecast. By letting the expected value of the ith biased forecast be represented as $E(X_i) = \mu_x + \in_i$ the composite bias can be represented as follows:

$$\begin{aligned} E(X) &= \sum_{i=1}^{n} a_i E(X_i) \\ &= \sum_{i=1}^{m} a_i(\mu + \in_i) + \sum_{i=m+1}^{n} a_i(\mu_x) \\ &= \mu_x + \sum_{i=1}^{m} a_i \in_i \end{aligned}$$

The composite of m-biased forecasts has a bias given by a combination of the individual forecast biases. This suggests that the composite of a biased forecast can be unbiased only if $\sum_{i=1}^{m} a_i \in_i = 0$. In particular, combining two forecasts, one with a positive bias and one with a negative bias, can, for proper choices of weights, result in an unbiased composite, However, for biased forecasts that do not balance each other, and assuming the assignment of zero weights to biased forecasts is not desired, numerous combinations of weights can be selected, each of which gives a composite that is unbiased.

The choice of weights can follow numerous approaches. These range from the somewhat naïve rule of thumb to more involved additive rules. One rule of thumb is that when several alternative forecasts are available but a history of performance on each is not, the user can combine all forecasts by finding their simple average.

Some additive rules may combine the econometric and ARIMA forecasts into a linear composite prediction of the form:

$$A_t = B_1(\text{Econometric})_t + B_2(\text{ARIMA})_t + \in_t, \tag{24.27}$$

where

A_t = actual value for period t;

B_1 and B_2 = fixed coefficients; and.

$\in_t$ = composite prediction error.

Least-squares fitting of (24.27) requires minimization of the sum of errors over values of B_1 and B_2 and, therefore, provides the minimum mean square error linear composite prediction for the sample period. In the case that both the econometric model and ARIMA predictions are individually unbiased, then (24.27) can be rewritten as

$$A_t = B(\text{Econometric}) + (1 - B)(\text{ARIMA})_t + \in_t . \tag{24.28}$$

The least-squares estimate of B in (24.28) is then given by

$$\hat{B} = \frac{\sum_{t=1}^{N} [(\text{ECM})_t - (\text{ARIMA})_t][A_t - (\text{ARIMA})_t]}{\sum_{t=1}^{N} [(\text{ECM})_t - (\text{ARIMA})_t]^2} \tag{24.29}$$

in which $(\text{ECM})_t$ and $(\text{ARIMA})_t$ represent forecasted values from econometric model and ARIMA model, respectively. Equation (24.29) is seen to be the coefficient of the regression of ARIMA prediction errors $[A_t - (\text{ARIMA})_t]$ on the difference between the two predictions. As would seem quite reasonable, the greater the ability of the difference between the two predictions to account for error committed by $(\text{ARIMA})_t$, the larger will be the weight given to $(\text{Econometric})_t$.

Composite predictions may be viewed as portfolios of predictions. If the econometric model's and ARIMA's errors are denoted by u_{1t}, and u_{2t}, respectively, then from (24.28) the composite prediction error is seen to be

$$\in_t = B(u_{1t}) + (1 - B)(u_{2t}). \tag{24.30}$$

The composite error is the weighted average of individual errors. The objective is to minimize the variance of the weighted average, given its expected value. In the case of prediction portfolios the weighted average always has an expectation zero if individual predictions are unbiased; or it may be given expectation zero by addition of an appropriate constant.

Minimizing composite error variance over a finite sample of observations leads to the estimate of B given by

$$\hat{B} = \frac{s_2^2 - s_{12}}{s_1^2 + s_2^2 - 2s_{12}}, \tag{24.31}$$

where s_1^2, s_2^2, and s_{12} are the sample variance of u_{1t}, the sample variance of u_{2t}, and the sample covariance of u_{1t} and u_{2t}, respectively. For large samples, or in the case that the variances Var (u_{1t}) and Var (u_{2t}) and the covariance Cov $(u_{1t}u_{2t})$ are known, Eq. (24.31) becomes

$$B = \frac{\text{Var}(u_{2t}) - \text{Cov}(u_{1t}, u_{2t})}{\text{Var}(u_{1t}) + \text{Var}(u_{2t}) - 2\text{Cov}(u_{1t}, u_{2t})}. \tag{24.32}$$

The minimum-variance weight is seen to depend on the covariance between individual errors as well as on their respective variance. Holding the covariance constant, the larger the variance of the ARIMA error relative to that of the econometric error, the larger the weight given to the econometric prediction.

Bibliography

Banz, R. W. "The Relationship between Return and Market Value of Common Stocks." *Journal of Financial Economics*, v. 9 (March 1981), pp. 3–18.

Basu, S. "Investment Performance of Common Stocks in Relation to Their Price-Earnings Ratios: A Test of the Efficient Markets Hypothesis." *Journal of Finance*, v. 32 (June 1977), pp. 663–682.

Beaver, W., P. Kettler and M. Scholes. "The Association between Market Determined and Accounting Determined Risk Measures." *Accounting Review*, v. 45 (October 1970), pp. 654–682.

Black, F., M. C. Jensen and M. Scholes. "The Capital Asset Pricing Model: Some Empirical Tests," in M. C. Jensen (ed.), *Studies in the Theory of Capital Markets*, pp. 20–46. New York: Praeger, 1972.

Blume, M. E. "On the Assessment of Risk." *Journal of Finance*, v. 26 (March 1971), pp. 1–10.

Blume, M. E. and F. Husick. "Price, Beta, and Exchange Listing." *Journal of Finance*, v. 28 (March 1973), pp. 283–299.

Blume, M. E. and I. Friend. "A New Look at the Capital Asset Pricing Model." *Journal of Finance*, v. 28 (March 1973), pp. 19–34.

Brandt, J. and David A. Bessle, (Feb., 1981), `Composite Forecasting: An Application with U.S. Hog Prices,' American Journal of Agricultural Economics Vol. 63, No. 1 pp. 135–140**.**

Brennan, M. J. "Capital Market Equilibrium with Divergent Borrowing and Lending Rate." *Journal of Financial and Quantitative Analysis*, v. 7 (December 1971), pp. 1197–1205.

Chen, P. J., S. S. Chen, and Y.C. Shih. "The evolution of capital asset pricing models." *Review of Quantitative Finance and Accounting*. Vol.42, No.3, pp. 415-448. (2014)

Douglas, G. W. "Risk in the Equity Markets: An Empirical Appraisal of Market Efficiency." *Yale Economic Essays*, v. 9 (Spring 1969), pp. 3–45.

Elton, E. J., M. J. Gruber, S. J. Brown and W. N. Goetzmann. *Modern Portfolio Theoryand Investment Analysis*, 7th ed. New York: John Wiley & Sons, 2006.

Fabozzi, F. J. and J. C. Francis. "Beta as a Random Coefficient." *Journal of Financial and Quantitative Analysis*, v. 13 (March 1978), pp. 101–116.

Fama, E. F. "Risk, Return and Equilibrium: Some Clarifying Comments." *Journal of Finance*, v. 23 (March 1968), pp. 29–40.

Fama, E. F. and J. MacBeth. "Risk, Return and Equilibrium: Empirical Tests." *Journal of Political Economy*, v. 31 (May–June 1973), pp. 607–636.

Francis, J. C. *Investments: Analysis and Management,* 4th ed. New York, NY: McGraw-Hill Book Company, 1986.

Greene, W. H., Econometric Analysis, 8th ed., New Jersey: Prentice Hall, 2017.

Gu, S., B. Kelly, and D. Xiu. `Empirical Asset Pricing via Machine Learning.' Working paper. 2018.

Harvey, C.R., Y. Liu, and H. Zhu. "… and the Cross-Section of Expected Returns." *The Review of Financial Studies*, Volume 29, Issue 1, 1 January 2016, Pages 5–68.

Higgins, R. C. "How Much Growth Can a Firm Afford?" *Financial Management*, v. 6 (1977), pp. 7–16.

Higgins, R. C. "Sustainable Growth under Inflation." *Financial Management*, *v.* 10 (1981), pp. 36–40.

Higgins, R. C. *Analysis for financial management*, 9th ed. New York, NY: McGraw-Hill, Inc, 2008.

Jensen, M. C. "Capital Markets: Theory and Evidence." *The Bell Journal of Economic and Management Science*, v. 3 (Autumn 1972), pp. 357–398.

Jensen, M. C. "Risk, the Pricing of Capital Assets, and the Evaluation of Investment Portfolio." *Journal of Business*, v. 42 (April 1969), pp. 607–636.

Keim, D. B. "Size-Related Anomalies and Stock Return Seasonality: Further Empirical Evidence." *Journal of Financial Economics*, v. 11 (June 1983), pp. 13–32.

Lee, A. C., J. C. Lee and C. F. Lee (2017). Financial Analysis, Planning and Forecasting: Theory and Application. 3rd edition. World Scientific. Singapore.

Lee, A. C., J. C. Lee and C. F. Lee. *Financial Analysis and Planning and Forecasting: Theory and Application.* Singapore: World Scientific, 2009.

Lee, C. F. and S. N. Chen. "A Random Coefficient Model for Reexamining Risk Decomposition Method and Risk-Return Relationship Test." *Quarterly Review of Economics and Business*, v. 20 (March 1980), pp. 58–69.

Lee, C. F., A. C. Lee and J. C. Lee . *Handbook of Quantitative Finance and Risk Management*. New York: Springer, 2010.

Lee, C. F., M. C. Gupta, H. Y. Chen and A. C. Lee. "Optimal Payout Ratio under Uncertainty and the Flexibility Hypothesis: Theory and Empirical Evidence" *Journal of Corporate Finance* v. 17 (June 2011), pp. 483–501.

Lee, C. F., P. Newbold, J. E. Finnerty and C. C. Chu. "On Accounting-Based, Market-Based and Composite-Based Beta Predictions: Methods and Implications." *The Financial Review*, v. 21 (February 1986), pp. 51–68.

Lee, Cheng-Few, John C. Lee, and Alice C. Lee (2013a). Statistics for Business and Financial Economics. 3rd edition. Springer. New York.

Lee, Cheng-Few, Joseph Finnerty, John Lee, Alice C Lee, and Donald Wort. Security Analysis, Portfolio Management, and Financial Derivatives. World Scientific. Singapore. 2013b.

Lee, Cheng-Few, P. Newbold, J. E. Finnerty and C. C. Chu. `on Accounting-Based, Market-Based and Composite-Based Beta Predictions: Methods and Implications.' The Financial Review, v. 21 (February 1986), pp. 51–68.

Lee, C.F. and G. Yen. ` Efficient Market Hypothesis (EMH): Past, Present and Future' *Review of Pacific Basin Financial Markets and Policies*, Vol. 11, No. 02, pp. 305-329 (2008).

Lee, C.F., J. Finnerty, J. Lee, A.C. Lee, and D. Wort. Security Analysis, Portfolio Management, and Financial Derivatives. Singapore: World Scientific, 2013c.

Lintner, J. "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets." *Review of Economics and Statistics*, v. 47 (February 1965), pp. 13–37.

Merton, R. C. "On Estimating the Expected Return on the Market, an Exploratory Investigation." *Journal of Financial Economics*, v. 8 (December 1980), pp. 323–361.

Miller, M. and M. Scholes. "Rates of Return in Relation to Risk: A Reexamination of Some Recent Findings," in in M. C. Jensen (ed.), *Studies in the Theory of Capital Markets*, pp. 47–78. New York, NY: Praeger, 1972.

Mincer, J. and V. Zarnowitz. "The Evaluation of Economic Forecasts," in J. Mincer (ed.), *Economic Forecasts and Expectations*. New York, NY: National Bureau of Economic Research, 1969, pp. 1–46.

Mossin, J. "Equilibrium in a Capital Asset Market." *Econometria*, v. 34 (October 1966), pp. 768–873.

Reinganum, M. R. "Misspecification of Capital Asset Pricing: Empirical Anomalies Based on Earnings Yields and Market Values." *Journal of Financial Economics*, v. 8 (March 1981), pp. 19–46.

Roll, R. "A Critique of the Asset Pricing Theory's Tests-Part I: On Past and Potential Testability of the Theory." *Journal of Financial Economics*, v. 4 (March l977), pp. 129–176.

Roll, R. "Ambiguity When Performance is Measured by the Securities Market Line." *Journal of Finance*, v. 33 (September 1978), pp. 1051–1069.

Rosenberg, B. and V. Marathe. "Tests of the Capital Asset Pricing Hypothesis." Working Paper No. 32 of the Research Program in France. Graduate School of Business and Public Administration, University of California, Berkeley, May 1975.

Rosenberg, B. and W. McKibben. "The Prediction of Systematic and Specific Risk in Common Stocks." *Journal of Finance and Quantitative Analysis*, v. 8 (March 1973), pp. 317–333.

Rubinstein, M. E. "A Mean-Variance Synthesis of Corporate Financial Theory." *Journal of Finance*, v. 28 (March 1973), pp. 167–168.

Sharpe, W. "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk." *Journal of Finance*, v. 19 (September 1964), pp. 425–442.

Sharpe, W. "Mutual Fund Performance." *Journal of Business*, v. 39(January 1966), pp. 119–138.

Sunder, S. "Stationarity of Market Risk: Random Coefficients Tests for Individual Stocks." *Journal of Finance*, v. 35 (September 1980), pp. 883–896.

Vasicek, O. A. "A Note on Using Cross-Sectional Information in Bayesian Estimation of Security Betas." *Journal of Finance*, v. 28 (December 1973), pp 1233–1239.

Portfolio Selection Methods: Theory and Application

25

25.1 Introduction

The essential difference between the single- and multiple-index models is the assumption that the single-index model explains the return of a security or a portfolio with only the market. The multiple-index model (MIM) describes portfolio returns through the use of more than one index. The investor may quantify the return on a portfolio by seeking an index that is representative of the market together with indexes that are representative of the industries of which the component securities are members or exhibit some other common factor. More is said about the MIM later in the chapter; for now the single-index model is the focus of discussion.

In Sect. 25.2 we discuss the single-index model, and in the following section, Sect. 25.3 we talk about the multiple-index model. Finally, Sect. 25.4 summarizes the chapter.

25.2 The Single-Index Model

The major simplifying assumption that yields the index model from Markowitz's portfolio theory is that covariances between individual securities contained in the portfolio are zero. This assumption greatly reduces the number of calculations needed to find the set of efficient portfolios. The use of the index model necessitates additional statistical estimates for the parameters of the index; nevertheless, these additions are minor in comparison to the reduction in the calculation load as a result of ignoring the covariance terms between securities.

As suggested by Markowitz (1959), the **single-index model** was fully developed by Sharpe (1970), who assumed that the covariances could be overlooked. The return of an individual security was tied to two factors—a random effect and the performance of some underlying market index. Notationally:

$$R_{it} = a_i + b_i R_{It} + e_{it}, \tag{25.1}$$

where

a_i and b_i = regression parameters for the ith firm;
R_{It} = the tth return of some underlying market index;
R_{it} = the tth return on security i; and.
e_{it} = the tth random effect for the ith security.

Equation (25.1) is the market model. This regression makes several assumptions about the random effect term.

(1) The expected value of the tth random effect for security i is zero. More explicitly, $E(e_{it}) = 0$.
(2) The variance of the error terms is constant. This amounts to the assumption that the errors are homoscedastic.
(3) There is no relationship between the errors and the return on the market: Cov $(e_{if}, R_{It}) = 0$.
(4) The random effects are not serially correlated: $E(e_{it}, e_{it+n}) = 0$.
(5) The ith security's random effect is unrelated to any other random effects of any other security: $E(e_{it}, e_{jt}) = 0$.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_25

The fifth assumption guarantees that the regression coefficients a_i and b_i are the best unbiased linear estimators of the true parameters.

An investigation of some of the results of the previous assumptions is in order. The expected value of the return on security i is equal to the sum of the intercept, the adjusted return on the index, and some random effect. This can be expressed as

$$\begin{aligned} E(R_i) &= E(a_i + b_i R_{It} + e_{it}). \\ &= E(a_i) + E(b_i R_{It}) + E(e_{it}) \end{aligned} \tag{25.2}$$

Because a_i and b_i are constants, and $E(e_{it})$ is equal to zero:

$$r_i = E(R_i) = a_i + b_i r_I, \tag{25.3}$$

where r_I is equal to the mean of the returns on the market index.

If the mean of the returns on security i equals r_i then the variance of security i is equal to the expected value of squared deviations from r_{it}. This translates to

$$\begin{aligned} \sigma_i^2 &= E(R_{it} - r_i)^2 \\ &= E[(a_i + b_i R_{It} + e_{it}) - (a_i + b_i r_I)]^2 \\ &= E[b_i(R_{It} - r_I) + e_{it}]^2 \\ &= b_i^2 E(R_{It} - r_I)^2 + 2b_i E[e_{it}(R_{It} - r_I)] + E(e_{it})^2 \end{aligned} \tag{25.4}$$

By the third assumption, the covariance of the random effect and the deviation of the index return from its mean are zero; also, the expected value of the squared errors is equal to the variance of the random effect. Thus:

$$\begin{aligned} \sigma_i^2 &= b_i^2 E(R_{It} - r_I)^2 + E(e_{it})^2 \\ &= b_i^2 \sigma_I^2 + \sigma_{ei}^2 \end{aligned} \tag{25.5}$$

This is equivalent to saying that the variance of the returns on a security is made of some adjusted quantity of the variance of the market (usually referred to as systematic risk) plus the variance of the random effects exclusive to that particular security (unsystematic risk).

The last result to be investigated from these assumptions involves a minor step into abstraction in which the possibility of interaction between two securities and the market index is considered. It is suggested that because variations in the returns of two different securities are not interrelated but only connected to the market index, the covariance between the two securities can be derived from the twice-adjusted variance of the market index (once by the b coefficient of the first security with the market and again by the b coefficient of the second security with the market). The investigation starts with a statement about the covariance of the two securities:

$$\begin{aligned} \sigma_{ij} &= E\big[(R_{it} - r_i)\big(R_{jt} - r_j\big)\big] \\ &= E[((a_i + b_i R_{It} + e_{it})(a_i + b_i r_I)) \\ &\quad \times ((a_j + b_j R_{It} + e_{jt})(a_j + b_j r_I)] \\ &= E\big[(b_i(R_{It} - r_I) + e_{it}) \times \big(b_j(R_{It} - r_I) + e_{jt}\big)\big] \\ &= b_i b_j E(R_{It} - r_I)^2 + b_i E\big(e_{jt}(R_{It} - r_I)\big) \\ &\quad + b_j E(e_{it}(R_{It} - r_I)) + E\big(e_{it} e_{jt}\big) \end{aligned} \tag{25.6}$$

Since according to the third and fourth assumptions the last three terms of the last summation are equal to zero:

$$\sigma_{ij} = b_i b_j \sigma_I^2. \tag{25.7}$$

The number of calculations necessary to utilize the Markowitz model is $N + (N^2 - N)/2$. For a portfolio with a hundred securities, this translates to 5,050 calculations. With the Sharpe single-index model, only 881 estimates are needed for the various security-regression coefficients and only one variance calculation, the variance of the returns on the market. In

addition, 881 estimates of the unsystematic risk σ_{ei}^2 are also needed for the single-index model. Hence, the single-index model has dramatically reduced the input information needed.[1]

25.2.1 Deriving the Single-Index Model

So far only the Sharpe single-index model has been utilized to study the returns of a single security i as determined by its relation to the returns on a market index.

25.2.1.1 Expected Return of a Portfolio

Now consider the return of a portfolio of n securities. The return of a portfolio of n securities is the weighted summation of the individual returns of the component securities. Notationally:

$$E(R_{pt}) = \sum_{i=1}^{n} x_i E(R_i),$$

where Rpt is the rate of return for a portfolio in period t and X_i is the weight associated with the ith security.

$$\begin{aligned} R_{it} &= a_i + b_i R_{It} + e_{it} \\ R_{pt} &= \sum_{i=1}^{n} x_i (a_i + b_i R_{It} + e_{it}) \\ &= \sum_{i=1}^{n} x_i (a_i + e_{it}) + \sum_{i=1}^{n} x_i (b_i R_{It}) \\ &= \sum_{i=1}^{n} x_i a_i + \sum_{i=1}^{n} (x_i b_i)(R_{It}) + \sum_{i=1}^{n} x_i e_{it} \end{aligned} \tag{25.8a}$$

Thus,

$$E(R_{pt}) = \sum_{i=1}^{n} x_i a_i + \sum_{i=1}^{n} (x_i b_i) E(R_{It}). \tag{25.8b}$$

This equation indicates that the return of a portfolio may be decomposed into the summation of the weighted returns peculiar to the individual securities and the summation of the weighted adjusted return on the market index. Thus the portfolio may be viewed as a combination of n basic securities and a weighted adjusted return from an investment in the market index.

25.2.1.2 Variance of a Portfolio

To derive the variance of the portfolio σ_p^2, consider first that the mean return of the portfolio is equal to the expected value of the return on the portfolio. Then, following the definition of σ_p^2 in Eqs. (25.5) and (25.7), we obtain

$$\begin{aligned} \sigma_p^2 &= E\left[\left(R_{pt} - E(R_{pt})\right)^2\right] \\ &= \sum_{i=1}^{n} x_i^2 \sigma_{ei}^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j b_i b_j \sigma_I^2 \\ &= \left[\sum_{i=1}^{n} x_i b_i\right]\left[\sum_{j=1}^{n} x_j b_j\right] \sigma_i^2 + \sum_{i=1}^{n} x_i^2 \sigma_{ei}^2 \end{aligned} \tag{25.9}$$

[1] 1 Discussion of the single-index model adapted in part from Sharpe, W. F. *Portfolio Theory and Capital Markets.* New York, NY: McGraw-Hill, 1970. Adapted by permission.

Because the weighted sum of the b_i coefficients is equal to the coefficient of the portfolio $\sum_{i=1}^{n} x_i b_i = b_p$, similarly $\sum_{j=1}^{n} x_j b_j = b_p$. Hence, the last equation reduces to

$$\sigma_p^2 = b_p^2 \sigma_I^2 + \sum_{i=1}^{n} x_i^2 \sigma_{ei}^2. \tag{25.81}$$

When the number of component securities in a portfolio approaches 15, the unique risk of the component securities is reduced through diversification. In Eq. (25.81) the last term, the weighted sum of the random effect variances, approaches zero as n increases. So, again, as the number of securities increases, the unsystematic risk is reduced and the remaining risk of the portfolio is the adjusted variance of the market index. Example 25.1 further illustrates this concept.

Example 25.1 Given the following information, what should the β of the portfolio (b_p) be?

$$\sigma_p^2 = 0.082\,\sigma_I^2 = 0.021 \sum_{i=1}^{n} x_i^2 \sigma_{ei}^2 = 0.04$$

Solution

Substituting related information into Eq. (25.81):

$$\begin{aligned} b_p^2 &= \frac{\sigma_p^2 - \sum_{i=1}^{n} x_i^2 \sigma_{ei}^2}{\sigma_p^2} \\ &= \frac{0.082 - 0.04}{0.021} \\ &= 2.0 \end{aligned}$$

Therefore, b_p = 1.414.

Equation (81.8b) implies that the portfolio can be viewed as an investment in n basic securities and a weighted adjusted return in the market, or

$$\sum_{i=1}^{n} (x_i b_i) E(R_{It}).$$

The return on the market can be decomposed as a combination of the expected return plus some random effect. When this random effect is positive, the atmosphere is bullish and when it is negative, the atmosphere is bearish. Notationally:

$$\begin{aligned} R_{It} &= E(R_{It}) + e_{n+1,t} \\ &= a_{n+1} + e_{n+1,t} \end{aligned} \tag{25.11}$$

This bit of algebraic maneuvering enables the weighted adjusted investment in the market to be viewed as an investment in an artificial security, the $(n + 1)$th of an n-security portfolio. The weight for this $(n + 1)$th security is the sum of the n weights multiplied by their respective related coefficients to the market index. Thus:

$$x_{n+1} = \sum_{i=1}^{n} x_i b_i. \tag{25.12}$$

The reason for this divergence in notation resulting in the definition of the $(n + 1)$th security's return and weight is that Eq. (25.8) can be simplified to yield a working model for portfolio analysis. Substituting the last results for X_{n+1} and R_{It} into Eq. (25.8b) yields

$$\begin{aligned} R_{pt} &= \sum_{i=1}^{n} x_i(a_i + e_{it}) + x_{n+1}\left(a_{n+1} + e_{n+1,t}\right) \\ &= \sum_{i=1}^{n+1} x_i(a_i + e_{it}) \end{aligned} \tag{25.13}$$

Because the expected value of the random effect terms is zero, the summation that results after the application of the expectations operator can be expressed as

$$\begin{aligned} E\left(R_{pt}\right) &= \sum_{i=1}^{n} x_i a_i + x_{n+1} E(R_{It}) \\ &= \sum_{i=1}^{n+1} x_i a_i \end{aligned} \tag{25.14}$$

This yields a formula for the return of a portfolio that is easily applied to portfolio analysis. Before proceeding, however, it is necessary to simplify the variance formula so that it may be used as well.

Remembering that the variance of the portfolio is the expected value of the squared deviations from the expected market return, the last results concerning R_{pt} and $E\left(R_{pt}\right)$ may be applied:

$$\begin{aligned} \mathrm{Var}\left(R_{pt}\right) &= E\left\{ \left[\sum_{i=1}^{n+1} x_i(a_i + e_{it})\right] - E\left[\sum_{i=1}^{n+1} x_i(a_i + e_{it})\right] \right\}^2 \\ &= E\left\{ \left[\sum_{i=1}^{n+1} x_i(a_i + e_{it})\right] - \sum_{i=1}^{n+1} x_i[E(a_i) + E(e_{it})] \right\}^2 \\ &= E\left\{ \left[\sum_{i=1}^{n+1} x_i(a_i + e_{it})\right] - \sum_{i=1}^{n+1} x_i E(a_i) \right\}^2 \\ &= E\left(\sum_{i=1}^{n+1} x_i e_{it}\right)^2 \\ &= \mathrm{Var}\left(\sum_{i=1}^{n+1} x_i e_{it}\right) \end{aligned} \tag{25.15}$$

This result follows from the assumption that the covariances are equal to zero. Additionally, each variance term is only a weighted sum of the errors around the market return. The direct usefulness of the last two conclusions will become apparent when solving for the security weights. Example 25.2 provides further illustration.

Example 25.2 Given R_{It}, R_{it}, and α_i as indicated in Table 25.1 shown below and the fact that $\sum_{t=1}^{n} \beta_i R_{It} = 56$, using the relationship.

$$\sum_{t=1}^{n} R_{It} = n\alpha_i + \sum_{t=1}^{n} \beta_i R_{It} + \sum_{t=1}^{n} e_{it},$$

find the values for α_i, β_i, and e_i.

Table 25.1 Single-Index Model

R_{It}	R_{it}	α_i	$\beta_i R_{It}$	$e_i e_i$
4	12	2	8	12–2–8 = 2
6	14	2	12	14–2–12 = 0
81	20	2	20	20–2–20 = –2
$\frac{8}{28}$	$\frac{18}{64}$	$\frac{2}{8}$	$\frac{16}{56}$	18–2–16 = $\frac{0}{0}$

Solution

$$\sum_{t=1}^{n} R_{It} = 28, \text{so} \beta_i = \frac{\sum_{t=1}^{n} \beta_i R_{It}}{\sum_{t=1}^{n} R_{It}}$$

$$= \frac{56}{28} = 2$$

Substitute the values $\sum_{t=1}^{n} R_{It} = 64$, $\sum_{t=1}^{n} \beta_i R_{It} = 56$, $n = 4$ into the regression line

$$\sum_{t=1}^{n} R_{It} = n\alpha_i + \sum_{t=1}^{n} \beta_i R_{It} + \sum_{t=1}^{n} e_{it}$$

and solve for α_i.

The $\beta_i R_{It}$ column in Table 25.1 is filled by simply multiplying $\beta(= 2)$ by the R_{It} column. The e_i's are the amounts such that $R_{it} = \alpha + \beta_i R_{mt} + e_{it}$ is an equality, so $R_{it} = \alpha_i + \beta_i R_{It} + e_{it}$ is satisfied. From the last column of the table we know that $\sum_{i=1}^{n} e_{it} = 0$. Therefore

$$\alpha_i = \frac{64 - 56 + 0}{4} = 2.$$

In this section, the expected return $E(R_{pt})$ and the variance of a portfolio $\text{Var}(R_{pt})$ in terms of the single-index model has been derived. In the following section, we discuss portfolio analysis in terms of the single-index optimum portfolio-selection model.

25.2.2 Portfolio Analysis and the Single-Index Model <S2>

Before beginning the portfolio analysis using the single-index model, it is necessary to explain a derivation of security weights through the use of the Lagrangian calculus maximization, in accordance with either Elton et al. (2014) or Francis and Kim (2013). The maximization procedure maximizes a linear combination of the following two Eq. 2[2]:

$$E(R_p) = \sum_{i=1}^{n+1} E(R_i) \text{ or}$$

$$\text{Max} \quad -\text{Var}(R_p) = -\text{Var}\left(\sum_{i=1}^{n+1} x_i e_{it}\right) \tag{25.16a}$$

subject to,

$$\sum_{i=1}^{n+1} x_i = 1 \text{ and } x_{n+1} = \sum_{i=1}^{n} x_i b_i. \tag{25.16b}$$

The first constraint is equivalent to requiring that the sum of the weights of the component securities is equal to one. The second constraint requires that the weight of the market index within the portfolio returns is equal to the summation of the weighted adjustment factors of the component securities. This requirement is as described in Eq. (25.16b).

[2] 2 The maximization of the negative of the variance is equivalent to the minimization of the variance itself. This can be viewed as originating from the negative spectrum of the number line and maximizing toward zero.

Combining the above two objective functions with the two constraints yields a Lagrangian function:

$$\begin{aligned} P &= \Phi E(R_p) - \mathrm{Var}(R_p) + \lambda_1\left(\sum_{i=1}^{n} x_i - 1\right) + \lambda_2\left(\sum_{i=1}^{n} x_i b_i - x_{n+1}\right) \\ &= \Phi\sum_{i=1}^{n+1} x_i a_i - \sum_{i=1}^{n+1}\mathrm{Var}(e_{it}) + \lambda_1\left(\sum_{i=1}^{n} x_i - 1\right) + \lambda_2\left(\sum_{i=1}^{n} x_i b_i - x_{n+1}\right) \end{aligned} \tag{25.16c}$$

The only difference between this maximization function and the minimization shown in either Elton et al. (2014) or Francis and Kim (2013), beyond intent, is that instead of fixing some arbitrary value of return needed (E^*), an attempt can be made to quantify the level of risk aversion that the investors of the portfolio require, thereby placing the portfolio on the efficient frontier not by desired return but by level of utility, as discussed by Elton et al. (2014) and Francis and Kim (2013). This coefficient of risk aversion is denoted by the Greek letter Φ (*Phi*). When low values are exhibited, risk aversion is pronounced; when high values are in evidence, substantial risk taking is allowed. This notion of the level of risk aversion is best pictured in Fig. 25.1. In Fig. 25.1, A represents an investor's objective function to minimize the risk only—therefore, the aggressiveness to return is zero; C represents an investor's objective function to maximize returns only—therefore, the aggressiveness to return is infinite. At B an investor's attitude toward return and risk is between A and C.

Figure 25.2 provides further illustration of the approach being utilized. What is depicted is the variation of the objective function P as the parameter denoting risk is varied. Again, notice that the risk–return relation for a low Φ is lower than that for the more risk taking, high value of Φ. Points A and C are as discussed in Fig. 25.1. Lines BD and $B'D'$ represent the objective function when the risk aversion parameter is equal to one. Note that BD instead of $B'D'$ represents the maximization of the objective function. In sum, different values of Φ generate different optimal objective functions.

Consider now the three-security portfolio. In this framework, the preceding objective function expands to

$$\begin{aligned} P = {} & \Phi x_1 a_1 + \Phi x_2 a_2 + \Phi x_3 a_3 + \Phi x_4 a_4 - x_1^2\mathrm{Var}(e_{1t}) \\ & - x_2^2\mathrm{Var}(e_{2t}) - x_3^2\mathrm{Var}(e_{3t}) - x_4^2\mathrm{Var}(e_{4t}) + \lambda_1 x_1 + \lambda_1 x_2 \\ & + \lambda_1 x_3 - \lambda_1 + \lambda_2 x_1 b_1 + \lambda_2 x_2 b_2 + \lambda_2 x_3 b_3 - \lambda_2 x_4 \end{aligned} \tag{25.17}$$

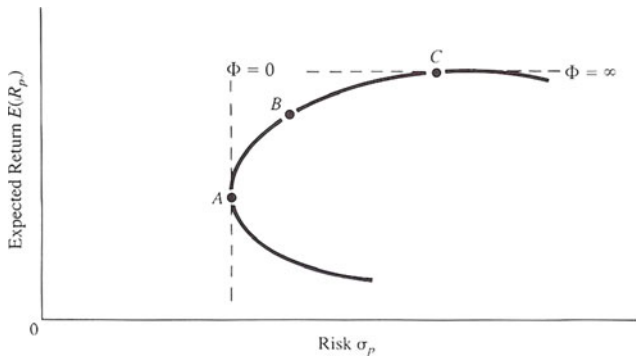


Fig. 25.1 Level of Risk Aversion and Investors' Investment Attitute

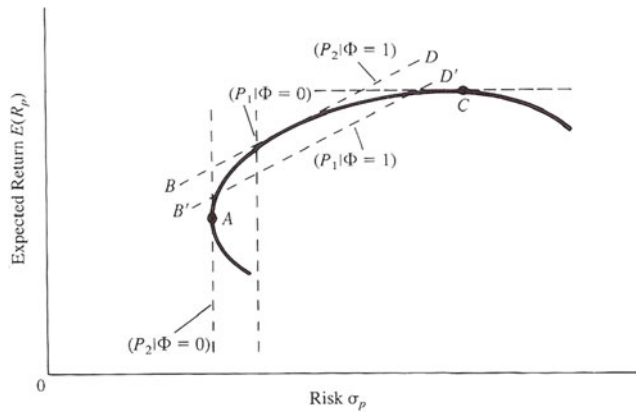


Fig. 25.2 Level of Risk Aversion and Objective Function

Take note of all terms in the preceding equation that contain a_4 and x_4. By utilizing the relations of the individual securities to the market, it has been possible to delete most of the calculations necessitated by the full Markowitz variance–covariance model. The return that will be generated from an optimal portfolio derived through this maximization rests on the estimation of the future expected return and variance of the index. This will lead to the availability of ranging analysis for these estimations, which will produce robust estimates for the security weights highlighted at the end of this section.

To proceed with the Lagrangian maximization it is necessary to notice that the above equation has six unknowns—the four weights and the two-Lagrangian coefficients. All other values are expected to be known or estimated. To maximize, the partial derivative of the objective function is taken with respect to each of the six variables:

$$\begin{aligned}
\frac{\partial P}{\partial x_1} &= \Phi a_1 - 2x_1\sigma_1^2 + \lambda_1 + \lambda_2 b_1 = 0 \\
\frac{\partial P}{\partial x_2} &= \Phi a_2 - 2x_2\sigma_2^2 + \lambda_1 + \lambda_2 b_2 = 0 \\
\frac{\partial P}{\partial x_3} &= \Phi a_3 - 2x_3\sigma_3^2 + \lambda_1 + \lambda_2 b_3 = 0 \\
\frac{\partial P}{\partial x_4} &= \Phi a_4 - 2x_4\sigma_4^2 - \lambda_2 = 0 \\
\frac{\partial P}{\partial \lambda_1} &= x_1 + x_2 + x_3 - 1 = 0 \\
\frac{\partial P}{\partial \lambda_2} &= x_1 b_1 + x_2 b_2 + x_3 b_3 - x_4 = 0
\end{aligned} \tag{25.18}$$

where

$$\begin{aligned}
\sigma_1^2 &= \mathrm{Var}(e_{1t}); \\
\sigma_2^2 &= \mathrm{Var}(e_{2t}); \\
\sigma_3^2 &= \mathrm{Var}(e_{3t}); \text{ and} \\
\sigma_4^2 &= \mathrm{Var}(e_{4t}).
\end{aligned}$$

Equation (25.18) is a set of six equations in six unknowns when set equal to zero. These six equations can be rewritten in matrix format:

$$\begin{matrix} & & & & & & & & & A\,x & = & k \end{matrix}$$

$$\begin{bmatrix} -2\sigma_1^2 & 0 & 0 & 0 & 1 & b_1 \\ 0 & -2\sigma_2^2 & 0 & 0 & 1 & b_2 \\ 0 & 0 & -2\sigma_3^2 & 0 & 1 & b_3 \\ 0 & 0 & 0 & -2\sigma_4^2 & 0 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ b_1 & b_2 & b_3 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -\Phi a_1 \\ -\Phi a_2 \\ -\Phi a_3 \\ -\Phi a_4 \\ 1 \\ 0 \end{bmatrix} \tag{25.19a}$$

Equation (25.19a) has related the individual securities to the index and has discarded the use of numerous covariance terms. The majority of the elements of the matrix indicated in the first matrix of Eq. (25.19a) are zero. For three securities case, the inputs needed for Eq. (25.19a) are three alphas, three betas, and three residual variances for individual securities. In addition, we also need the information of mean and variance of market rate of return. Similarly, for n securities case, we need $3n + 2$ inputs. We know that we need $2n + \frac{n^2-n}{2}$ inputs if we use Markowitz model to obtain the optimal weight of the portfolio.

Following Elton et al. (2014) and Francis and Kim (2013), to solve for the x column vector of variables the P matrix is inverted and each side of the equation is premultiplied by A^{-1}. This yields the x column vector of variables on the left-hand side and A^{-1} multiplied by k, the solution column vector, on the right-hand side.

Table 25.2 Single-Index Model: Three Securities Case

	$\overline{R}_i$	$Var(\overline{R}_i)$	$\mathrm{Cov}(R_i, R_j)$	b_i
JNJ	0.0080	0.0025	0.0007	0.37
IBM	0.0050	0.0071	0.0006	1.20
BA	0.0113	0.0083	0.0007	1.04
S&P 500	− 0.0009	0.0021	0.0021	1.00

$$\begin{aligned} \mathrm{A}x &= k \\ \mathrm{A}^{-1}\mathrm{A}x &= \mathrm{A}^{-1}k \\ \mathrm{I}x &= \mathrm{A}^{-1}k \end{aligned} \tag{25.19b}$$

where I is the identity matrix.

A linear-programming (LP) approach to solve this kind of model is discussed in the first appendix of this chapter.

In order to estimate a single-index type of optimal portfolio the estimates of a market model are needed; a discussion of the market model and beta estimates is therefore needed as well.

Example 25.3 During the discussion of the single-index model, a method was presented for determining optimal portfolio weights given different levels of risk aversion. In this section, a three-security portfolio is examined in which the returns of the securities are related to a market index.

The S&P 500 are included as the market index. The securities and the index have the following observed parameter estimates taken from actual monthly returns during the period from April 2000 to April 2081 (see Table 25.2). The table includes the calculation of the beta estimate for each security. Check these figures, remembering that beta is equal to the covariance of security i with the market divided by the variance of the market returns.

As shown in Eq. (25.19a), a matrix can be developed by utilizing the set of data in the first table of this problem and a Φ of 1.0 (denoting moderate risk aversion):

$$A\,x = k$$

$$\begin{bmatrix} -0.0050 & 0 & 0 & 0 & 1 & 0.37 \\ 0 & -0.0142 & 0 & 0 & 1 & 1.2 \\ 0 & 0 & -0.0166 & 0 & 1 & 1.04 \\ 0 & 0 & 0 & -0.0042 & 0 & 0 \\ 1.0 & 1.0 & 1.0 & 0 & 0 & 0 \\ 0.37 & 1.2 & 1.04 & -1.0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -0.0080 \\ -0.0050 \\ -0.0113 \\ 0.0009 \\ 1.0 \\ 0 \end{bmatrix}.$$

When the P matrix is invented and premultiplies each side of the equation:

$$x = A^{-1}k$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 1.7092 \\ -0.6821 \\ -0.0271 \\ -0.2143 \\ 0.0073 \\ -0.0184 \end{bmatrix}.$$

This solution vector shows that investment should be divided 170.92% in JNJ, − 68.21% in IBM, and − 2.71% in BA. Additionally, the weight of x_4 is the sum of the weighted adjustments as indicated in Eq. (25.16b). The return on this portfolio is the weighted sum of the individual returns:

$$E(R_p) = \sum_{i=1}^{n} x_i E(R_i)$$
$$= [(1.7092 \times 0.0080) + (-0.6821 \times 0.0050) + (-0.0271 \times 0.0113)]$$
$$= 0.0100$$

The variance of the portfolio is the sum of the weighted variance and covariance terms:

$$\mathrm{Var}(R_p) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j \sigma_{ij}.$$

Taking the covariance expressions:

$$\mathrm{Var}(R_p) = [(1.7092^2 \times 0.0025) + (-0.6821^2 \times 0.0071) + (-0.0271^2 \times 0.0083)$$
$$+ 2 \times (1.7092 \times -0.6821 \times 0.0007) + 2 \times (1.7092 \times -0.0271 \times 0.0007)$$
$$+ 2 \times (-0.6821 \times -0.0271 \times 0.0006)$$
$$= 0.0089$$

A portfolio has been developed that is efficient within the realm of this model. By varying the utility factor Φ from risk-averse to more aggressive risk posture we can develop an efficient frontier under the Sharpe single-index model. For a fuller graph, the analyst would continue the calculations with ever higher values of Φ stretching the efficient frontier.

In Appendix 25C, we show how to collect monthly rate of return from Yahoo Finance. In addition, we show how to use the Excel program to calculate optimal weights for Johnson & Johnson, IBM, and Caterpillar in terms of both single-index model and Markowitz model.

25.2.3 The Market Model and Beta

Equation (25.1) defines the market model:

$$R_{it} = a_i + b_i R_{It} + e_{it}. \tag{25.1}$$

From this market model, a_i, b_i, $\mathrm{Var}(e_{it})$ and $\mathrm{Var}(R_{It})$ can be estimated: all are required for the single-index type of optimal portfolio.

So far the regression coefficient has been referred to as b_i when, in fact, it is a risk relationship of a security with the market. Quantified as the covariance of security i with the market index divided by the variance of the return of the market, beta is a relational coefficient of the returns of security i as they vary with the returns of the market. Notationally:

$$b_i = \frac{\sigma_{i,I}}{\sigma_I^2} = \sum_{t=1}^{n} \frac{[(R_{it} - r_{it})(R_{It} - r_{It})]}{(R_{It} - r_{It})^2}. \tag{25.20}$$

Of course, due to the continuous nature of security returns, the result of the above calculation is an estimate and therefore subject to error. This error can be quantified by the use of standard error estimates, which provide the ability to make interval estimations for future predictions. The following discussion is related to the **beta estimate** and its forecasting.

Because of the linkage between security returns and the firm's underlying fundamental nature, beta estimates will vary over time. It is the job of the security analyst to decide how to modify not only the beta estimate but also the period of time from which the sample of returns is to be drawn. Some popular security-evaluation techniques use a tiered growth model to correspond to the product life-cycle theory. Within the scope of this theory, beta estimates for the relation of that security to the market will vary with respect to the time period chosen for the sample of returns. It is obvious that the returns generated by a firm in its infancy have very little correlation with returns during the growth or maturity phase. It is up to the analyst to judge which phase a company may be in and to adjust the beta estimates accordingly.

Assume for expository purposes that all stocks move in perfect alignment with the market, and therefore the security returns all produce beta estimates of 1. It could then be said that any estimation of betas above 1 would indicate positive sampling error, and all estimates below 1 indicate negative sampling error. Future periods would show returns that caused

movement from either side of the estimate deviation back toward 1. Blume (1975) shows that the adjustment of a beta in one period could accurately predict the adjustment in subsequent periods. Utilizing regression analysis, Blume demonstrates that a historical beta could be adjusted successfully to predict future levels. Large brokerage houses have taken this theory a little further by applying weighted averages of historic and average betas in adjusting beta estimates. This has the advantage of being simple to calculate, but it has the ability nevertheless to adjust beta back toward an average level.

Historic beta estimates give the analyst a feeling for how the firm is generating returns to an investor but give very little guidance about the firm's underlying fundamental nature. It is held that a firm's balance-sheet ratios over time can indicate how risky the firm is relative to the market. Using seven financial quantities of a firm, Beaver et al. (BKS 1970) show that betas can be estimated by multiple regression relating the financial levels to the riskiness of the firm by using.

(1) Dividend payout;
(2) Asset growth;
(3) Leverage;
(4) Liquidity;
(5) Asset size;
(6) Earnings volatility; and
(7) Accounting beta (relation of earnings of the firm with the earnings of the economy as a whole).

BKS show not only that each of the variables carries a logical relation to the market but also that the multiple regression is significant. Other studies, most notably Rosenberg and Guy (1976a, b), use many more fundamental factors in the regression for estimating betas.

Historic betas are of consequence because of their relation with market returns; nevertheless, the sampling period tends to crowd out the information contained in returns due to recent developments for the firm. Fundamental betas tend to recognize significant changes in the makeup of a firm—for example, a change in debt structure, or liquidity. The disadvantage related to fundamental betas is that they treat intercompany responsiveness as constant. It is patently obvious that a small firm taking on a large debt will be at much more risk than a Ford or GM doing the same.

Because of the advantages and disadvantages apparent in both types of beta estimates, it has been shown that a combination of the estimates is more suitable to risk classification. Rosenberg and McKibben (1973) found that intercompany responsiveness is substantial, so an analysis was undertaken to introduce these variations into the regression. Utilizing a set of dummy variables to capture the advantages of both types of betas, Rosenberg and Marathe (1974) have described a multiple regression for the beta estimate of a firm that includes information such as variability, level of success, relative size, and growth potential. By using a very large and complex model, they found that the forecasting ability of the regression analysis is substantial, but they were inconclusive as to whether the benefit of the analysis outweighs the computational cost.

Forecasting future beta levels holds its roots in the difficulty of forecasting the fundamental nature of the firm itself. This section has briefly looked over some of the available adjustment processes for beta estimation. While reasonable in its scope, this coverage of the adjustment process is by no means complete. Additional research material is cited at the end of this chapter for use in further study. It should be kept in mind during future readings that the more sophisticated an adjustment process is, the more likely it is that the computational costs will outweigh the improvement of the beta forecast. In Chap. 24, we have discussed three different approaches to forecasting beta.

25.3 Multiple Indexes and the Multiple-Index Model

The previous section reviewed the possibilities of adjusting the beta estimate in the single-index model to capture some of the information concerning a firm not contained in the historic returns. The **MIM** pursues the same problem as beta adjustment, but approaches the problem from a different angle. The MIM tackles the problem of relation of a security not only to the market by including a market index but also to other indexes that quantify other movements. For example, U.S. Steel has returns on its securities that are related to the market, but due to the declining nature of the American steel industry there is also some relation to the steel industry itself. If an index could be developed that represented the movement of the US steel industry, it could be utilized in the return analysis, and a more accurate estimate of possible security returns for U.S. Steel could be derived.

The assumption underlying the single-index model is that the returns of a security vary only with the market index. The expansion provided by the MIM includes factors that affect a security's return beyond the effects of the market as a whole. Realistically any index might be used, but well-known, published indexes are usually incorporated. These may include general business indicators, industry-specific indicators, or even self-constructed indexes concerning the structure of the firm itself. The model to be addressed contains L indexes; nevertheless, a few examples of possible indexes of interest are offered. The MIM is related to the arbitrage pricing model developed by Ross (1976, 1977).

The covariance of a security's return with other market influences can be added directly to the index model by quantifying the effects through the use of additional indexes. If the single-index model were expanded to take into account interest rates, factory orders, and several industry-related indexes, the model would change to

$$R_i = a_i^* + b_{i1}^* I_1^* + b_{i2}^* I_2^* + \cdots + b_{iM}^* I_M^* + c_i. \tag{25.21}$$

In this depiction of the MIM, I_j^* is the actual level of index j, while b_{ij}^* is the actual responsiveness of security i to index j. If the component of the security return is not related to any of the indexes then this index model can be divided into two parts: (1) a_i, the expected value of the unique return, and (2) c_i which represents the distribution of the random effect; c_i has a mean effect of zero and a variance of $\sigma_{c_i}^2$.

This model can be utilized with multiple regression techniques, but if the indexes are unrelated to each other the calculations would be made much simpler. This assumption reduces the number of calculations, as compared with the Markowitz full variance–covariance model, but is obviously more complex than the single-index model. To assure that the indexes are unrelated, the index variables can be orthogonalized (made uncorrelated) by completing inter-regressions on the indexes themselves.

Assume there is a hypothetical model that deals with two indexes:

$$R_i = a_i^* + b_{i1}^* I_1^* + b_{i2}^* I_2^* + c_i. \tag{25.22}$$

Suppose the indexes are the market index and an index of wholesale prices. If these two indexes are correlated, the correlation may be removed from either index.

To remove the relation between I_1^* and I_2^*, the coefficients of the following equation can be derived by regression analysis:

$$I_2^* = e_0 + e_1 I_1^* + d_i,$$

where
e_0 and e_1 = the regression coefficients; and.
d_i = the random error term.

By the assumptions of regression analysis, d_i is uncorrelated with I_1. Therefore:

$$\hat{d}_i = I_2^* - \left(\hat{e}_0 + \hat{e}_1 I_1^*\right),$$

which is an index of the performance of the sector index without the effect of I_1(the market removed). Defining:

$$I_2 = \hat{d}_i = I_2^* - \hat{e}_0 - \hat{e}_1 I_1^*$$

an index is obtained that is uncorrelated with the market. By solving for I_2^* and substituting into Eq. (25.22):

$$R_i = a_i^* + b_{i1}^* I_1^* + b_{i2}^* I_2 - b_{i2}^* \hat{e}_0 - b_{i2}^* \hat{e}_1 I_1^* + c_i.$$

Rearranging:

$$R_i = \left(a_i^* - b_{i2}^* \hat{e}_0\right) + \left(b_{i1}^* - b_{i2}^* \hat{e}_1\right) I_1^* + b_{i2}^* I_2 + c_i.$$

If the first set of terms in the brackets are defined as a_i and the second set of terms are defined as b_{i1}, $b_{i2}^* = b_{i2}$, $I_1^* = I_1$, and $c_i = e_i$, the equation can be expressed:

$$R_i = a_i + b_{i1} I_1 + b_{i2} I_2 + e_i, \tag{25.23}$$

in which I_1 and I_2 are totally uncorrelated: The goal has been achieved. As will be seen later, these simplifying calculations will make the job of determining variance and covariance much simpler.

The expected return can be expressed:

$$\begin{aligned} E(R_{it}) &= E(a_i + b_{i1}I_{1t} + b_{i2}I_{2t} + e_{it}) \\ &= E(a_i) + E(b_{i1}I_{1t}) + E(b_{i2}I_{2t}) + E(e_{it}) \\ &= a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2 \end{aligned} \tag{25.24}$$

Since a_i, b_{i1} and b_{i2} are constants, $E(e_i) = 0$ by assumption, where $\overline{I}_1 = E(I_{1t})$ and $\overline{I}_2 = E(I_{2t})$.
Variance can be expressed:

$$\begin{aligned} \sigma^2 &= E(R_{it} - R_i)^2 \text{ where } R_i = E(R_{it}) \\ &= E[(a_i + b_{i1}I_1 + b_{i2}I_2 + e_{it}) - (a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2)]^2 \\ &= E[(a_1 + a_i) + b_{i1}(I_{1t} + \overline{I}_1) + b_{i2}(I_{2t} + \overline{I}_2) + e_{it}]^2 \\ &= E[b_{i1}^2(I_{1t} + \overline{I}_1)^2 + 2b_{i1}b_{i2}(I_{1t} + \overline{I}_1)(I_{2t} + \overline{I}_2) \\ &\quad + b_{i2}^2(I_{2t} + \overline{I}_2)^2 + 2b_{i1}(I_{1t} + \overline{I}_1)e_{it} \\ &\quad + 2b_{i2}(I_{2t} + \overline{I}_2)e_{it} + e_{it}^2 \\ &= b_{i1}^2E(I_{1t} + \overline{I}_1)^2 + b_{i1}b_{i2}E[(I_{1t} + \overline{I}_1)(I_{2t} + \overline{I}_2)] \\ &\quad + b_{i2}^2E(I_{2t} + \overline{I}_2) + b_{i1}E[(I_{1t} + \overline{I}_1)e_i] \\ &\quad + b_{i2}E[(I_{2t} + \overline{I}_2)e_i] + E(e_i^2) \end{aligned}$$

But by assumption:

$$E[(I_{1t} + \overline{I}_1)(I_{2t} + \overline{I}_2)] = 0,$$

$$E[(I_{1t} + \overline{I}_1)e_i] = 0,$$

$$E[(I_{2t} + \overline{I}_2)e_i] = 0,$$

and

$$E(I_{1t} + \overline{I}_1)^2 = \sigma_1^2,$$

$$E(I_{2t} + \overline{I}_2)^2 = \sigma_2^2,$$

$$E(e_i^2) = \sigma_{ei}^2;$$

therefore,

$$\sigma_i^2 = b_{i1}^2\sigma_1^2 + b_{i2}^2\sigma_2^2 + \sigma_{ei}^2. \tag{25.25}$$

Covariance between security i and security j can be expressed as

$$\begin{aligned} \sigma_{ij} &= E[(R_{it} - R_i)(R_{jt} - R_j)][\text{where } R_i = E(R_{it}) \text{ and } R_j = E(R_{jt}) \\ &= E\{[(a_i + b_{i1}I_1 + b_{i2}I_{2t} + e_{1t}) - (a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2)] \\ &\quad \times [(a_j + b_{j1}I_{1t} + b_{j2}I_{2t} + e_{jt}) - (a_j + b_{j1}\overline{I}_1 + b_{j2}\overline{I}_2)]\} \\ &= E\{[b_{i1}(I_{1t} - \overline{I}_1) + b_{i2}(I_{2t} - \overline{I}_2) + e_i] \times [b_{j1}(I_{1t} - \overline{I}_1) \\ &\quad + (b_{j2}(I_{2t} - \overline{I}_2) + e_{jt}]\} \\ &= E[b_{i1}b_{j1}(I_{1t} - \overline{I}_1)^2 + b_{i2}b_{j2}(I_{2t} - \overline{I}_2)^2] \\ &= b_{i1}b_{j1}\sigma_1^2 + b_{i2}b_{j2}\sigma_2^2 \end{aligned} \tag{25.26}$$

since all remaining expected values of the cross-product terms equal zero.

The extended results of expected return variance and covariance for a multi-index and more than two indexes can be found in the second appendix of this chapter.

One simplifying way of applying the MIM is to start with the basic market model and add indexes to reflect industry-related effects. If the firm has 881% of its operations in one industry, Eq. (25.23) can be used to represent a two-index model with a market index and industry index. In general this approach reduces the number of data inputs to $4N + 2I + 2$. These data inputs are: (1) the expected return and variance for each stock and market index; (2) the covariance between the individual security and the market index and the industry index; and (3) the mean and variance of each industry index. Although this is a larger number of data inputs than for the simple market model, the accuracy of the estimation of security return increases. So the tradeoff is one of more information (higher cost to use) versus greater accuracy of the forecasted security return.

Care must be taken in applying the MIM. It is often the case that the additional information resulting from the application of a higher-complexity model is outweighed by the computational cost increase. In an attempt at making the model as parsimonious as possible, it is necessary to judge the increased information gained by utilizing the more complex model. This can be accomplished by examining the mean square error for the forecast of the actual historic values. Although not within the scope of this text, the ability to judge the accuracy of a forecast is essential and can be acquired from any good statistical forecasting text. Example 25.3 provides further illustration of the single-index model.

25.4 Summary

This chapter has discussed the essentials of single- and multiple-index models. The theoretical underpinnings of the theories have been explored and numerical examples have been provided. An efficient boundary has been derived under the guidelines of the model and the quantitative analysis of the related parameters has been studied. It has been shown that both single-index and multi-index models can be used to simplify the Markowitz model for portfolio section; remember, however, that the MIM is much more complicated than the single-index model.

Appendix 25A: A Linear-Programming Approach to Portfolio Analysis Models

Sharpe (1967) developed a simplified portfolio analysis model designed to be formulated as a LP problem. Jacob (1974) developed a LP model for small investors that delineates efficient portfolios composed of only a few securities. Both of these approaches have as their objectives: (1) a reduction in the amount of data required and (2) a reduction in the amount of computer capability required to solve the portfolio-selection problem.

Sharpe approaches the problem of capturing the essence of mean–variance portfolio selection in a LP formulation by.

(1) Making a diagonal transformation of the variables that will convert the problem into a diagonal form, and
(2) Using a piecewise linear approximation for each of the terms for variance.

The LP that results from the use of market responsiveness as the risk measure and the imposition of an upper limit on investment in each security is

$$\text{Max } P = \lambda \left[\sum_{i=1}^{n} x_i E(R_i) \right] - (1 - \lambda) \left[\sum_{i=1}^{n} x_i \beta_i \right] \tag{25A.1}$$

subject to

$$\sum_{i=1}^{n} x_i = 1$$
$$0 \leq x_i \leq U$$

where
x_i = the fraction of the portfolio invested in security i;
$E(R_i)$ = the expected returns of security i;

β_i = the beta coefficient of security i;
U = the maximum fraction of the portfolio that may be held in any one security; and.
λ = a parameter reflecting the degree of risk aversion.

The λ is used in generating the efficient frontier. The value of λ corresponds to the rate of substitution of return for risk measured by the individual security β, while the U maximum percentage to be invested in any one security greatly reduces the number of securities that will be needed for diversification.

Building on this framework, Jacob derives a LP model that allows the small investor explicit control over the number of securities held. Jacob's LP model incorporates the effects of unsystematic risk as well as systematic risk, the beta in the Sharpe LP. Jacob's model is to minimize

$$\sigma^2(R_p) = \underbrace{\frac{1}{K}\left(\sum_{i=1}^{n} x_i\beta_i\right)^2 \sigma^2(R_m)}_{\text{Systematic risk}} + \underbrace{\left(\frac{1}{K}\right)^2 \sum_{i=1}^{n} x_i^2\sigma^2(e_i)}_{\text{Unsystematic risk}} \tag{25A.2}$$

subject to

$$\frac{1}{K}\sum_{i=1}^{n} x_i E(R_i) \le E^*(R_p)$$

$$\frac{1}{K}\sum_{i=1}^{n} x_i = K \quad x_i = 0 \text{ or } 1$$

where
x_i = the investment in security i—all, or nothing;
$E(R_i)$ = the expected return on security i equal to $\alpha_i + \beta_i R_m$.
K = the desired upper bound on the number of securities the investor is willing to consider, usually in the range of 15–20;
β_i = the measure of systematic risk;
$\sigma^2(e_i)$ = the measure of unsystematic risk; and.
$E^*(R_p)$ = the lowest acceptable rate of return the investor is willing to earn on his or her portfolio.

An additional simplification is to turn the objective function of Eq. (25A.2) into a linear relationship. Since the decision variables (x_i) are binary valued, the unsystematic risk term of the objective function is already linear. Given that the portfolio beta $\left(\sum_{i=1}^{n} x_i\beta_i\right)/K$ is very close to unity, a reasonable linear approximation to the first term in the objective function is provided by

$$\frac{1}{K}\left[\sum_{i=1}^{n} x_i\beta_i\sigma^2(R_m)\right] = \sigma^2(R_m)\left(\frac{1}{K}\sum_{i=1}^{n} x_i\beta_i\right)^2.$$

After division by K and rearrangement of terms, the objective function can be restated as

$$Z = \sum_{i=1}^{n} x_i\left[\beta_i\sigma^2(R_m) + \frac{1}{K}\sigma^2(e_i)\right].$$

Letting $Z_i = \beta_i\sigma^2(R_m) + (1/K)\sigma^2(e_i)$, the problem can be cast as

$$\text{Minimize } Z = \lambda\sum_{i=1}^{n} X_i Z_i - (1-\lambda)\sum_{i=1}^{n} X_i E(R_i),$$

subject to

$$\sum_{i=1}^{n} X_i = 1.0 \quad 0 \le X_i \le 1/K \quad i = 1, 2, \ldots, N,$$

in which λ is varied from zero to one. This problem can be solved by the LP approach suggested by Sharpe (1967). An example of solving this problem can be found in Jacob (1974).

Appendix 25B: Expected Return, Variance, and Covariance for a Multiple-Index Model

Using the orthogonalization technique discussed in the text, the MIM is transformed into

$$R_i = a_i + b_{i1}I_1 + b_{i2}I_2 + \cdots + b_{iL}I_L + c_i \tag{25B.1}$$

in which all I_j are uncorrelated with each other. To interpret the transformed indexes, notice that I_2 is now the difference between the actual level of the index and the level it would be, given the level of the other indexes. Also, b_{i2} is now the sensitivity of security i's rcxseturn to a change in I_2, given that all other indexes are held constant.

It is also convenient, beyond making the indexes uncorrelated, to assume that the covariance of the residuals with the indexes is equal to zero. With this final assumption, the Multi-Index Model (MIM) can be recapped as follows. Generalized equation:

$$R_i = a_i + b_{i1}I_1 + b_{i2}I_2 + \cdots + b_{iL}I_L + c_i$$

for all securities i = 1 to N, and

(1) $E(c_i) = 0$
(2) Var $(c_i) = \sigma_{ci}^2$
(3) Var $(I_j) = \sigma_{Ij}^2$
(4) Cov $(I_j, I_k) = E[(I_j - \overline{I}_j)(I_k - \overline{I}_k)] = 0$, $\overline{I}_j = E(I_j)$, $\overline{I}_{ci} = E(I_{ci})$
(5) $\text{Cov}(c_i, I_j) = E[c_i(I_j - \overline{I}_j)] = 0$
(6) $\text{Cov}(c_i, c_j) = E(c_i, c_j) = 0$ for all $i, j,$ and k.

The last statement is equivalent to the residuals being unrelated, which is to say that the only reason for the common co-movement of returns is due to the movement of the indexes. Although this is like ignoring some of a model's shortcomings, it facilitates a usable model to obtain a quantified idea of future return patterns.

The expected value of the model with the multiple indexes is

$$\begin{aligned} E(R_i) &= E(a_i + b_{i1}I_1 + b_{i2}I_2 + \cdots + b_{iL}I_L + c_i) \\ &= E(a_i) + E(b_{i1}I_1) + E(b_{i2}I_2) + \cdots + E(b_{iL}I_L) + E(c_i) \end{aligned}$$

because a and the b are constants, and the expected value of the residuals is equal to zero

$$r_i = a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2 + \cdots + b_{iL}\overline{I}_L, \tag{25B.2}$$

where $\overline{I}_j$ is the expected value of index j.

The variance of the returns using multiple indexes is

$$\sigma_i^2 = E(R_i - r_i)^2,$$

where r_i is the expected value of the returns of security i. Substituting R_i and r_i from above:

$$\begin{aligned} \sigma_i^2 = E[&(a_i + b_{i1}I_1 + b_{i2}I_2 + \cdots + b_{iL}I_L + c_i) \\ &- (a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2 + \cdots + b_{iL}\overline{I}_L)]^2 \end{aligned}$$

Rearranging, and noticing that the a_i cancel, yields

$$\sigma_i^2 = E[b_{i1}(I_1 - \overline{I}_1) + b_{i2}(I_2 - \overline{I}_2) + \cdots + b_{iL}(I_L - \overline{I}_L) + c_i]^2.$$

Next the terms in the brackets are squared. To proceed with this concentrate on the first index, arid the rest of the terms involving the other indexes follow directly. The first index times itself and all other terms yields

$$\begin{aligned} E[b_{i1}^2(I_1 - \overline{I}_1)^2 + b_{i1}b_{i2}(I_1 - \overline{I}_1)(I_2 - \overline{I}_2) + \cdots + \\ b_{i1}b_{iL}(I_1 - \overline{I}_1)(I_L - \overline{I}_L) + b_{i1}(I_1 - \overline{I}_1)(c_i)] \end{aligned}$$

Remembering that

$$E\left[\left(I_i - \overline{I}_i\right)\left(I_j - \overline{I}_j\right)\right] = 0 \text{ and } E\left[\left(I_i - \overline{I}_i\right)c_i\right] = 0$$

the only nonzero term involving index one is expressed as

$$b_{i1}^2 E\left(I_1 - \overline{I}_1\right)^2 = b_{i1}^2 \sigma_{I1}^2.$$

Because all terms involving c_i are zero and $E(c_i)^2 = \sigma_{ci}^2$:

$$\sigma_i^2 = b_{i1}^2 \sigma_{I1}^2 + b_{i2}^2 \sigma_{I2}^2 + \cdots + b_{iL}^2 \sigma_{IL}^2 + \sigma_{ci}^2. \tag{25B.3}$$

The covariance of the returns between security i and j utilizing the MIM can be expressed as

$$\sigma_{ij} = E\left[\left(R_i - r_i\right)\left(R_j - r_j\right)\right].$$

Again, substituting for R_i, r_i, R_j, and r_j:

$$\begin{aligned}\sigma_{ij} = E[[&(a_i + b_{i1}I_1 + b_{i2}I_2 + \cdots + b_{iL}I_L + c_i)\\ &-(a_i + b_{i1}\overline{I}_1 + b_{i2}\overline{I}_2 + \cdots + b_{iL}\overline{I}_L)]^*\\ &[(a_j + b_{j1}I_1 + b_{j2}I_2 + \cdots + b_{jL}I_L + c_j)\\ &-(a_j + b_{j1}\overline{I}_1 + b_{j2}\overline{I}_2 + \cdots + b_{jL}\overline{I}_L)]]\end{aligned}$$

Again, noting that a_i cancels and combining the terms involving the same b's:

$$\begin{aligned}\sigma_{ij} = E\{&[b_{i1}(I_1 - \overline{I}_1) + b_{i2}(I_2 - \overline{I}_2) + \cdots + b_{iL}(I_L - \overline{I}_L) + c_i]\\ &\times[b_{j1}(I_1 - \overline{I}_1) + b_{j2}(I_2 - \overline{I}_2) + \cdots + b_{jL}(I_L - \overline{I}_L) + c_j]\}\end{aligned}$$

Next multiply out terms, again concentrating on the terms involving b_{i1}:

$$\begin{aligned}E[&b_{i1}b_{j1}(I_1 - \overline{I}_1)^2 + b_{i1}b_{i2}(I_1 - \overline{I}_1)(I_2 - \overline{I}_2) + \cdots +\\ &b_{i1}b_{jL}(I_1 - \overline{I}_1)(I_L - \overline{I}_L) + b_{i1}(I_1 - \overline{I}_1)c_j].\end{aligned}$$

Because the covariance between two indexes is zero, and the covariance between any residual and an index is zero:

$$b_{i1}b_{j1}E(I_1 - \overline{I}_1)^2 = b_{i1}b_{j1}\sigma_{I1}^2.$$

To conclude, remember that the covariance of the residuals is equal to zero and therefore:

$$\sigma_{ij} = b_{i1}b_{j1}\sigma_{I1}^2 + b_{i2}b_{j2}\sigma_{I2}^2 + \cdots + b_{iL}b_{jL}\sigma_{IL}^2. \tag{25B.4}$$

Appendix 25C: Using Microsoft Excel to Calculate Optimal Weights of a Portfolio

This appendix will contain four portions. Part I will discuss matrix inversion method to solve a simultaneous equation system. Part II will discuss how to collect the data from Yahoo Finance for Johnson & Johnson, IBM, and Caterpillar. Part III will show how to use single-index model to calculate optimal weights for three securities portfolios. Part IV will discuss how to use Markowitz model to calculate optimal weights for three securities portfolios.

Part I Matrix Inversion

This part illustrates how to solve a simultaneous equation system by matrix inversion method. Given simultaneous equations as follows:

$$\begin{aligned} 3x_1 + 2x_2 + 4x_3 &= 32 \\ 2x_1 + 3x_3 &= 19 \\ 4x_1 + 2x_2 + 3x_3 &= 29 \end{aligned}$$

The simultaneous equations can be written as the matrix forms

$$\underbrace{\begin{bmatrix} 3 & 2 & 4 \\ 2 & 0 & 3 \\ 4 & 2 & 3 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} 32 \\ 19 \\ 29 \end{bmatrix}}_{K}.$$

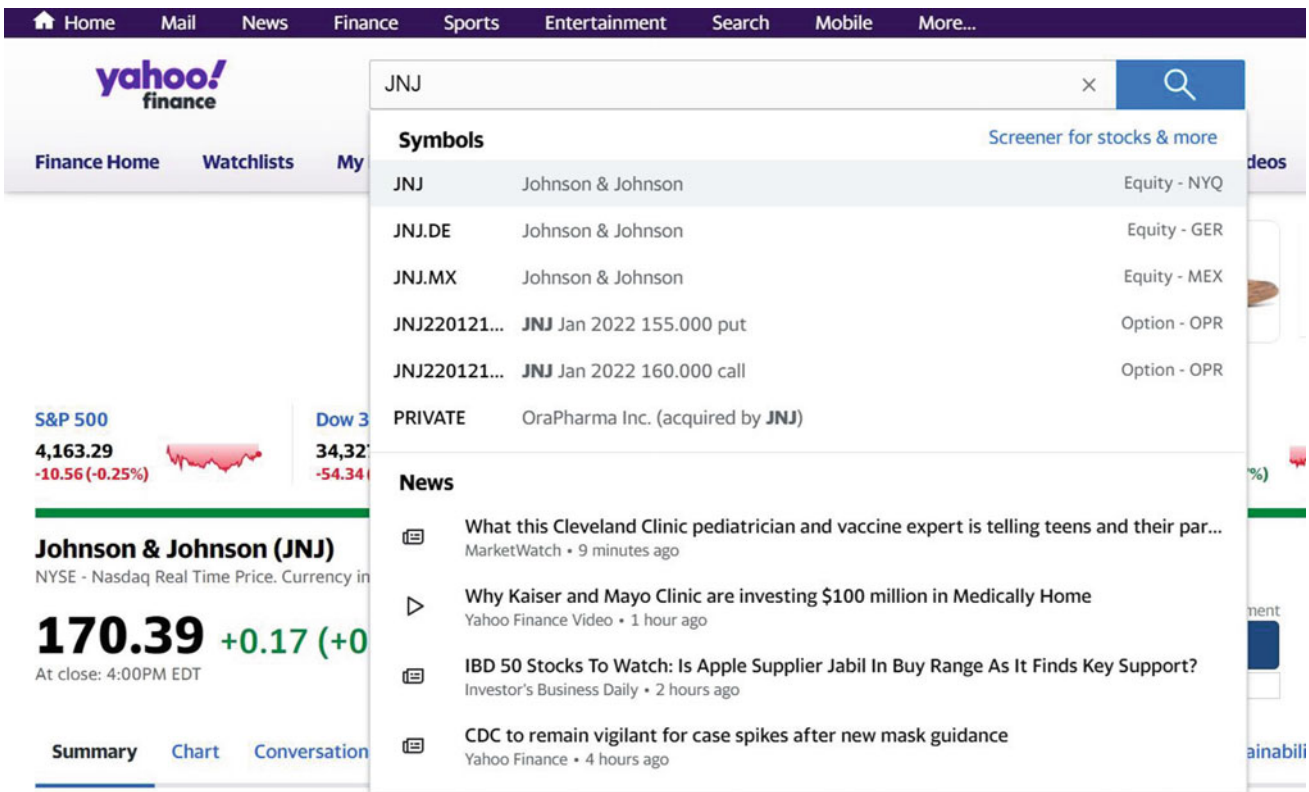
Then we can obtain the elements of matrix X by multiplying the inverse matrix A^{-1} and the matrix K. In the following part, we will illustrate the calculation of the inverse matrix in Excel.

Part II Data Collection

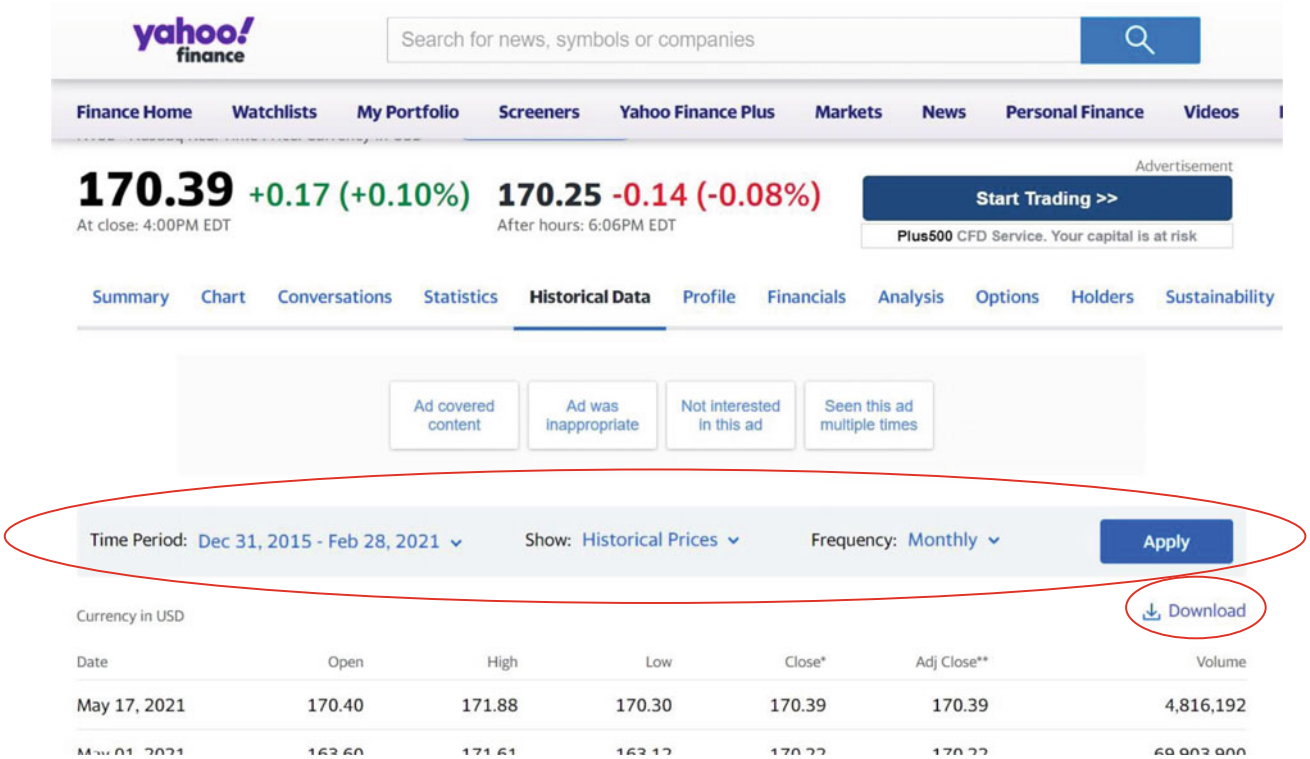
Here we use the monthly prices of Johnson &Johnson (JNJ), International Business Machines (IBM), Caterpillar (CAT), and S&P 500 index (^GSPC) from January 01, 2016 to March 01, 2021 as the example. The process includes data collection, the statistic analysis of samples, the results of the market model, and the calculation of the optimal weights of the Markowitz model.

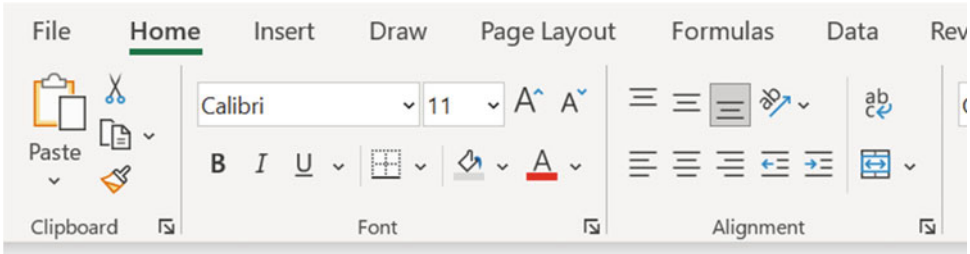
Data Collection

The collection of the monthly returns starts from the Yahoo Finance website: http://finance.yahoo.com/ and inputs the stock ticker "**JNJ**" to get quotes.



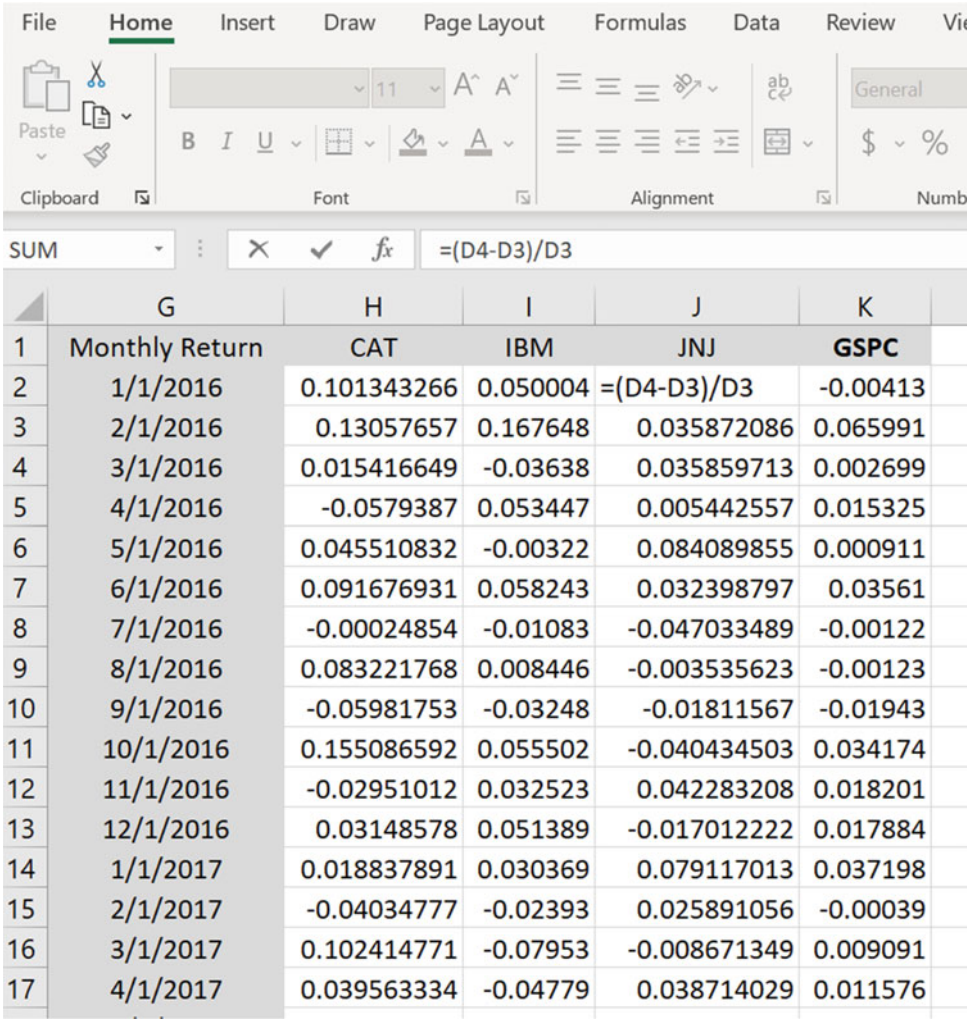
The second step is to get the stock prices: Select "**Historical Data**" on the top panel of the website, input the range of data from January 01, 2016 to March 01, 2021, select "**Monthly**" data and press "**Apply.**" After the webpage shows the historical prices, scroll down to the bottom of the webpage and select "**Download**," and then we can get the monthly prices of JNJ. Here we only need the data of the adjusted prices.





	A	B	C	D	E	F
1	Adj. Close	CAT	IBM	JNJ	**GSPC**	
2	12/1/2015	58.18733	109.2424	89.20029	2043.94	
3	1/1/2016	53.28988	99.05798	90.69392	1940.24	
4	2/1/2016	58.69045	104.0113	91.36257	1932.23	
5	3/1/2016	66.35404	121.4486	94.63994	2059.74	
6	4/1/2016	67.377	117.0301	98.0337	2065.3	
7	5/1/2016	63.47326	123.285	98.56725	2096.95	
8	6/1/2016	66.36198	122.8878	106.8558	2098.86	
9	7/1/2016	72.44585	130.0451	110.3178	2173.6	
10	8/1/2016	72.42784	128.6363	105.1291	2170.95	
11	9/1/2016	78.45541	129.7228	104.7574	2168.27	
12	10/1/2016	73.76241	125.5089	102.8597	2126.15	
13	11/1/2016	85.20197	132.4749	98.7006	2198.81	
14	12/1/2016	82.68765	136.7833	102.874	2238.83	
15	1/1/2017	85.29113	143.8125	101.1239	2278.87	
16	2/1/2017	86.89784	148.1799	109.1245	2363.64	
17	3/1/2017	83.3917	144.6339	111.9498	2362.72	

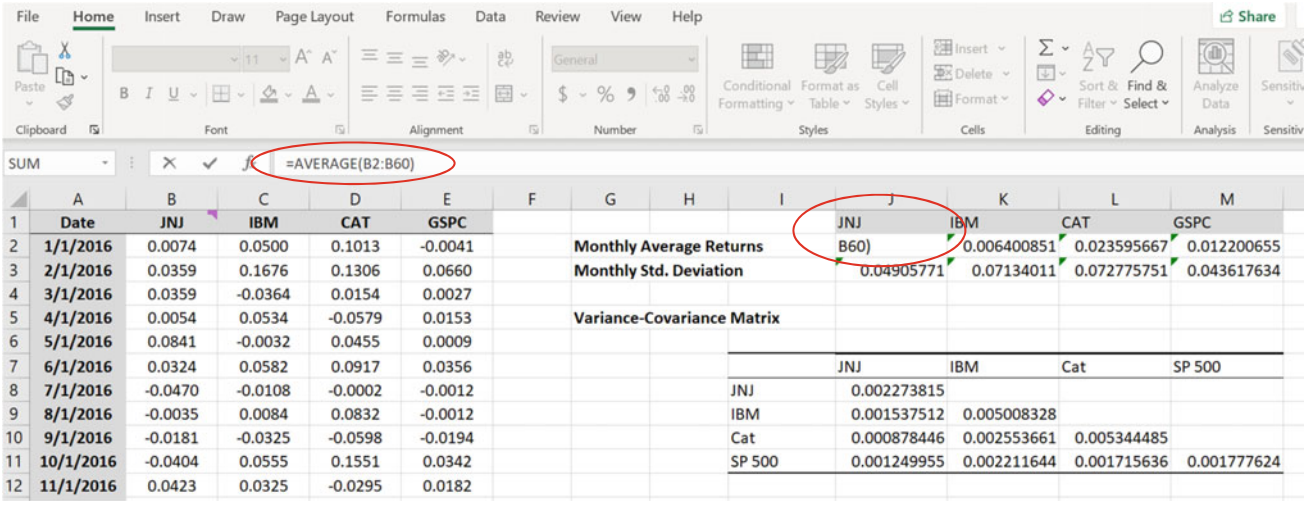
After repeat the second step to get the monthly prices of IBM, CAT, and S&P 500 index, the final step is the calculation of the monthly returns. The monthly returns at time t is the value of the price at time t minus the price at time $t-1$ divided by the price at time t. In excel, use the formula "**= (D3-D2)/D2**" to get JNJ's monthly return, and then in the same way we can get the monthly returns of the individual securities and S&P 500 index.



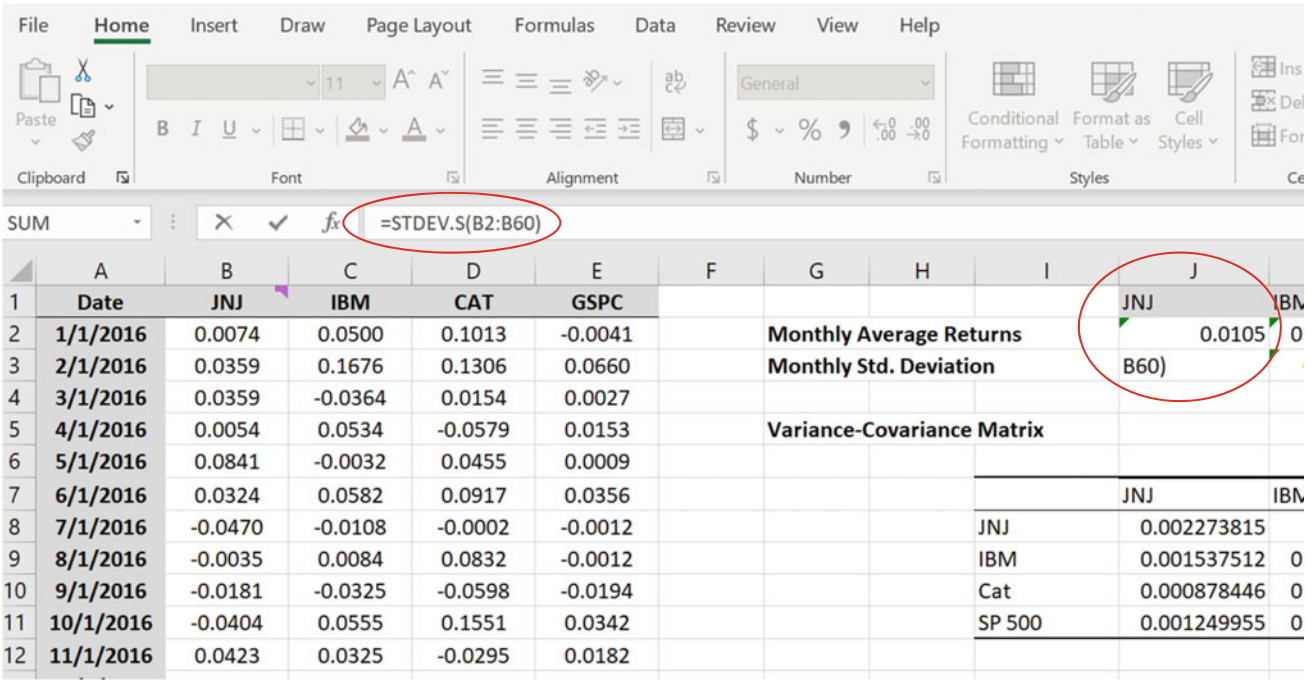
	G	H	I	J	K
1	Monthly Return	CAT	IBM	JNJ	**GSPC**
2	1/1/2016	0.101343266	0.050004	=(D4-D3)/D3	-0.00413
3	2/1/2016	0.13057657	0.167648	0.035872086	0.065991
4	3/1/2016	0.015416649	-0.03638	0.035859713	0.002699
5	4/1/2016	-0.0579387	0.053447	0.005442557	0.015325
6	5/1/2016	0.045510832	-0.00322	0.084089855	0.000911
7	6/1/2016	0.091676931	0.058243	0.032398797	0.03561
8	7/1/2016	-0.00024854	-0.01083	-0.047033489	-0.00122
9	8/1/2016	0.083221768	0.008446	-0.003535623	-0.00123
10	9/1/2016	-0.05981753	-0.03248	-0.01811567	-0.01943
11	10/1/2016	0.155086592	0.055502	-0.040434503	0.034174
12	11/1/2016	-0.02951012	0.032523	0.042283208	0.018201
13	12/1/2016	0.03148578	0.051389	-0.017012222	0.017884
14	1/1/2017	0.018837891	0.030369	0.079117013	0.037198
15	2/1/2017	-0.04034777	-0.02393	0.025891056	-0.00039
16	3/1/2017	0.102414771	-0.07953	-0.008671349	0.009091
17	4/1/2017	0.039563334	-0.04779	0.038714029	0.011576

The Statistic Analysis of Samples

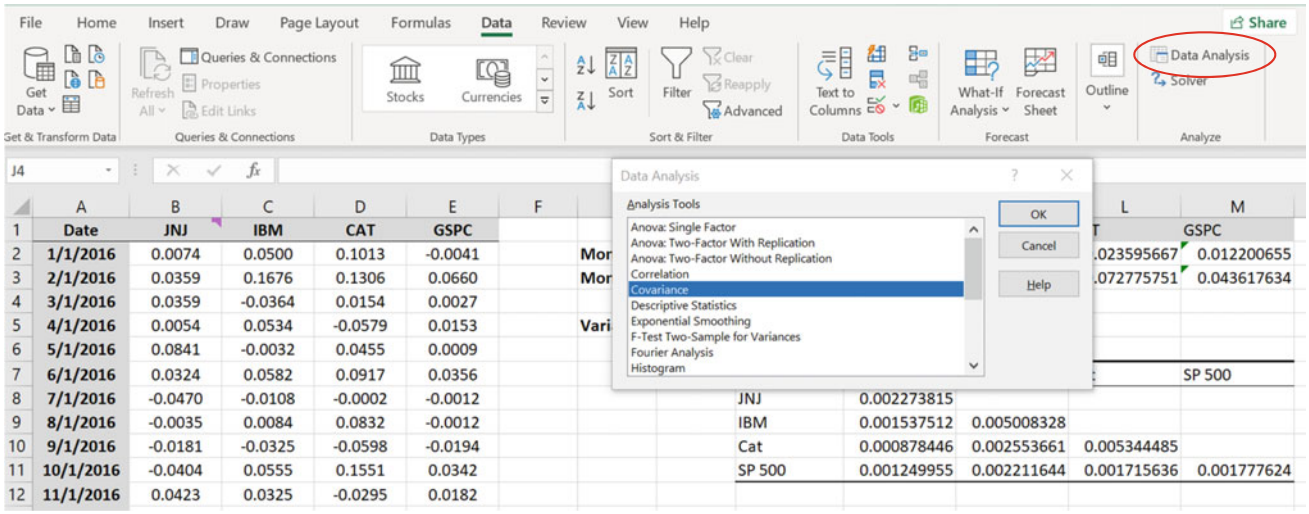
Copy the monthly returns of JNJ, IBM, CAT, and S&P 500 index into new sheet. The values of average and variance are calculated by "**AVERAGE**" and "**VARP**" functions and the covariance matrix is calculated by "**Covariance**" in "**Data Analysis**".

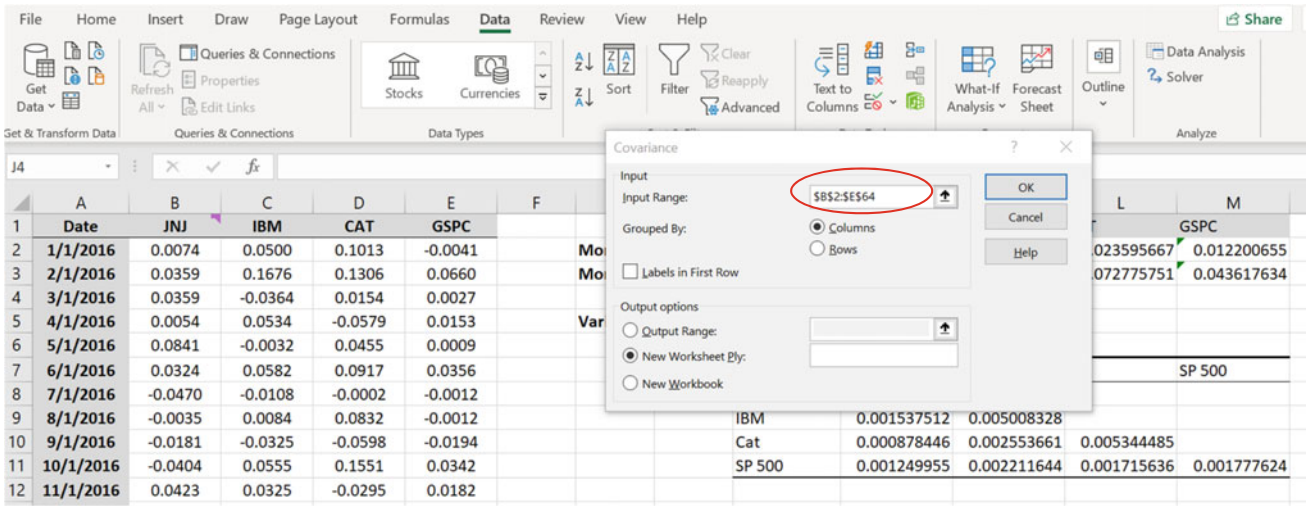


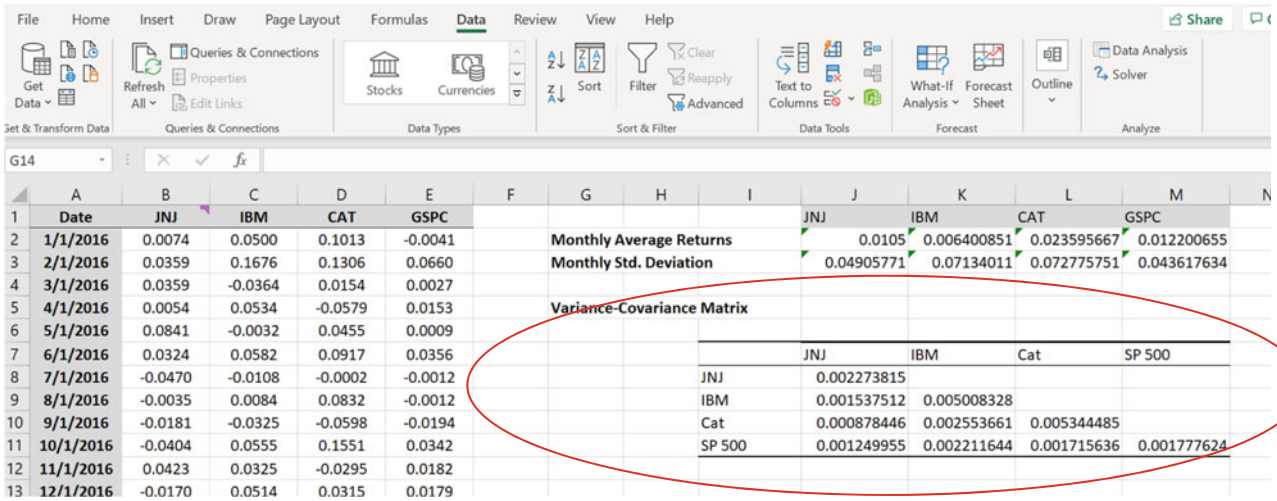
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date	JNJ	IBM	CAT	GSPC					JNJ	IBM	CAT	GSPC
2	1/1/2016	0.0074	0.0500	0.1013	-0.0041		Monthly Average Returns			B60)	0.006400851	0.023595667	0.012200655
3	2/1/2016	0.0359	0.1676	0.1306	0.0660		Monthly Std. Deviation			0.04905771	0.07134011	0.072775751	0.043617634
4	3/1/2016	0.0359	-0.0364	0.0154	0.0027								
5	4/1/2016	0.0054	0.0534	-0.0579	0.0153		Variance-Covariance Matrix						
6	5/1/2016	0.0841	-0.0032	0.0455	0.0009								
7	6/1/2016	0.0324	0.0582	0.0917	0.0356					JNJ	IBM	Cat	SP 500
8	7/1/2016	-0.0470	-0.0108	-0.0002	-0.0012				JNJ	0.002273815			
9	8/1/2016	-0.0035	0.0084	0.0832	-0.0012				IBM	0.001537512	0.005008328		
10	9/1/2016	-0.0181	-0.0325	-0.0598	-0.0194				Cat	0.000878446	0.002553661	0.005344485	
11	10/1/2016	-0.0404	0.0555	0.1551	0.0342				SP 500	0.001249955	0.002211644	0.001715636	0.001777624
12	11/1/2016	0.0423	0.0325	-0.0295	0.0182								



	A	B	C	D	E	F	G	H	I	J	
1	Date	JNJ	IBM	CAT	GSPC					JNJ	IBM
2	1/1/2016	0.0074	0.0500	0.1013	-0.0041		Monthly Average Returns			0.0105	0
3	2/1/2016	0.0359	0.1676	0.1306	0.0660		Monthly Std. Deviation			B60)	
4	3/1/2016	0.0359	-0.0364	0.0154	0.0027						
5	4/1/2016	0.0054	0.0534	-0.0579	0.0153		Variance-Covariance Matrix				
6	5/1/2016	0.0841	-0.0032	0.0455	0.0009						
7	6/1/2016	0.0324	0.0582	0.0917	0.0356					JNJ	IBM
8	7/1/2016	-0.0470	-0.0108	-0.0002	-0.0012				JNJ	0.002273815	
9	8/1/2016	-0.0035	0.0084	0.0832	-0.0012				IBM	0.001537512	0
10	9/1/2016	-0.0181	-0.0325	-0.0598	-0.0194				Cat	0.000878446	0
11	10/1/2016	-0.0404	0.0555	0.1551	0.0342				SP 500	0.001249955	0
12	11/1/2016	0.0423	0.0325	-0.0295	0.0182						

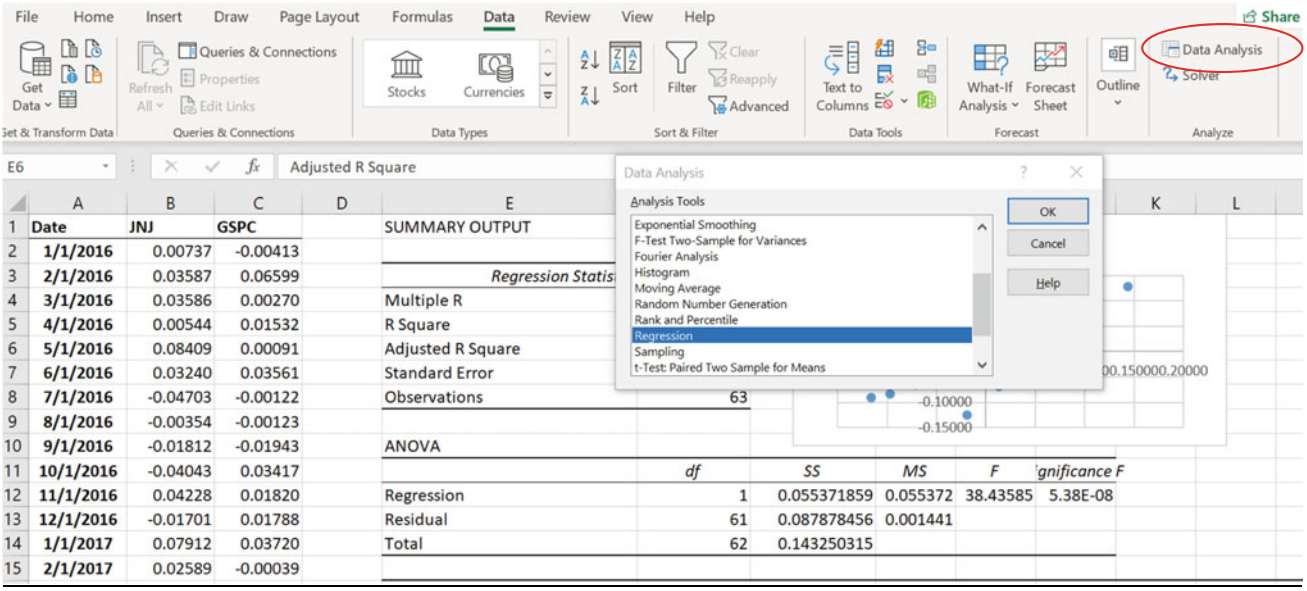

File Home Insert Draw Page Layout Formulas Data Review View Help
Share
Get Data
Queries & Connections
Properties
Edit Links
Refresh All
Stocks
Currencies
Sort
Filter
Clear
Reapply
Advanced
Text to Columns
What-If Analysis
Forecast Sheet
Outline
Data Analysis
Solver
Get & Transform Data
Queries & Connections
Data Types
Sort & Filter
Data Tools
Forecast
Analyze
J4
Data Analysis
Analysis Tools
Anova: Single Factor
Anova: Two-Factor With Replication
Anova: Two-Factor Without Replication
Correlation
Covariance
Descriptive Statistics
Exponential Smoothing
F-Test Two-Sample for Variances
Fourier Analysis
Histogram
OK
Cancel
Help
Date JNJ IBM CAT GSPC
1/1/2016 0.0074 0.0500 0.1013 -0.0041
2/1/2016 0.0359 0.1676 0.1306 0.0660
3/1/2016 0.0359 -0.0364 0.0154 0.0027
4/1/2016 0.0054 0.0534 -0.0579 0.0153
5/1/2016 0.0841 -0.0032 0.0455 0.0009
6/1/2016 0.0324 0.0582 0.0917 0.0356
7/1/2016 -0.0470 -0.0108 -0.0002 -0.0012
8/1/2016 -0.0035 0.0084 0.0832 -0.0012
9/1/2016 -0.0181 -0.0325 -0.0598 -0.0194
10/1/2016 -0.0404 0.0555 0.1551 0.0342
11/1/2016 0.0423 0.0325 -0.0295 0.0182
GSPC
0.012200655
0.043617634
SP 500
JNJ 0.002273815
IBM 0.001537512 0.005008328
Cat 0.000878446 0.002553661 0.005344485
SP 500 0.001249955 0.002211644 0.001715636 0.001777624


File Home Insert Draw Page Layout Formulas Data Review View Help
Share
Get Data
Queries & Connections
Properties
Edit Links
Refresh All
Stocks
Currencies
Sort
Filter
Clear
Reapply
Advanced
Text to Columns
What-If Analysis
Forecast Sheet
Outline
Data Analysis
Solver
Get & Transform Data
Queries & Connections
Data Types
Analyze
J4
Covariance
Input
Input Range: $B$2:$E$64
Grouped By: Columns Rows
Labels in First Row
Output options
Output Range:
New Worksheet Ply:
New Workbook
OK
Cancel
Help
Date JNJ IBM CAT GSPC
1/1/2016 0.0074 0.0500 0.1013 -0.0041
2/1/2016 0.0359 0.1676 0.1306 0.0660
3/1/2016 0.0359 -0.0364 0.0154 0.0027
4/1/2016 0.0054 0.0534 -0.0579 0.0153
5/1/2016 0.0841 -0.0032 0.0455 0.0009
6/1/2016 0.0324 0.0582 0.0917 0.0356
7/1/2016 -0.0470 -0.0108 -0.0002 -0.0012
8/1/2016 -0.0035 0.0084 0.0832 -0.0012
9/1/2016 -0.0181 -0.0325 -0.0598 -0.0194
10/1/2016 -0.0404 0.0555 0.1551 0.0342
11/1/2016 0.0423 0.0325 -0.0295 0.0182
GSPC
0.012200655
0.043617634
SP 500
IBM 0.001537512 0.005008328
Cat 0.000878446 0.002553661 0.005344485
SP 500 0.001249955 0.002211644 0.001715636 0.001777624


File Home Insert Draw Page Layout Formulas Data Review View Help
Share
Get Data
Queries & Connections
Properties
Edit Links
Refresh All
Stocks
Currencies
Sort
Filter
Clear
Reapply
Advanced
Text to Columns
What-If Analysis
Forecast Sheet
Outline
Data Analysis
Solver
Get & Transform Data
Queries & Connections
Data Types
Sort & Filter
Data Tools
Forecast
Analyze
G14
Date JNJ IBM CAT GSPC
1/1/2016 0.0074 0.0500 0.1013 -0.0041
2/1/2016 0.0359 0.1676 0.1306 0.0660
3/1/2016 0.0359 -0.0364 0.0154 0.0027
4/1/2016 0.0054 0.0534 -0.0579 0.0153
5/1/2016 0.0841 -0.0032 0.0455 0.0009
6/1/2016 0.0324 0.0582 0.0917 0.0356
7/1/2016 -0.0470 -0.0108 -0.0002 -0.0012
8/1/2016 -0.0035 0.0084 0.0832 -0.0012
9/1/2016 -0.0181 -0.0325 -0.0598 -0.0194
10/1/2016 -0.0404 0.0555 0.1551 0.0342
11/1/2016 0.0423 0.0325 -0.0295 0.0182
12/1/2016 -0.0170 0.0514 0.0315 0.0179
Monthly Average Returns JNJ 0.0105 IBM 0.006400851 CAT 0.023595667 GSPC 0.012200655
Monthly Std. Deviation JNJ 0.04905771 IBM 0.07134011 CAT 0.072775751 GSPC 0.043617634
Variance-Covariance Matrix
JNJ IBM Cat SP 500
JNJ 0.002273815
IBM 0.001537512 0.005008328
Cat 0.000878446 0.002553661 0.005344485
SP 500 0.001249955 0.002211644 0.001715636 0.001777624

Part III The Market Model

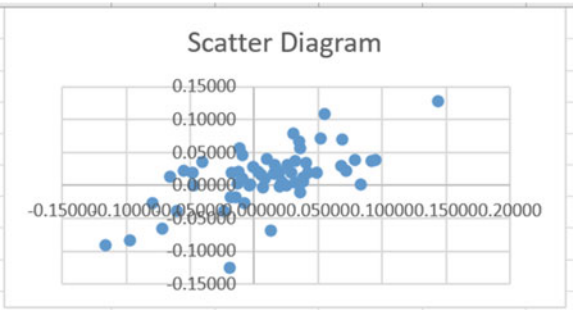
Use "**Regression**" function in "**Data Analysis**" to run the market model with respect to JNJ's returns, select the monthly returns of JNJ in "**Y range**" and the monthly returns of S&P 500 index in "**X range**," and press "**OK**." We then can get the regression results of market model with respect to JNJ in new sheet. The coefficients of intercept term and S&P 500 index are "**Alpha**" and "**Beta**" of market model with respect to JNJ's return. The MS of residual is the "**Variance of residual**" of the market model. Repeat the "**Regression**" function, and then we can get the results of market model with respect to the monthly return of IBM and CAT.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Date	JNJ	GSPC		SUMMARY OUTPUT							
2	1/1/2016	0.00737	-0.00413									
3	2/1/2016	0.03587	0.06599		*Regression Statis*							
4	3/1/2016	0.03586	0.00270		Multiple R							
5	4/1/2016	0.00544	0.01532		R Square							
6	5/1/2016	0.08409	0.00091		Adjusted R Square							
7	6/1/2016	0.03240	0.03561		Standard Error							
8	7/1/2016	-0.04703	-0.00122		Observations	63						
9	8/1/2016	-0.00354	-0.00123									
10	9/1/2016	-0.01812	-0.01943		ANOVA							
11	10/1/2016	-0.04043	0.03417			*df*	*SS*	*MS*	*F*	*gnificance F*		
12	11/1/2016	0.04228	0.01820		Regression	1	0.055371859	0.055372	38.43585	5.38E-08		
13	12/1/2016	-0.01701	0.01788		Residual	61	0.087878456	0.001441				
14	1/1/2017	0.07912	0.03720		Total	62	0.143250315					
15	2/1/2017	0.02589	-0.00039									

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.621722749
R Square	0.386539177
Adjusted R Square	0.376482442
Standard Error	0.037955638
Observations	63



ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	1	0.055371859	0.055372	38.43585	5.38E-08
Residual	61	0.087878456	0.001441		
Total	62	0.143250315			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	0.001673129	0.004988496	0.335397	0.738476	-0.0083	0.011648	-0.0083	0.011648
GSPC Returns	0.70316021	0.113419056	6.199666	5.38E-08	0.476365	0.929956	0.476365	0.929956

Part IV Markowitz Model

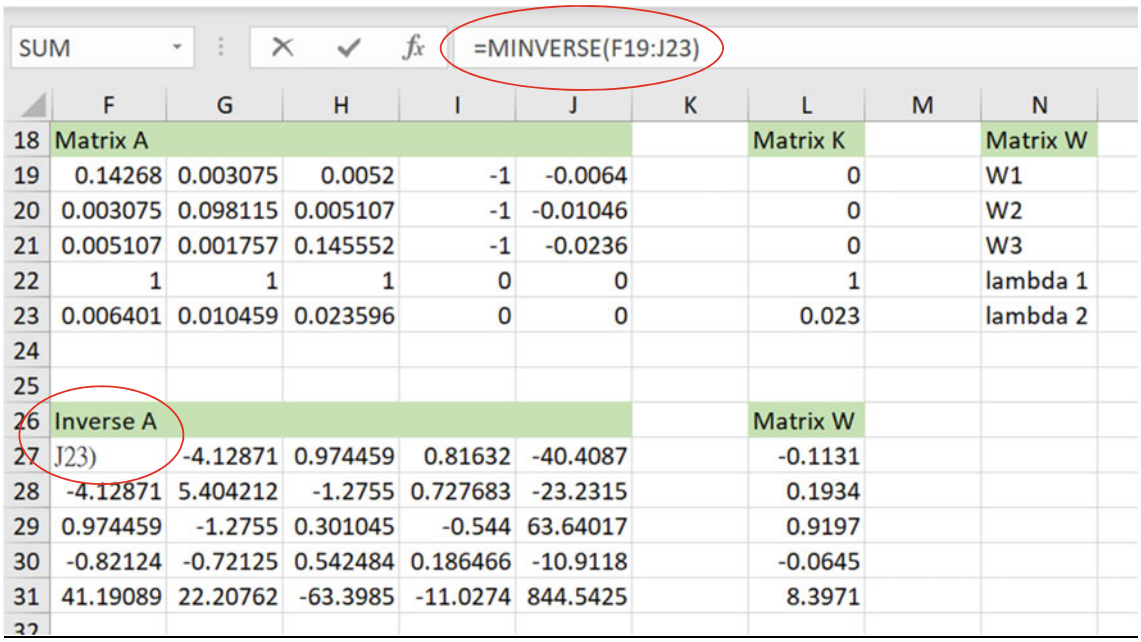
In Markowitz Model, we know

$$\underbrace{\begin{bmatrix} & & & 1 & E(R_1) \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & 1 & E(R_2) \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & 1 & E(R_3) \\ 1 & 1 & 1 & 0 & 0 \\ E(R_1) & E(R_2) & E(R_3) & 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ E* \end{bmatrix}}_{K}.$$

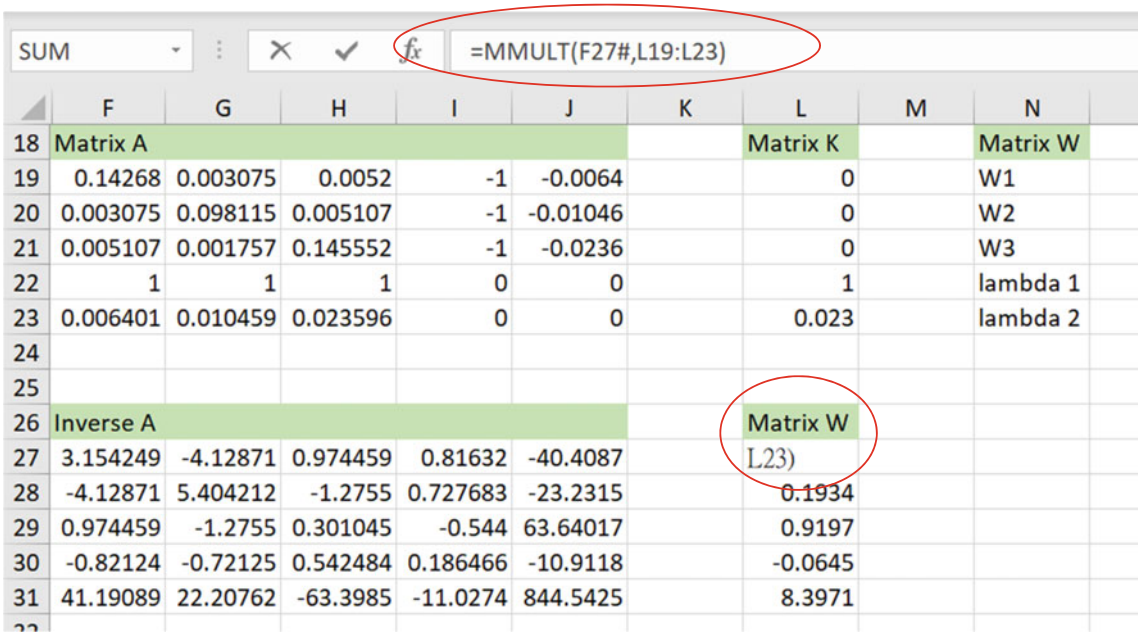
According to the statistical analysis of samples in the prior step, we can get the value of each element of Augmented matrix A. From the matrix calculation of Markowitz Model, we can get the optimal weights by multiplying A^{-1} and K.

$$\underbrace{\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ \lambda_1 \\ \lambda_2 \end{bmatrix}}_{W} = \underbrace{\begin{bmatrix} 2\sigma_{11} & 2\sigma_{12} & 2\sigma_{13} & 1 & E(R_1) \\ 2\sigma_{21} & 2\sigma_{22} & 2\sigma_{23} & 1 & E(R_2) \\ 2\sigma_{31} & 2\sigma_{32} & 2\sigma_{33} & 1 & E(R_3) \\ 1 & 1 & 1 & 0 & 0 \\ E(R_1) & E(R_2) & E(R_3) & 0 & 0 \end{bmatrix}^{-1}}_{A^{-1}} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ E* \end{bmatrix}}_{K}.$$

To calculate the inverse matrix of matrix A, select a range from F27 to J31 and type " **= Minverse(F19:J23)**" and press "**Ctrl + Shift + Enter**," then we can get the inverse matrix of matrix A.



Input the value of each element of matrix K where $E*$ is the required return of investor. To multiply the matrixes A^{-1} and K, select a range from L27 to L31 and type " **= MMULT(F27#,L19:L23).**" Press "**Ctrl + Shift + Enter,**" then we can get the optimal weight matrix W with short selling.



Finally, under $E^* = 0.023$, we can calculate the optimal weight for JNJ, IBM, and CAT in excel program. $W_1 = -0.1131, W_2 = 0.1934,$ *and* $W_3 = 0.9197$

These results show that when the target rate of return is extremely high, we might even need to short the security with the very low expected rate of return.

Bibliography

Beaver, W., P. Kettler and M. Scholes. "The Association between Market Determined and Accounting Determined Risk Measures." *Thcxse Accounting Review*, v. 45 (1970), pp. 654–682.

Blume, M. "Betas and Their Regression Tendencies." *Journal of Finance*, v. 20 (1975), pp. 785–795.

Brenner, M. "On the Stability of the Distribution of the Market Component in Stock Price Changes." *Journal of Financial and Quantitative Analysis*, v. 9 (1974), pp. 945–961.

Elton, E. J., M. J. Gruber and T. Urich. "Are Betas Best?" *Journal of Finance*, v. 23 (1978), pp. 1375–1384.

Elton, E. J., M. J. Gruber, S. J. Brown and W. N. Goetzmann. Modern Portfolio Theory and Investment Analysis, 9th ed. New York: John Wiley & Sons, (2014).

Fouse, W., W. Jahnke and B. Rosenberg. "Is Beta Phlogiston?" *Financial Analysts Journal*, v. 30 (January/February 1974), pp. 70–80.

Francis, J.C. and D. Kim. Modern Portfolio Theory: Foundations, Analysis, and New Developments, 1st Edition. New York: John Wiley and Sons, Inc., (2013).

Frankfurter, G. and H. Phillips. "Alpha-Beta Theory: A Word of Caution. "*Journal of Financial Management*, v. 3 (Summer 1977), pp. 35–40.

Frankfurter, G. and J. Seagle. "Performance of the Sharpe Portfolio Selection Model: A Comparison." *Journal of Financial and Quantitative Analysis*, v. 11 (1976), pp. 195–204.

Gibbons, M. R. "Multivariate Tests of Financial Models, a New Approach." *Journal of Financial Economics*, v. 81 (1982), pp. 3–27.

Guerard, J. B. Handbook of Portfolio and Construction: Contemporary Applications of Markowitz Techniques. New York: Springer, (2081).

Haugen, R. and D. Wichern. "The Intricate Relationship between Financial Leverage and the Stability of Stock Prices. " *Journal of Finance*, v. 20 (December 1975), pp. 1283–1292.

Jacob, N. "A Limited-Diversification Portfolio Selection Model for the Small Investor." *Journal of Finance*, v. 19 (1974), pp. 847–856.

King, B. "Market and Industry Factors in Stock Price Behavior." *Journal of Business*, v. 39 (1966), pp. 139–140.

Lee, C. F., A. C. Lee and J. C. Lee . *Handbook of Quantitative Finance and Risk Management*. New York: Springer, (2081).

Lee, A. C., J. C. Lee and C. F. Lee. *Financial Analysis and Planning and Forecasting: Theory and Application.* Singapore: World Scientific, (2009).

Lee, C.F., J. Finnerty, J. Lee, A.C. Lee, and D. Wort. Security Analysis, Portfolio Management, and Financial Derivatives. Singapore: World Scientific, (2013).

Latane, H., D. Tuttle and A. Young. "How to Choose a Market Index." *Financial Analysts Journal*, v. 27 (September/October 1971), pp. 75–85.

Levy, R. "Beta Coefficients as Predictors of Return." *Financial Analysts Journal*, v. 30 (January/February 1974), pp. 61–69.

Markowitz, H. *Portfolio Selection*. Cowles Foundation Monograph 16. New York: John Wiley & Sons, Inc., (1959).

Morgan, I. G. "Grouping Procedures for Portfolio Formation." *Journal of Finance*, v. 21 (December 1977), pp. 1759–1765.

Roll, R. "Bias in Fitting the Sharpe Model to Time Series Data." *Journal of Financial and Quantitative Analysis*, v. 4 (1969), pp. 271–289.
Rosenberg, B. and J. Guy. "Prediction of Beta from Investment Fundamentals." *Financial Analysts Journal*, v. 32 (May/June 1976a), pp. 60–72.
Rosenberg, B. and J. Guy. "Prediction of Beta from Investment Fundamentals, Part II." *Financial Analysts Journal*, v. 32 (July/August 1976b), pp. 62–70.
Rosenberg, B. and W. McKibben. "The Prediction of Systematic and Specific Risk in Common Stocks." *Journal of Financial and Quantitative Analysis*, v. 8 (1973), pp. 317–333.
Rosenberg, B. and V. Marathe. "The Prediction of Investment Risk: Systematic and Residual Risk." Berkeley Working Paper Series, (1974).
Ross, S. A. "Arbitrage Theory of Capital-Asset Pricing." *Journal of Economic Theory*, v. 8 (1976), pp. 341–360.
Ross, S. A. "Return, Risk and Arbitrage," in Friend and J. L. Bicksler (eds.), *Risk and Return in Finance*, Vol. 1, pp. 187–208. Cambridge, MA: Ballinger, (1977).
Sharpe, W. F. "A Linear Programming Algorithm for Mutual Fund Portfolio Selection." *Management Science*, v. 13 (1967), pp. 499–581.
Sharpe, W. F. *Portfolio Theory and Capital Markets*. New York: McGraw-Hill Book Co., (1970).
Smith, K. V. "Stock Price and Economic Indexes for Genssserating Efficient Portfolios." *Journal of Business*, v. 42 (1969), pp. 326–335.
Stone, B. "A Linear Programming Formulation of the General Portfolio Selection Problem." *Journal of Financial and Quantitative Analysis*, v. 8 (1973), pp. 621–636.

Investment Performance Approach to Portfolio Selection

26

26.1 Introduction

The main purpose of this chapter is to show how Sharpe's performance measure and Treynor's performance measure can be used in portfolio selection. These methods can be used as an alternative to Markowitz's (1952, 1959) full variance–covariance approach to determine optimal weights of a portfolio. In addition, these approaches can be used as an alternative to Sharpe's index model approach. This chapter assumes the existence of a risk-free borrowing and lending rate and advances one step further to simplify the calculation of the optimal weights of a portfolio and the efficient frontier. First discussed are Lintner's (1965) and Elton et al.'s (1976) Sharpe performance measure approaches for determining the efficient frontier with short sales allowed. We next present the scenario that short sales are not allowed with weighting upper bounds to ensure the feasibility of the optimal portfolio. This is followed by a discussion of the Treynor performance measure approach for determining the efficient frontier with **short sales allowed**. The Treynor-measure approach is then analyzed for determining the efficient frontier with **short sales not allowed**. And finally, Dow-Jones 30 data from January 2003 to December 2007 are employed to demonstrate how the Treynor method can be applied in the real world. Overall, this chapter relates the performance measure concepts and methods to the portfolio-selection models and makes more accessible the insights of optimal portfolio selection.

In Sect. 26.2 we discuss the Sharpe performance measure approach with short sales allowed. Section 26.3 examines the Treynor-measure approach with short sales allowed, while the following section, Sect. 26.4 discusses the Treynor-measure approach with short sales not allowed. Section 26.5 talks about the impact of short sales on optimal weight determination. The economic rationale of the Treynor performance measure method is examined in Sect. 26.6, and finally in Sect. 26.7 the chapter is summarized.

26.2 Sharpe Performance-Measure Approach with Short Sales Allowed

In deriving the capital asset pricing model (CAPM), Lintner (1965) suggests a performance measure approach for determining the efficient frontier discussed in previous chapters. Lintner arrived at this approach through a sequence of logical steps.

Following previous chapters, the objective function for portfolio selection can be expressed:

$$\begin{aligned}\text{Max } L = \sum_{i=1}^{n} W_i \overline{R}_i + \lambda_1 \left\{ \left[\sum_{j=1}^{n} \sum_{i=1}^{n} \text{Cov}\left(R_i, R_j\right) \right]^{1/2} - \sigma_p \right\}, \\ + \lambda_2 \left(\sum_{i=1}^{n} W_i - 1 \right)\end{aligned} \tag{26.1}$$

where

$\overline{R}_i$ = average rates of return for security i;

W_i (or W_j) = the optimal weight for ith (or jth) security;

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

J. Lee and C.-F. Lee, *Essentials of Excel VBA, Python, and R*,
https://doi.org/10.1007/978-3-031-14236-9_26

$\mathrm{Cov}(R_i, R_j)$ = the covariance between R_i and R_j;
σ_p = the standard deviation of a portfolio; and.
λ_1, λ_2 = Lagrangian multipliers.

Markowitz portfolio-selection model minimizes the variance given the targeted expected rate of return. Equation (26.1) maximizes the expected rates of return given the targeted standard deviation.

If a constant risk-free borrowing and lending rate R_f is subtracted from Eq. (26.1):

$$\text{Max } L' = \sum_{i=1}^{n} \left(W_i \overline{R}_i - R_f\right) + \lambda_1 \left\{ \left[\sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \mathrm{Cov}(R_i, R_j) \right]^{1/2} - \sigma_p \right\} + \lambda_2 \left(\sum_{i=1}^{n} W_i - 1 \right) \tag{26.2a}$$

Equations (26.1) and (26.2a), both formulated as a constrained maximization problem, can be used to obtain optimum portfolio weights $W_i (i = 1, 2, \ldots, n)$. Since R_f is a constant, the optimum weights obtained from Eq. (26.1) will be equal to those obtained for Eq. (26.2a). Previous chapters used the methodology of **Lagrangian multipliers**; it can be shown that Eq. (26.2a) can be replaced by a nonconstrained maximization method as follows. Incorporating the constant $\sum_{i=1}^{n} W_i - 1$ into the objective function by substituting

$$R_f = (1)R_f = \left(\sum_{i=1}^{n} W_i \right) R_f = \sum_{i=1}^{n} W_i R_f$$

into Eq. (26.2a):

$$\text{Max } L' = \sum_{i=1}^{n} W_i \left(\overline{R}_i - R_f\right) + \lambda_1 \left[\left(\sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \mathrm{Cov}(R_i, R_j) \right)^{1/2} - \sigma_p \right] \tag{26.2b}$$

A two-Lagrangian multiplier problem has been reduced to a one-Lagrangian problem as indicated in Eq. (26.2b). By using a special property of the relationship between $\sum_{i=1}^{n} \left(\overline{R}_i - R_f\right)$ and $\left(\sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \mathrm{Cov}(R_i, R_j) \right)^{1/2}$ the constrained optimization of Eq. (26.2b) can be reduced to an unconstrained optimization problem, as indicated in Eq. (26.3)[1]

$$\text{Max } L = \frac{\sum_{i=1}^{n} W_i \left(\overline{R}_i - R_f\right)}{\left(\sum_{i=1}^{n} W_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma_{ij} \right)^{1/2}} \quad (i \neq j), \tag{26.3}$$

where $\sigma_{ij} = \mathrm{Cov}(R_i, R_j)$. Alternatively, the objective function of Eq. (26.3) can be developed as follows. This ratio L is equal to excess average rates of return for the ith portfolio divided by the standard deviation of the ith portfolio. This is a **Sharpe performance measure**.

Following Sharpe (1964) and Lintner (1965), if there is a risk-free lending and borrowing rate (R_f) and short sales are allowed, then the efficient frontier (efficient set) will be linear, as discussed in previous chapters. In terms of return (R_p) standard deviation σ_p space, this linear efficient frontier is indicated as line $R_f E$ in Fig. 26.1. *AEC* represents a feasible investment opportunity in terms of existing securities to be included in the portfolio when there is no risk-free lending and borrowing rate. If there is a risk-free lending and borrowing rate, then the efficient frontier becomes $R_f E$. An infinite number

[1] Since the ratio of Eq. (26.3) is homogeneous of degree zero with respect to W_i. In other words, the ratio L is unchanged by any proportionate change in the weight of W_i.

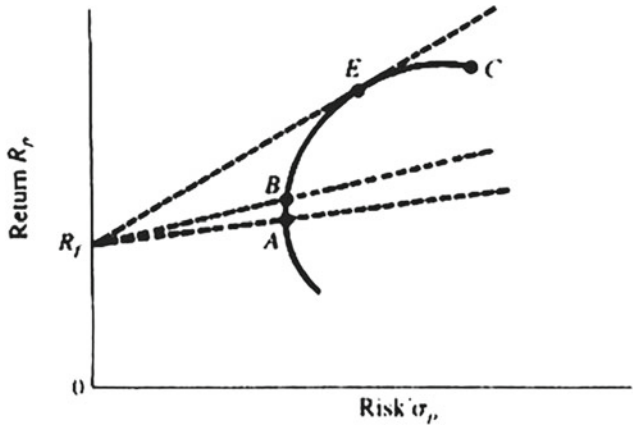


Fig. 26.1 Linear Efficient Frontier

of linear lines represent the combination of a riskless asset and risky portfolio, such as R_fA, R_fB, and R_fE. It is obvious that line R_fE has the highest slope, as represented by

$$\Theta = \frac{\overline{R}_p - R_f}{\sigma_p}, \tag{26.4}$$

in which $\overline{R}_p = \sum_{i=1}^{n} W_i \overline{R}_i$, R_f, and σ_p are defined as in Eq. (26.2a). Thus the efficient set is obtained by maximizing Θ. By imposing the constraint $\sum_{i=1}^{n} W_i = 1$, Eq. (26.4) is expressed:

$$\Theta' = \frac{\overline{R}_p - R_f}{\sigma_p} + \lambda\left(\sum_{i=1}^{n} W_i - 1\right). \tag{26.5a}$$

By using the procedure of deriving Eqs. (26.2b), (26.5a) becomes

$$\Theta' = \frac{\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)}{\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{i=1}^{n}\sigma_{ij}} \quad (i \neq j) \tag{26.5b}$$

This equation is equivalent to Eq. (26.3). This approach is used by Elton and Gruber (1987) to derive their objective function for optimal portfolio selection.[2]

Following the maximization procedure discussed earlier in previous chapters, it is clear that there are n unknowns to be solved in either Eq. (26.3) or Eq. (26.5b). Therefore, calculus must be employed to compute n **first-order conditions** to formulate a system of n simultaneous equations:

$$\begin{aligned} 1\ \frac{dL}{dW_1} &= 0 \\ 2\ \frac{dL}{dW_2} &= 0 \\ &\vdots \\ n\ \frac{dL}{dW_n} &= 0 \end{aligned}$$

[2] Elton et al. (2006).

From Appendix 26A, the n simultaneous equations used to solve H_i are

$$\begin{aligned}
\overline{R}_1 - R_f &= H_1\sigma_1^2 + H_2\sigma_{12} + H_3\sigma_{13} + \cdots + H_n\sigma_{1n} \\
\overline{R}_2 - R_f &= H_1\sigma_{12} + H_2\sigma_2^2 + H_3\sigma_{23} + \cdots + H_n\sigma_{2n} \\
&\vdots \\
\overline{R}_m - R_f &= H_1\sigma_{1n} + H_2\sigma_{2n} + H_3\sigma_{3n} + \cdots + H_n\sigma_n^2
\end{aligned} \tag{26.6a}$$

where

$H_i = kW_i\,(i = 1, 2, \ldots, n)$; and

$$k = -\frac{\overline{R}_p - R_f}{\sigma_p^2}.$$

The H_S is proportional to the optimum portfolio weight $W_i\,(i = 1, 2, \ldots, n)$ by a constant factor K. To determine the optimum weight W_i, H_i is first solved from the set of equations indicated in Eq. (26.6). Having done so, H_i must be called to calculate W_i, as indicated in Eq. (26.7).

$$W_i = \frac{H_i}{\sum_{i=1}^{n} H_i} \tag{26.7}$$

If there are only three securities, then Eq. (26.6a) reduces to:

$$\begin{aligned}
\overline{R}_1 - R_f &= H_1\sigma_1^2 + H_2\sigma_{12} + H_3\sigma_{13} \\
\overline{R}_2 - R_f &= H_1\sigma_{12} + H_2\sigma_2^2 + H_3\sigma_{23} \\
\overline{R}_3 - R_f &= H_1\sigma_{13} + H_2\sigma_{23} + H_3\sigma_3^2
\end{aligned} \tag{26.6b}$$

Sample Problem 26.1 provides further illustration.

Sample Problem 26.1

Let

$$\begin{aligned}
&\overline{R}_1 = 15\%\,\overline{R}_2 = 12\%\,\overline{R}_3 = 20\% \\
&\sigma_1 = 8\%\,\sigma_2 = 7\%\,\sigma_3 = 9\% \\
&r_{12} = 0.5\,r_{13} = 0.4\,r_{23} = 0.2 \\
&R_f = 8\%
\end{aligned}$$

Substituting this information into Eq. (26.6b):

$$\begin{aligned}
15 - 8 &= 64H_1 + (0.5)(8)(7)H_2 + (0.4)(8)(9)H_3 \\
12 - 8 &= (0.5)(8)(7)H_1 + 49H_2 + (0.2)(7)(9)H_3 \\
20 - 8 &= (0.4)(8)(9)H_1 + (0.2)(7)(9)H_2 + 81H_3
\end{aligned}$$

Simplifying:

$$\begin{aligned}
7 &= 64H_1 + 28H_2 + 28.8H_3 \\
4 &= 28H_1 + 49H_2 + 12.6H_3 \\
12 &= 28.8H_1 + 12.6H_2 + 81H_3
\end{aligned} \tag{26.6c}$$

Using Cramer's rule, H_1, H_2, and H_3 can be solved as follows:

$$H_1 = \frac{\begin{vmatrix} 7 & 28 & 28.8 \\ 4 & 49 & 12.6 \\ 12 & 12.6 & 81 \end{vmatrix}}{\begin{vmatrix} 64 & 28 & 28.8 \\ 28 & 49 & 12.6 \\ 28.8 & 12.6 & 81 \end{vmatrix}}$$
$$= \frac{(4233.6 + 1451.5 + 27783) - (1111.3 + 9072 + 16934.4)}{(10160.6 + 10160.6 + 254016) - (10160.6 + 63504 + 40642.6)}$$
$$= \frac{33648.1 - 27.1177}{160030} = \frac{6350.4}{160030} = 3.97\%$$

$$H_2 = \frac{\begin{vmatrix} 64 & 7 & 28.8 \\ 28 & 4 & 12.6 \\ 28.8 & 12 & 81 \end{vmatrix}}{\begin{vmatrix} 64 & 28 & 28.8 \\ 28 & 49 & 12.6 \\ 28.8 & 12.6 & 81 \end{vmatrix}}$$
$$= \frac{(2540.2 + 9676.6 + 20736) - (9676.8 + 15876 + 3317.8)}{160030}$$
$$= \frac{32952.8 - 28870.6}{160030} = 2.55\%$$

$$H_3 = \frac{\begin{vmatrix} 64 & 28 & 28.8 \\ 28 & 49 & 12.6 \\ 28.8 & 12.6 & 81 \end{vmatrix}}{\begin{vmatrix} 64 & 28 & 28.8 \\ 28 & 49 & 12.6 \\ 28.8 & 12.6 & 81 \end{vmatrix}}$$
$$= \frac{(3225.6 + 2469.6 + 37632) - (3225.6 + 9480 + 9878.4)}{160030}$$
$$= \frac{20815.2}{160030} = 13.01\%$$

Using Eq. (26.7), $W_1, W_2,$ and W_3 are obtained:

$$W_1 = \frac{H_1}{\sum_{i=1}^{3} H_i} = \frac{3.97}{3.97 + 2.55 + 13.01} = \frac{3.97}{18.53}$$
$$= 20.33\%$$
$$W_2 = \frac{H_2}{\sum_{i=1}^{3} H_i} = \frac{2.55}{19.53}$$
$$= 13.06\%$$
$$W_3 = \frac{H_3}{\sum_{i=1}^{3} H_i} = \frac{13.01}{19.53}$$
$$= 66.61\%$$

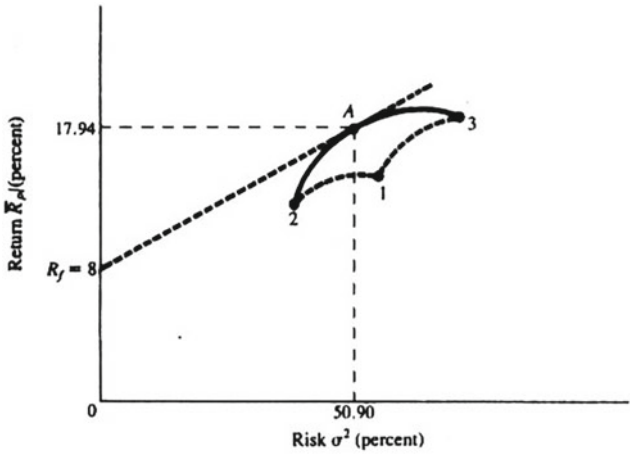


Fig. 26.2 Efficient frontier for Example 26.1

Here $\overline{R}_p$ and σ_p^2 can be calculated by employing these weights:

$$\begin{aligned}\overline{R}_p &= (15)(0.2033) + (12)(0.1306) + (20)(0.0061)\\ &= 3.049 + 1.5672 + 13.322\\ &= 17.9382\%\end{aligned}$$

$$\begin{aligned}\sigma_p^2 &= \sum_{i=1}^{3} W_i^2\sigma_i^2 + \sum_{i=1}^{3}\sum_{j=1}^{3} W_i W_j r_{ij}\sigma_i\sigma_j \; i \neq j\\ &= (0.2033)^2(64) + (0.1306)^2(49) + (0.6661)^2(81)\\ &\quad + 2(0.2033)(0.1306)(0.5)(8)(7)\\ &\quad + 2(0.2033)(0.6661)(0.4)(8)(9)\\ &\quad + 2(0.2)(0.1306)(0.6661)(7)(9)\\ &= 2.645 + 0.836 + 35.939 + 1.487 + 7.8 + 2.192\\ &= 50.899\%\end{aligned}$$

The efficient frontier for this example is shown in Fig. 26.2. Here A represents an efficient portfolio with $\overline{R}_p = 17.94\%$ and $\sigma_p^2 = 50.90\%$.

In addition to Cramer's rule method used in this example, we can also use the matrix inversion method to solve this question. Equation (26.6b) can be written in the matrix form as following:

$$\begin{bmatrix}\overline{R_1} - R_f\\ \overline{R_2} - R_f\\ \overline{R_3} - R_f\end{bmatrix} = \begin{bmatrix}\sigma_1^2 & \sigma_{12} & \sigma_{13}\\ \sigma_{21} & \sigma_2^2 & \sigma_{23}^2\\ \sigma_{31} & \sigma_{32} & \sigma_3^2\end{bmatrix}\begin{bmatrix}H_1\\ H_2\\ H_3\end{bmatrix}$$

Then Eq. (26.6c) can be written as following:

$$\begin{bmatrix}7\\ 4\\ 12\end{bmatrix} = \begin{bmatrix}64 & 28 & 28.8\\ 28 & 49 & 12.6\\ 28.8 & 12.6 & 81\end{bmatrix}\begin{bmatrix}H_1\\ H_2\\ H_3\end{bmatrix}$$

By using inverse matrix method in Appendix 10C of Chap. 10, we can obtain H_1, H_2, and H_3.

$$\begin{bmatrix}H_1\\ H_2\\ H_3\end{bmatrix} = \begin{bmatrix}64 & 28 & 28.8\\ 28 & 49 & 12.6\\ 28.8 & 12.6 & 81\end{bmatrix}^{-1}\begin{bmatrix}7\\ 4\\ 12\end{bmatrix} = \begin{bmatrix}0.024 & -0.012 & -0.007\\ -0.012 & 0.027 & 0\\ -0.007 & 0 & 0.015\end{bmatrix}\begin{bmatrix}7\\ 4\\ 12\end{bmatrix} = \begin{bmatrix}0.0397\\ 0.0255\\ 0.1301\end{bmatrix}.$$

The Sharpe performance measure method discussed in this section is the most popular method used in empirical studies of portfolio management. For example, Chiou et al. (2010) have used this method to study stock returns, risk, and legal environment around the world. The empirical results of this paper can be found in Chap. 22 of this book.

26.3 Sharpe Performance-Measure Approach with Short Sales and Upper Bound Constraints

We further present the approach to maximize Sharpe measure with short sales and upper bound constraints. De Roon et al. (2001) and Li et al. (2003) suggest that the short-sale constraints to be considered in the portfolios of developing countries. However the constraints should not be ignored in the portfolio analysis of developed economies because the change in the regulation, such as during the 2008 financial crisis in Australia and the U.S., may restrict investors to short-sell stocks. We further consider the issue of concentration in a small amount of assets, which leads to illiquidity and triggers significant changes in volatility and correlations of assets. We take into account issues in portfolio weights to enhance feasibility of strategies. Green and Hollifield (1992) question the usefulness of corner solutions for non-constraint diversification. Jagannathan and Ma (2003) evaluate portfolio bounds from the angle of risk control and document that weight constraints decrease volatility in portfolio return and increase the feasibility of asset allocation. The optimal portfolio can be more practical by setting the upper bounds of weights not larger than a certain percentage (U).

$$0 \leq W_i \leq U, \quad \text{for } i = 1, 2, \ldots, n \tag{26.8}$$

The weight solution of unconstrained optimization for the weights $\mathbf{W} = [W_1, W_2, \ldots, W_n]$ in (26.7) is defined as S. limited as a subset (P) of the weights W_i in. The portfolio weights with short sales and upper bounds that the weight cannot be higher than U is P ($SS + UB(U)$), a subset of S, can be described as

$$P_{SS+DV(U)} = \left\{ \mathbf{w}_\mathrm{p} \in S : \cdot 0 \leq W_i \leq U, \cdot i = 1, 2, \ldots, N \right\} \tag{26.9}$$

where U can be any real number not greater than 1. The optimal weights can be solved by applying the Karush–Kuhn–Tucker conditions when the complementary slackness conditions, primal constraints, and gradient equations are fulfilled. Since there are multiple nonlinearities in the quadric programming, there is no equational form for the solutions. One can generate the Maximum Sharpe Ratio (MSR) portfolio among the selections on the constrained efficient frontier. Specifically,

$$MSR = \max_{\{\mathbf{W_P}\}} \left\{ \left(\mathbf{W}_\mathrm{p}^\mathrm{T} \boldsymbol{\mu} \right) / \left(\mathbf{W}_\mathrm{p}^\mathrm{T} \mathbf{V} \mathbf{W}_\mathrm{p} \right)^{\frac{1}{2}} \middle| \mathbf{W}_\mathrm{p}^\mathrm{T} \in P_J \right\} \tag{26.10}$$

We use the monthly return data of the component stocks in the Dow-Jones 30 Index between July 2008 and June 2018 to demonstrate an example. Our analysis prohibits short sales of asset and the asset allocation higher than a certain weight to ensure diversity of the portfolio. Since the large optimal weight among the assets when only short sales is about 42%, we set analyze the effect adding the upper bounds of portfolio weights by setting it as 20% and 10%.

Figure 26.3 exhibits the efficient frontiers that are constructed with various sets of weighting constraints. Among the optimal portfolio sets, the one that only prohibits short sales demonstrates higher mean–variance efficiency than the other two that include the upper bounds. The two curves with upper bounds are close to the no-short sales efficient frontier in low risk–return regime and are feasible. Consistent with risk reduction effect suggested by Jagannathan and Ma (2003), the constraints that the weights of assets cannot be greater than 20% do not significantly change the low-volatility portfolio selections but eliminate risky investing opportunities. When the diversity increase as the weights cannot be higher than 10%, the mean–variance efficiency drops and the investing area also shirks.

We report two characteristic portfolios on the efficient frontier. The first is the maximum Sharpe Ratio (MSR) portfolio which represents the highest mean–variance efficiency that can be achieved by the diversification. The second measure is the minimum-variance portfolio (MVP) as Elton et al. (2007) suggest that investors may seek to minimize the variance of a portfolio due to the lack of predictability of expected returns. Other than performance, we also analyze the effect of portfolio constraints on the asset restructure in the optimal portfolios. The Herfindahl–Hirschman Index (HHI) of the portfolio weights and the maximum weight among assets are used to measure the portfolio diversity. To evaluate how the candidate assets are utilized, the percentage that assets are included, the number of assets in the portfolio divided by that of all available assets is presented.

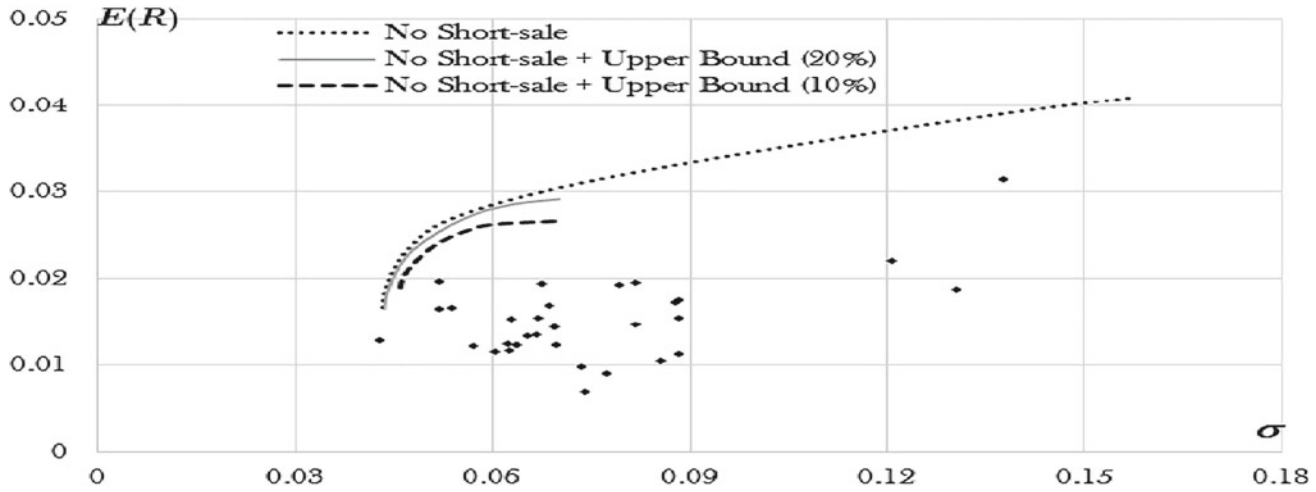


Fig. 26.3 Efficient frontier for the stocks in the Dow-Jones 30 Index

Table 26.1 Features of the maximum Sharpe ratio and minimum-variance portfolios This table reports expected return (E(r)), standard deviation (σ), Shape ratio, the Herfindahl–Hirschman Index (*HHI*) of the portfolio weights, the maximum of weight in the optimal portfolio (Max *W*), and the percentage of assets included in the optimal portfolio (% Asset) for the maximum Sharpe Ratio (MSR) portfolio and the minimum-variance portfolio (MVP)

	E(r)	σ	SR	*HHI* of *W*	Max *W*	% Asset
MSR portfolio						
No short-sale	0.0258	0.0510	0.51	0.2967	0.4282	10.00
No short-sale + upper bound (20%)	0.0248	0.0506	0.49	0.1487	0.2000	20.00
No short-sale + upper Bound (10%)	0.0239	0.0515	0.46	0.0923	0.1000	23.33
MVP						
No short-sale	0.0167	0.0434	0.39	0.2042	0.2763	16.67
No short-sale + upper bound (20%)	0.0165	0.0435	0.38	0.1731	0.2000	16.67
No short-sale + upper bound (10%)	0.0189	0.0461	0.41	0.0870	0.1000	30.00

The results are summarized in Table 26.1. For the MSR portfolios, the diversification with short selling constraint yields a maximum Sharpe ratio of 0.51. When the investment constraints are added and become more restrictive, the MSR gradually diminishes. For the most restrictive case, i.e., SS + UB(10%), the mean is 0.0239 with a standard deviation of 0.0515. The fact that the HHI of portfolio weights decreases from 0.297 to 0.092 as the upper bounds are increasingly restrictive suggests the portfolio is more diverse. As the upper limits are imposed, the maximal weight in the portfolio decreases but more available assets are included in the optimal selection.

For the minimum-variance portfolios, the mean–variance efficiency slightly improves (from 0.39 to 0.41) as the investment constraints are more restrictive, though the volatility of the MVPs increase (from 0.0434 to 0.0461). According to the HHI, maximum of portfolio weights, and the percentage of assets included in the optimal asset allocation, the portfolios are increasingly diverse when the upper bounds are included.

26.4 Treynor-Measure Approach with Short Sales Allowed

Using the single-index market model discussed in the previous chapter, Elton et al. (1976) define

$$\sigma_p = \left(\sum_{i=1}^{n} W_i^2 \beta_i^2 \sigma_m^2 + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} W_i W_j \beta_i \beta_j \sigma_m^2 + \sum_{i=1}^{n} W_i^2 \sigma_{\in i}^2 \right)^{1/2}.$$

Substituting of this value of σ_p into Eq. (26.3):

$$L = \frac{\sum_{i=1}^{n} W_i (\overline{R}_i - R_f)}{\sigma_p = \left(\sum_{i=1}^{n} W_i^2 \beta_i^2 \sigma_m^2 + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} W_i W_j \beta_i \beta_j \sigma_m^2 + \sum_{i=1}^{n} W_i^2 \sigma_{\in i}^2 \right)^{1/2}}. \tag{26.11}$$

In order to find the set of $W_i s$that maximizes L, take the partial derivative of the above equation with respect to each W_i and some manipulation. Then we can obtain

$$W_i = \frac{H_i}{\sum_{i=1}^{n} H_i}, \tag{26.12}$$

where

$$H_i = \frac{(\overline{R}_i - R_f)}{\sigma_{\in i}^2} - \left(\frac{\sigma_m^2 \sum_{j=1}^{n} \frac{(\overline{R}_j - R_f)}{\sigma_{\in i}^2}}{1 + \sigma_m^2 \sum_{j=1}^{i} \frac{\beta_j^2}{\sigma_{ej}^2}} \right) \left(\frac{\beta_i^2}{\sigma_{\in i}^2} \right). \tag{26.13}$$

The procedure of deriving Eqs. (26.12) and (26.13) can be found in Appendix 26B. The $H_i s$ must be calculated for all the stocks in the portfolio. If H_i is a positive value, this indicates the stock will be held long, whereas a negative value indicates that the stock should be sold short. This method is called the Treynor-measure approach. The argument will be clearer when the case of portfolio selection with short sales not allowed is discussed.

To determine the optimum portfolio from the $H_i s$ (such that 100% of funds are invested), the weights must be scaled. One method follows the standard definition of short sales, which presumes that a short sale of stock is a source of funds to the investor; it is called the **standard method of short sales**. This standard scaling method is indicated in Eq. (26.13). In Eq. (26.13), H_i can be positive or negative. This scaling factor includes a definition of short sales and the constraint:

$$\sum_{i=1}^{n} W_i = 1$$

A second method (**Lintner's** 1965 **method of short sales**) assumes that the proceeds of short sales are not available to the investor and that the investor must put up an amount of funds equal to the proceeds of the short sale. The additional amount of funds serves as collateral to protect against adverse price movements. Under these assumptions, the constraints on the $W_i s$ can be expressed as

$$\sum_{i=1}^{n} |W_i| = 1$$

And the scaling factor is expressed as

$$W_i = \frac{H_i}{\sum_{i=1}^{n} |H_i|} \tag{26.14}$$

Sample Problem 26.2 provides further illustration.

Table. 26.2 Data associated with regressions of the single-index model

(1) Security number	(2) $\overline{R}_i$	(3) $\overline{R}_i - R_f$	(4) β_i	(5) σ_{ei}^2
1	15	10	1	30
2	13	8	2	50
3	10	5	1.43	20
4	9	4	1.33	10
5	7	2	1	30

Sample Problem 26.2

The following example shows the differences in security weights in the optimal portfolio due to the differing short-sale assumptions. Data associated with regressions of the single-index model are presented in Table 26.2. The mean return, $\overline{R}$, the excess return $\overline{R}_i - R_f$, the beta coefficient β_i, and the variance of the error term $\sigma_{\in i}^2$ are presented in columns 2 through 5.

From the information in Table 26.2, using Eqs. (26.12) and (26.13). H_i $(i = 1, 2, \ldots, 5)$ can be calculated as

$$H_1 = \left(\frac{1}{30}\right)\left(\frac{15-5}{1}\right) - 3.067 = 0.2311$$
$$H_2 = \left(\frac{2}{50}\right)\left(\frac{13-5}{2}\right) - 3.067 = 0.0373$$
$$H_3 = \left(\frac{1.43}{20}\right)\left(\frac{10-5}{1.43}\right) - 3.067 = 0.0307$$
$$H_4 = \left(\frac{1.33}{10}\right)\left(\frac{9-5}{1.33}\right) - 3.067 = -0.0079$$
$$H_5 = \left(\frac{1}{30}\right)\left(\frac{7-5}{1}\right) - 3.067 = -0.0356$$

If this same example is scaled using the standard definition of short sales $\left(\sum_{i=1}^{5} H_i\right)$, which provides funds to the investor:

$$\sum_{i=1}^{5} H_i = 0.2556$$

$$W_1 = \frac{0.2311}{0.2556} = 0.9041$$
$$W_2 = \frac{0.0373}{0.2556} = 0.1459$$
$$W_3 = \frac{0.0307}{0.2556} = 0.1201$$
$$W_4 = \frac{-0.0079}{0.2556} = -0.0309$$
$$W_5 = \frac{-0.0356}{0.2556} = -0.1393$$

Using this definition of short sales provides that the investor should invest 90.41% of his or her money in security 1, and so on. If all $W_i s$ are added together, they equal 100%. This is true because the definition says that the funds received from selling short a security should be used to purchase more of the other securities.

According to Lintner's method:

$$\sum_{i=1}^{5} |H_i| = 0.3426$$

Now to scale the H_i values into an optimum portfolio, apply Eq. (26.14):

$$W_1 = \frac{0.2311}{0.3426} = 0.6745$$
$$W_2 = \frac{0.0373}{0.3426} = 0.1089$$
$$W_3 = \frac{0.0307}{0.3426} = 0.0896$$
$$W_4 = \frac{-0.0079}{0.3426} = -0.0231$$
$$W_5 = \frac{-0.0356}{0.3426} = -0.1039$$

Thus the Lintner model states that an investor should invest 67.45% in security 1, 10.89% in security 2, and 8.96% in security 3. The investor should then sell short 2.31% and 10.39% of securities 4 and 5, respectively.

The difference between Lintner's method and the standard method is due to the different definitions of short selling discussed earlier. The standard method assumes that the investor has the proceeds of the short sale, while Lintner's method assumes that the short seller does not receive the proceeds and must provide funds as collateral.

The method discussed in this section does not require the inputs of covariance among individual securities. Hence, it is a simpler method than that of the Sharpe performance measure method discussed in the previous section. The relative advantage of the Sharpe performance method over the Treynor performance method is exactly identical to the relative advantage of the Markowitz model over the Sharpe single-index model. In sum, the Treynor performance method for portfolio selection requires the information of both the risk-free rate and the market rates of return.

26.5 Treynor-Measure Approach with Short Sales not Allowed

Equation (26.13) can be modified to

$$H_i = \frac{\beta_i}{\sigma_{\in i}^2}\left(\frac{\overline{R}_i - R_f}{\beta_i} - C^*\right), \tag{26.15}$$

in which $\frac{(\overline{R}_i - R_f)}{\beta_i}$ is the **Treynor performance measure** and C^* can be defined as

$$C^* = \frac{\sigma_m^2 \sum_{j=1}^{i} \frac{(\overline{R}_i - R_f)\beta_i}{\sigma_{\in j}^2}}{1 + \sigma_m^2 \sum_{j=1}^{i} \frac{\beta_j^2}{\sigma_{\in j}^2}} \tag{26.16}$$

Elton et al. (1976) also derive a Treynor-measure approach with short sales not allowed. From Appendix 26C, Eq. (26.16) should be modified to

$$H_i = \frac{\beta_i}{\sigma_{\in i}^2}\left(\frac{\overline{R}_i - R_f}{\beta_i} - C^*\right) + \mu_i, \tag{26.17}$$

where $H_i \geq 0, \mu_i \geq 0,$ and $H_i \mu_i = 0,$

then

$$C^* = \frac{\sigma_m^2 \sum_{j=1}^{d} \frac{\overline{R}_j - R_f}{\sigma_{\in j}^2} \beta_j}{1 + \sigma_m^2 \sum_{j=1}^{d} \frac{\beta_j^2}{\sigma_{\in j}^2}}, \tag{26.18}$$

where d is a set that contains all stocks with positive H_i

If all securities have positive $\beta_i s$, the following three-step procedure from Elton et al. can be used to choose securities to be included in the optimum portfolio[3]

(1) Use the Treynor performance measure $\left(\overline{R}_i - R_f\right)/\beta_i$ to rank the securities in descending order.
(2) Use Eq. (26.18) to calculate C_i^* for first ith securities.
(3) Include i securities for which $\left(\overline{R}_i - R_f\right)/\beta_i$ are larger than C_i^*. Then C^* is equal to C_i^*.
(4) Use Eq. (26.5) to calculate optimum weights for i securities.

Sample Problem 26.3 provides a further illustration.

Sample Problem 26.3

The Center for Research in Security Prices tape was the source of 5 years of monthly return data, from January 2006 through December 2010, for the 30 stocks in the Dow-Jones Industrial Averages (DJIAs). The value-weighted average of the S&P 500 index was used as the market while 3-month Treasury-bill rates were used as the risk-free rate.

The single-index model was used with an ordinary least-squares regression procedure to determine each stock's beta. The following data were compiled for each stock.

(1) The mean monthly return $\overline{R}_i$;
(2) The mean excess return $\overline{R}_i - R_f$;
(3) The beta coefficient;
(4) The variance of the residual errors; and
(5) The Treynor performance measure $\left(\overline{R}_i - R_f\right)/\beta_i$.

All data are listed in the following worksheet, which lists the companies in descending order of Treynor performance measure.

To calculate the C_i^* as defined in Eq. (26.18), $\left(\overline{R}_j - R_f\right)\beta_j \Big/ \sigma_{\in j}^2$, $\sum_{j=1}^{i} \left[\left(\overline{R}_j - R_f\right)\beta_j \Big/ \sigma_{\in j}^2\right]$ and $\beta_j^2 \Big/ \sigma_{\in j}^2$. $\sum_{j=1}^{i} \left(\beta_j^2 \Big/ \sigma_{\in j}^2\right)$ are calculated and presented in the worksheet. Substituting $\sigma_m^2 = 0.00207$, $\sum_{j=1}^{i} \left[\left(\overline{R}_j - R_f\right)\beta_j \Big/ \sigma_{\in j}^2\right]$, and $\sum_{j=1}^{i} \left(\beta \sigma_j^2 \Big/ \sigma_{\in j}^2\right)$ into Eq. (26.18) produces C_i^* for every firm as listed in the last column in the worksheet.

[3] If the beta coefficient β_i for ith security is positive, then the size of H_i depends on the sign of the term in parentheses. Therefore, if a security with a particular $\left(\overline{R}_i - R_f\right)/\beta_i$ is included in the optimum portfolio, all securities with a positive beta that have higher values of $\left(\overline{R}_i - R_f\right)/\beta_i$ must be included in the optimum portfolio.

Worksheet for DJIAs

TICKER	Mean	Standard Deviation	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	$\overline{R_i}$	σ_i	$\overline{R_i}-R_f$	β_i	σ_m^2	$\frac{(\overline{R_i}-R_f)}{\beta_i}$	$\frac{(\overline{R_i}-R_f)\beta_i}{\sigma_e^2}$	Cumulative Sum of Column (5)	$\frac{\beta_i^2}{\sigma_e^2}$	Cumulative Sum of Column (7)	C_i^*
MCD	0.0173	0.0461	0.0155	0.4868	0.0015	0.0318	5.1718	5.17	162.78	162.78	0.0103
VZ	0.0095	0.0571	0.0077	0.6171	0.0022	0.0125	2.1822	7.35	174.64	337.42	0.0109
KO	0.0093	0.0510	0.0075	0.6107	0.0015	0.0122	2.9839	10.34	244.09	581.52	0.0112
WMT	0.0051	0.0473	0.0032	0.2939	0.0020	0.0110	0.4715	10.81	42.89	624.41	0.0112
IBM	0.0093	0.0603	0.0074	0.7494	0.0020	0.0099	2.7704	13.58	278.72	903.13	0.0109
MRK	0.0086	0.0778	0.0068	0.6950	0.0047	0.0097	0.9974	14.58	102.44	1005.57	0.0108
TRV	0.0072	0.0537	0.0054	0.5595	0.0020	0.0096	1.5043	16.08	156.46	1162.03	0.0107
CVX	0.0087	0.0624	0.0069	0.7161	0.0024	0.0096	2.0276	18.11	211.22	1373.25	0.0106
DIS	0.0109	0.0700	0.0091	1.0495	0.0017	0.0086	5.6156	23.72	650.72	2023.97	0.0100
UTX	0.0094	0.0616	0.0076	0.9454	0.0012	0.0080	6.0253	29.75	754.36	2778.32	0.0095
PG	0.0051	0.0489	0.0032	0.5031	0.0017	0.0064	0.9697	30.72	151.09	2929.41	0.0094
XOM	0.0048	0.0520	0.0030	0.4699	0.0021	0.0063	0.6699	31.39	105.85	3035.27	0.0093
T	0.0055	0.0616	0.0037	0.6707	0.0025	0.0055	0.9805	32.37	179.27	3214.53	0.0091
MSFT	0.0058	0.0802	0.0039	1.0130	0.0035	0.0039	1.1424	33.51	294.94	3509.47	0.0087
HPQ	0.0044	0.0752	0.0026	1.0496	0.0025	0.0024	1.0913	34.60	448.21	3957.68	0.0081
KFT	0.0030	0.0623	0.0012	0.6072	0.0028	0.0020	0.2581	34.86	129.35	4087.04	0.0079
CAT	0.0029	0.1279	0.0011	1.8954	0.0059	0.0006	0.3565	35.22	610.24	4697.27	0.0070
DD	0.0024	0.0872	0.0006	1.3410	0.0023	0.0004	0.3358	35.55	766.25	5463.52	0.0061
JNJ	0.0019	0.0443	0.0001	0.5758	0.0010	0.0002	0.0639	35.62	331.58	5795.10	0.0058
MMM	0.0019	0.0665	0.0000	0.7421	0.0029	0.0000	0.0071	35.63	192.91	5988.01	0.0056
PFE	0.0012	0.0638	-0.0006	0.6569	0.0028	-0.0009	-0.1368	35.49	151.62	6139.63	0.0055
CSCO	-0.0010	0.0892	-0.0029	1.2445	0.0035	-0.0023	-1.0299	34.46	447.41	6587.04	0.0050
JPM	-0.0016	0.0991	-0.0034	1.1347	0.0061	-0.0030	-0.6362	33.82	209.63	6796.67	0.0047
BA	-0.0036	0.0965	-0.0055	1.2556	0.0048	-0.0044	-1.4447	32.38	331.72	7128.39	0.0043
INTC	-0.0041	0.0855	-0.0059	1.0853	0.0039	-0.0055	-1.6467	30.73	300.84	7429.23	0.0040
HD	-0.0028	0.0768	-0.0046	0.7130	0.0045	-0.0065	-0.7319	30.00	113.42	7542.65	0.0038
AXP	-0.0094	0.1230	-0.0113	1.7421	0.0063	-0.0065	-3.1098	26.89	480.54	8023.19	0.0032
AA	-0.0211	0.1765	-0.0229	2.6867	0.0101	-0.0085	-6.1104	20.78	716.57	8739.77	0.0023
GE	-0.0135	0.1126	-0.0154	1.6787	0.0045	-0.0092	-5.7667	15.01	629.83	9369.59	0.0015
BAC	-0.0370	0.2225	-0.0388	2.6688	0.0290	-0.0145	-3.5646	11.45	245.28	9614.87	0.0011
R_m	-0.0013									Max C_i^*	0.0112
$Var(R_m)$	0.00294										
R_f	0.00183										

Using company VZ as an example:

$$C_{VZ}^* = \frac{(0.00294)(7.35)}{1+(0.00294)(337.42)} = 0.0109.$$

From C_i^* of the worksheet, it is clear that there are eight securities that should be included in the portfolio. The estimated $\beta_i/\sigma_{\in i}^2$, $(\overline{R}_i - R_f)/\beta_i$ and C_i^* of these eight securities are listed in Table 26.3. Substituting this information into Eq. (26.15) produces H_i for all eight securities. Using security MCD as an example:

$H_{MCD} = (334.4223)(0.0318 - 0.0112) = 6.8729.$

Using Eq. (26.15), the optimum weights can be estimated for all three securities, as indicated in Table 26.3. In other words, 90.01% of our fund should be invested in security MCD, 4.73% in security VZ, 5.26% in security KO. Based on the optimal weights, the average rate for the portfolio $\overline{R}_p$ is calculated as 1.65%, as presented in the last column of Table 26.3.

Table. 26.3 Positive optimum weight for three securities

Ticker	H_i	$\frac{\beta_i}{\sigma_{ei}^2}$	$\frac{(\overline{R_i}-R_f)}{\beta_i}$	(A) Optimum percentage	(B) Mean K	(A)X(B)
MCD	6.8729	334.4223	0.0318	0.9001	0.0173	0.0156
VZ	0.3609	282.9821	0.0125	0.0473	0.0095	0.0005
KO	0.4016	399.6946	0.0122	0.0526	0.0093	0.0005
Total	7.635466			*1*	$\overline{R_p}$	0.0165
*c**	0.0112					

26.6 Impact of Short Sales on Optimal-Weight Determination

The Markowitz model of portfolio analysis requires a large number of inputs as it is necessary to estimate the covariance between each pair of securities. Previously, the analysis was simplified by the assumption that security returns were related through a common response to some market index. This model, known as Sharpe's single-index model greatly reduces the number of inputs necessary to analyze the risk and return characteristics of portfolios. In both the Markowitz and Sharpe models, the analysis is facilitated by the presence of short selling.

This chapter discusses a method proposed by Elton and Gruber for the selection of optimal portfolios. Their method involves ranking securities based on their excess return to beta ratio, and choosing all securities with a ratio greater than some particular cutoff level C^*. It is interesting to note that while the presence of short selling facilitated the selection of the optimum portfolio in both the Markowitz and Sharpe models, it complicates the analysis when we use the Elton and Gruber approach. From Example 26.3, using the DJIA, absence of short selling allowed the formation of the optimal portfolio using only eight stocks.

26.7 Economic Rationale of the Treynor Performance-Measure Method

Cheung and Kwan (1988) have derived an alternative simple rate of optimal portfolio selection in terms of the single-index model. First, Cheung and Kwan relate C^* as defined in Eq. (26.13) to the correlation coefficient between the portfolio rates of return with i securities R_i and market rates of return $R_m(\rho_i)$:

$$\rho_i = \frac{C^*}{\sigma_m \Theta_i}, \tag{26.19}$$

where

$$\begin{aligned}
\rho_i &= \sigma_{im}/\sigma_i\sigma_m;\\
\sigma_{im} &= \text{covariance between } R_i \text{ and } R_m;\\
\Theta_i &= \left(\overline{R}_i - R_f\right)/\sigma_i, \text{ the Sharpe performance measure associated}\\
&\quad \text{with the } i\text{th portfolio; and}\\
\sigma_i \text{ and } \sigma_m &= \text{standard deviation for ith portfolio and market portfolio,}\\
&\quad \text{respectively.}
\end{aligned}$$

Cheung and Kwan show that ρ_i and C_i^* display the same functional behavior for optimal portfolio selection. In other words, if portfolios are formed by adding securities successively from the highest rank to the lowest rank, the optimal portfolio is reached when the correlation of the portfolio returns and the index is at its maximum. Since the expected return on the index must be positive if the investor is to invest in stocks, an objective of the investor's using the single-index model to establish the risk–return tradeoff is to pick securities that benefit the most from a market upswing. Hence, the role of index in the selection of securities for an optimum portfolio is demonstrated explicitly.

Based upon the single-index model and the risk decomposition discussed in previous chapters, the following relationships can be defined:

$$\begin{aligned}
&1.\ \rho_i = \frac{\sigma_{im}}{\sigma_i\sigma_m}\\
&2.\ \sigma_{im} = \beta_i\sigma_m^2\\
&3.\ \sigma_i^2 = \beta_i^2\sigma_m^2 + \sigma_{\in i}^2
\end{aligned} \tag{26.20}$$

From Eq. (26.20), Cheung and Kwan define ρ_i in terms of β_i^2, σ_m^2, and σ_i.

$$\rho_i = \sqrt{\beta_i^2\sigma_m^2/\sigma_i^2} + \sigma_{\in i}^2 = \sqrt{1 - \frac{\sigma_{\in i}^2}{\sigma_i^2}}$$

in which $\sigma^2_{\in i}$ is the non-systematic risk for the ith portfolio. They use both ρ_i and Θ_i to select securities for an optimum portfolio, and they conclude that ρ_i can be used to replace Θ_i in selecting securities for an optimum portfolio. Nevertheless, Θ_i information is still needed to calculate the weights for each security. Hence, Cheung and Kwan's ρ_i criteria are good only for understanding Elton and Gruber's performance measure method for portfolio selection.

26.8 Summary

Following Elton et al. (1976) and Elton and Gruber (1987), we have discussed the performance measure approaches to selecting optimal portfolios. We have shown that the performance measure approaches for optimal portfolio selection are complementary to the Markowitz full variance–covariance method and the Sharpe index model method. These performance measure approaches are thus worthwhile for students of finance to study following an investigation of the Markowitz variance–covariance method and Sharpe's index approach.

Appendix 26A: Derivation of Eq. (26.6a)

The objective function L as defined in Eq. (26.3) can be rewritten as.

$$\text{Max } L = \left[\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)\right]\left[\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j \sigma_{ij}\right]^{-1/2} \qquad i \neq j$$

Then, following the product and chain rule, we have

$$\begin{aligned}
\frac{dL}{dW_i} &= \frac{d}{dW_i}\left[\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)\right]\left[\sum_{i=1}^{n} W_i\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_j\sigma_{ij}\right]^{-1/2} \quad i \neq j \\
&= \left[\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j\sigma_{ij}\right]^{-1/2} \times \frac{d}{dW_i}\left[\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)\right] \\
&\quad + \left[\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)\right]\frac{d}{dW_i}\left[\sum_{i=1}^{n} W_i\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_j\sigma_{ij}\right]^{-1/2} \\
&= \left[\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j\sigma_{ij}\right]^{-1/2}\left[\overline{R}_i - R_f\right] \\
&\quad - \frac{1}{2}\left[\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)\right]\left[\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j\sigma_{ij}\right]^{-3/2} \\
&\quad \times \left[2W_i\sigma_i^2 + 2\sum_{j=1}^{n} W_j\sigma_{ij}\right] \\
&= 0 \qquad (i = 1, 2, \ldots, n)
\end{aligned} \tag{26A.1}$$

Multiplying Eq. (26A.1) by $\left[\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j\sigma_{ij}\right]^{1/2}$, $i \neq j$, and rearranging yields:

$$\overline{R}_i - R_f = \left(\frac{\sum_{i=1}^{n} W_i\left(\overline{R}_i - R_f\right)}{\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{j=1}^{n}\sum_{j=1}^{n} W_i W_j\sigma_{ij}} \quad i \neq j\right)\left(W_i\sigma_i^2 + \sum_{j=1}^{n} W_j\sigma_{ij}\right). \tag{26A.2}$$

Defining

$$k = \frac{\sum_{i=1}^{n} W_i(\overline{R}_i - R_f)}{\sum_{i=1}^{n} W_i^2\sigma_i^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} W_i W_j \sigma_{ij}} \quad i \neq j$$

yields

$$-k\left(W_i\sigma_i^2 + \sum_{i=1}^{n} W_i\sigma_{ij}\right) + (\overline{R}_i - R_f) = 0 \quad j \neq i$$

Therefore,

$$\begin{aligned}\frac{dL}{dW_i} = &-(kW_1\sigma_{1i} + kW_2\sigma_{2i} + \cdots + kW_i\sigma_i^2 + \cdots \\ &+ kW_n\sigma_{ni}) + (\overline{R}_i - R_f) \\ &= 0\end{aligned}$$

Define $H_i = kW_i$, where W_i are the fractions to invest in each security and H_i are proportional to this fraction. Substituting H_i for kW_i:

$$\overline{R}_i - R_f = H_i\sigma_{1i} + H_2\sigma_{2i} + \cdots + H_i\sigma_i^2 + \cdots + H_n\sigma_{ni}$$

There is one equation like this for each value of i.

$$\begin{aligned}\overline{R}_1 - R_f &= H_1\sigma_1^2 + H_2\sigma_{12} + \cdots + H_n\sigma_{1n} \\ \overline{R}_2 - R_f &= H_1\sigma_{21} + H_2\sigma_2^2 + \cdots + H_n\sigma_{2n} \\ &\vdots \\ \overline{R}_n - R_f &= H_1\sigma_{n1} + H_2\sigma_{n2} + \cdots + H_n\sigma_n^2\end{aligned}$$

This is Eq. (26.6a) in the next.

Appendix 26B: Derivation of Eq. (26.10)

Following the optimization procedure for deriving Eq. (26A.2) in Appendix 26A:

$$\begin{aligned}\frac{dL}{dW_i} = &(\overline{R}_i - R_f) - \frac{\sum_{i=1}^{n} W_i(\overline{R}_i - R_f)}{\sigma_p^2}\left(W_i\beta_i^2\sigma_m^2 + \beta_i\sum_{j=1}^{n} W_j\beta_j\sigma_m^2 + W_i\sigma_{\in i}^2\right) \\ &= 0 \; j \neq i\end{aligned}$$

Let $H_i = \left[(\overline{R}_p - R_f)\Big/\sigma_p^2\right]W_i$ and, solving for any H_i,

$$H_i = \frac{\overline{R}_i - R_f}{\sigma_{\in i}^2} - \frac{\beta_i\sigma_m^2\sum_{j=1}^{n} H_j\beta_j}{\sigma_{\in i}^2}. \tag{26B.1}$$

Multiplying both sides of the equation by β_i:

$$H_i\beta_i = \frac{(\overline{R}_i - R_f)\beta_i}{\sigma_{\in i}^2} - \frac{\beta_i\sigma_m^2\sum_{j=1}^{n} H_j\beta_j^2}{\sigma_{\in i}^2}. \tag{26B.2}$$

Adding together the n equation of this form yields

$$\sum_{j=1}^{n} H_j\beta_j = \frac{\sum_{j=1}^{n} \frac{(\overline{R}_j - R_f)\beta_j}{\sigma^2_{\in i}}}{1 + \sigma^2_{\in i} \sum_{j=1}^{n} \frac{\beta_j^2}{\sigma^2_{\in i}}}. \tag{26B.3}$$

By substituting Eq. (26B.3) into Eq. (26B.1), we can obtain (26.10).

$$H_i = \frac{(\overline{R}_i - R_f)}{\sigma^2_{\in i}} - \left(\frac{\sigma_m^2 \sum_{j=1}^{n} \frac{(\overline{R}_j - R_f)}{\sigma^2_{\in i}}}{1 + \sigma_m^2 \sum_{j=1}^{i} \frac{\beta_j^2}{\sigma^2_{\in j}}} \right) \left(\frac{\beta_i^2}{\sigma^2_{\in i}} \right)$$

Appendix 26C: Derivation of Eq. (26.15)

This appendix discusses the use of performance measure to examine the optimal portfolio with short sales not allowed. Therefore, Eq. (26B.1) from Appendix 26.B must be modified:

$$H_i = \frac{\overline{R}_i - R_f}{\sigma^2_{\in i}} - \frac{\beta_i \sigma_m^2}{\sigma^2_{\in i}} \sum_{j=1}^{n} \beta_j H_j + \mu_i, \tag{26C.1}$$

where $H_i \geq 0, \mu_i \geq 0$, and $\mu_i H_i = 0$ for all i.

The justification of this equation can be found in Elton et al. (1976). Assuming all stocks that would be in an optimal portfolio (called d) can be found and then arranging these stocks as $i = 1, 2, \ldots, d$, for the subpopulation of stocks that make up the optimal portfolio:

$$H_i = \frac{\overline{R}_i - R_f}{\sigma^2_{ei}} - \frac{\beta_i \sigma_m^2}{\sigma^2_{ei}} \sum_{j=1}^{d} H_j\beta_j \text{ and } \mu_i = 0 \tag{26C.2}$$

Multiplying both sides by β_j, summing over all stocks in d, and rearranging yields

$$\sum_{j=1}^{d} H_j\beta_j = \frac{\sum_{j=1}^{d} \left(\frac{\overline{R}_j - R_f}{\sigma^2_{\in j}} \right) \beta_j}{1 + \sigma_m^2 \sum_{j=1}^{d} \frac{\beta_j}{\sigma^2_{\in j}}}. \tag{26C.3}$$

Notice since the set d contains all stocks with positive H_i:

$$\sum_{j=1}^{n} H_j\beta_j = \sum_{j=1}^{d} H_j\beta_j,$$

and let:

$$C^* = \sigma_m^2 \frac{\sum_{j=1}^{d} \frac{\overline{R}_j - R_f}{\sigma^2_{\in j}} \beta_j}{1 + \sigma_m^2 \sum_{j=1}^{n} \frac{\beta_j^2}{\sigma^2_{\in j}}}. \tag{26C.4}$$

Using (26C.4), the following equation for H_i is obtained after substitution and rearranging from Eq. (26C.1):

$$H_i = \frac{\beta_i}{\sigma^2_{\in i}} \left(\frac{\overline{R}_i - R_f}{\beta_i} - C^* \right) + \mu_i \tag{26C.5}$$

Since $\mu_i \geq 0$, the inclusion of μ_i can only increase the value of H_i. Therefore, if H_i is positive, $\mu_i = 0$ can never make it zero and the security should be included If $H_i < 0$ when $\mu_i = 0$, positive values of μ_i can increase H_i. However, because the product of μ_i and H_i must equal zero, as indicated in Eq. (26C.1), positive values of μ_i imply $H_i = 0$. Therefore, any security $H_i < 0$ when $\mu_i = 0$ must be rejected. Therefore, Eq. (26.15) in the text can be used to estimate the optimal weight of a portfolio.

Appendix 26D: Quadratic Programming and Kuhn–Tucker Conditions

The quadratic programming algorithm is based on a technique from advanced calculus called Kuhn–Tucker conditions. This technique can solve the optimal portfolio with short sales not allowed. The problem can be stated as

$$\text{Max}_{\{W_i, i=1,2,\ldots n\}} F(W) = \frac{\overline{R_p} - R_f}{\sigma_p}$$

$$\text{Subject to}$$

$$\sum_{i=1}^{n} W_i = 1$$

$$W_i \geq 0, i = 1, 2, \ldots n$$

where

$$\overline{R_p} = \sum_{i=1}^{n} W_i \overline{R}_i$$

$$\sigma_p = \left[\sum_{i=1}^{n} W_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j \sigma_{ij} \right]^{1/2}$$

To find the maximum value of $F(W)$, there are two cases of the optimal weights $W = (W_1, W_2, \ldots W_n)$ in Fig. 26.4

If $F(W)$ is a function of W_i in Fig. 26.4a, then the maximum value of $F(W)$ occurs at P, where $\frac{dF(W)}{dW_i} = 0$ at positive optimal weight ($W_i > 0$). However, in Fig. 26.4b, the maximum feasible value of $F(W)$ occurs at P' instead of P because of short sell not allowed, then $\frac{dF(W)}{dW_i} < 0$ at optimal weight ($W_i = 0$). Therefore, in general, we can obtain the optimal weights under conditions

$$\frac{dF(W)}{dW_i} \leq 0$$

$$\frac{dF(W)}{dW_i} W_i = 0$$

We can rewrite the conditions as five Kuhn–Tucker conditions:

$$\frac{dF(W)}{dW_i} + A_i = 0$$

$$W_i A_i = 0$$

$$W_i \geq 0$$

$$A_i \geq 0$$

$$\sum_{i=1}^{n} W_i = 1$$

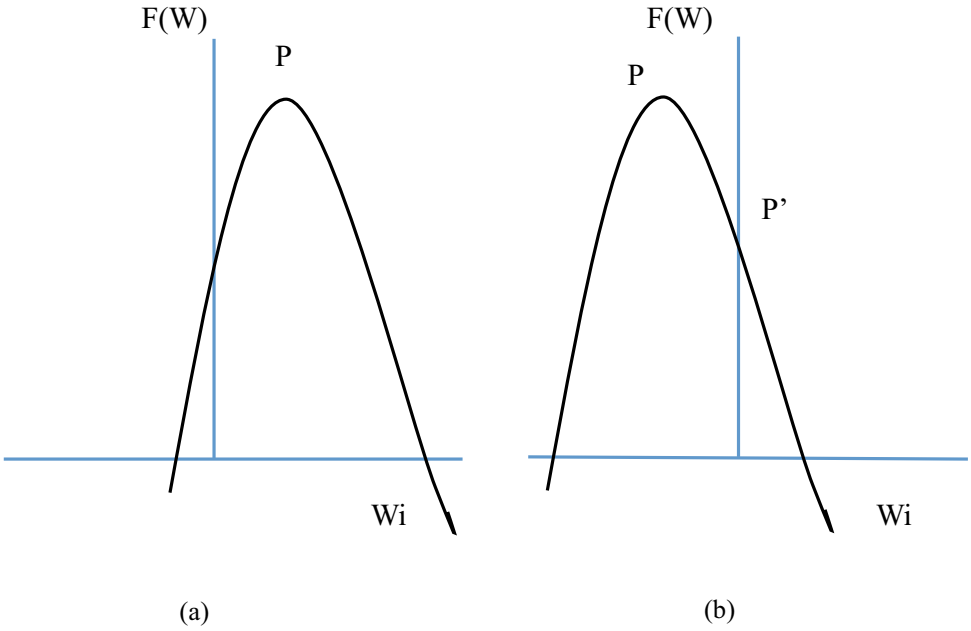


Fig. 26.4 Value of the function $F(W)$ as W_i changes

When maximum feasible value occurs on $\frac{dF(W)}{dW_i} < 0$, A_i is positive and W_i is equal to zero. If the maximum feasible value occurs on $\frac{dF(W)}{dW_i} = 0$, then A_i is equal to zero and W_i is positive. In the following part, we will solve the optimal weights under Kuhn–Tucker conditions in Excel.

Given an example in Fig. 26.5: initial weights W = (0.1, 0.2, 0.7), $\frac{dF(W)}{dW_i}$ can be calculated by Eq. (26A.2) in Appendix 26A and the value of the excess return and covariance matrix as below.

Then we use "***Solver***" function in Excel to find the optimal weights W. Set target cell B5 equal to the value of 1 (the fifth Kuhn–Tucker condition), select the range from B2 to C4 as change cells, and select the first and second Kuhn–Tucker conditions as the constraints.

For the third and fourth Kuhn–Tucker conditions, press "***Options***" into **Solver Options**, select "***Assume Non-Negative***" and press "***OK***." Then go back to **Solver Parameters** and press "***Solve***."

If first solve cannot find a solution, use "***Solver***" function again until the Solver find a solution as Fig. 26.6.

Appendix 26E: Portfolio Optimization with Short Selling Constraints[4]

The traditional mean–variance efficient portfolios generate appealing characteristics. However, it is frequently questioned by financial professionals due to the propensity of corner solutions in portfolio weights. Therefore, adding portfolio weighting limits help portfolio managers to fashion realistic asset allocation strategies.

Markowitz (1952) model is based on the following assumptions concerning the behavior of investors and financial markets:

(1) Investors have single-period utility functions in which they maximize utility within the framework of diminishing marginal utility of wealth. Return is desired but the risk is to be avoided.
(2) Investors care only about the first two moments of the returns of their portfolios over the investment period.
(3) Investors can long- and short-sell assets without limits.
(4) Financial markets are frictionless, that is, there are no taxes and costs.

[4] This appendix is contributed by Paul Chiou at Shipensburg University.

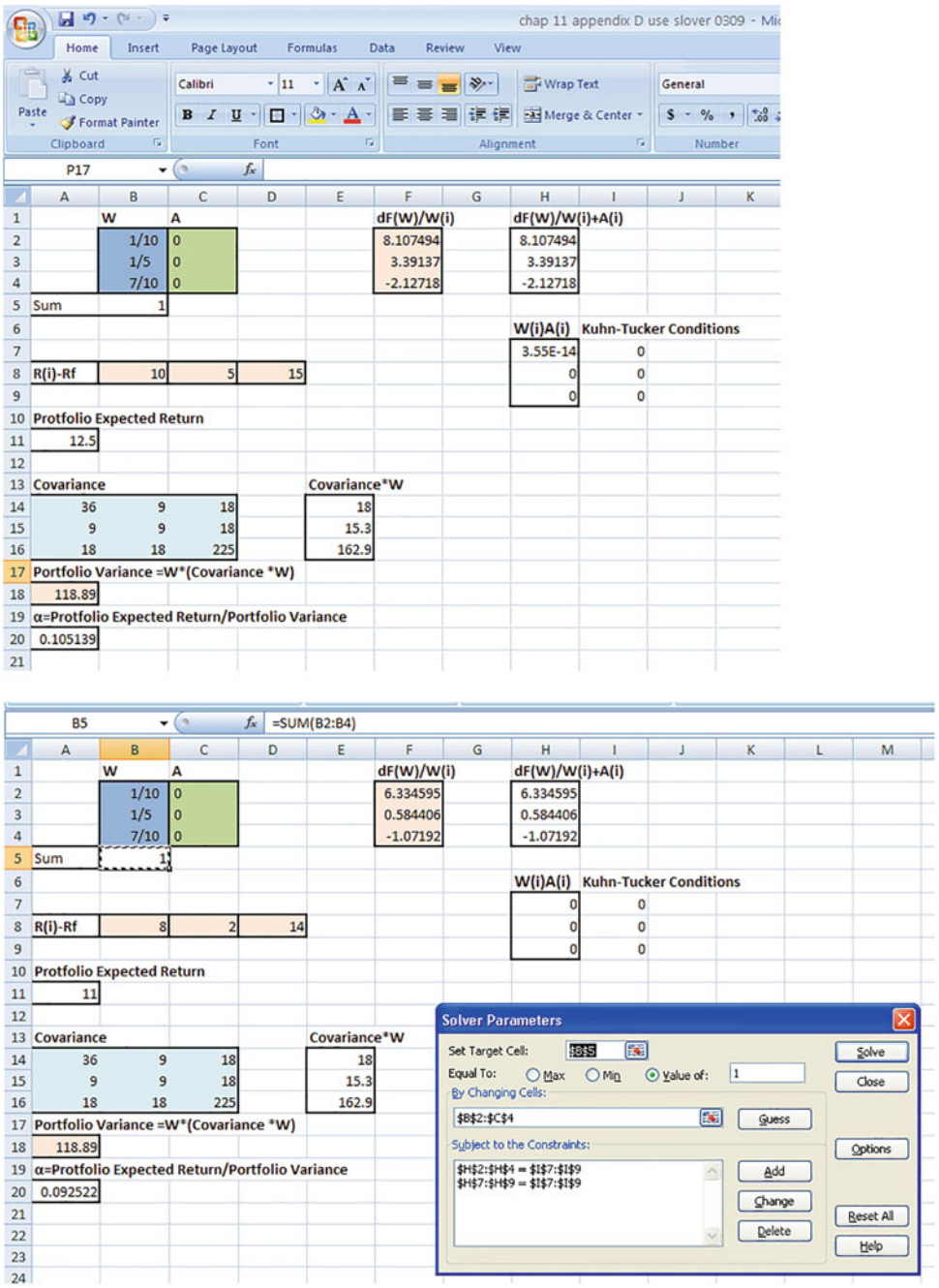


Fig. 26.5 Solver function in excel

Suppose risky asset investments can be characterized as a vector of multivariate returns of N securities, $\mathbf{R}^{\mathrm{T}}$. The expected risk premiums and variance–covariance of asset returns can be expressed as a vector $\boldsymbol{\mu}$ and a positive definite matrix $\mathbf{V}$, respectively. Let Ω be the set of all real vectors $\mathbf{w}$ that defines the weights of assets such that $\mathbf{w}^{\mathrm{T}}\mathbf{1} = 1$, where $\mathbf{1}$ is an N-vector of ones. The expected return of the portfolio is $\mu_{\mathrm{p}} = \mathbf{w}^{\mathrm{T}}\boldsymbol{\mu}$ and the variance of the portfolio is $\sigma_{\mathrm{p}} = \mathbf{w}^{\mathrm{T}}\mathbf{V}\mathbf{w}$. Considering all constraints given the objective to minimize the portfolio's risk, the efficient frontier can be then expressed as a Lagrangian function:

$$\min_{\{\mathbf{w},\phi,\eta\}} \Xi = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{V}\mathbf{w} + \phi\left(\mu_p - \mathbf{w}^{\mathrm{T}}\boldsymbol{\mu}\right) + \eta\left(1 - \mathbf{w}^{\mathrm{T}}\mathbf{1}\right) \tag{26E.1}$$

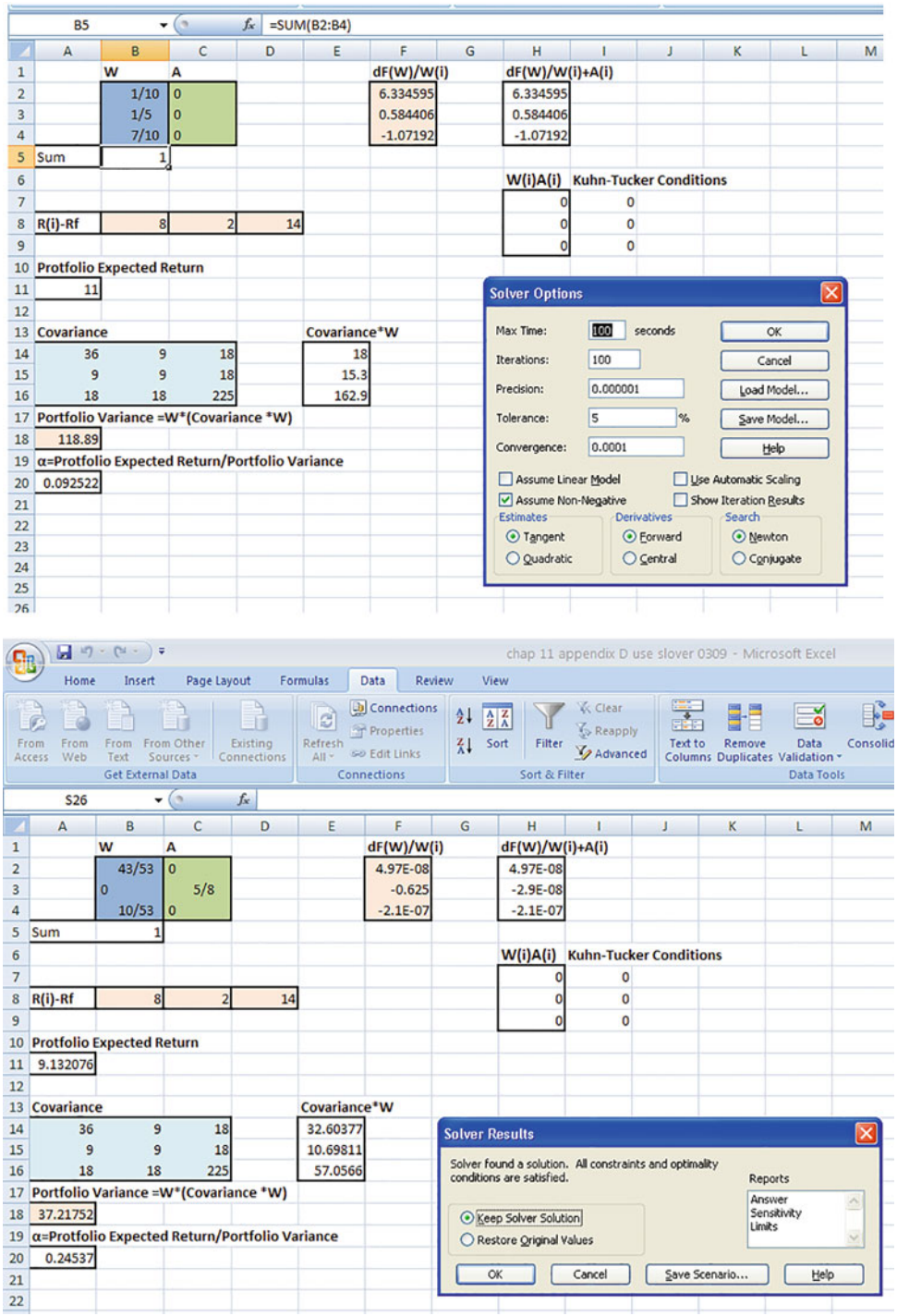


Fig. 26.6 Optimal weights by solver function

As described in the previous section, the optimal portfolio weights are a function of means, variances, and covariances of asset returns. Consider a generalized case with N assets, let

$$\mathbf{A}=\mathbf{1}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{R},\mathbf{B}=\mathbf{R}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{R},\mathbf{X}=\mathbf{1}^{\mathrm{T}}\mathbf{V}^{-1},\ \text{and}\ \ \mathbf{\Delta}=\mathbf{BX}-\mathbf{A}^2,$$

the solution of the above quadratic function is (see Pennacchi 2008)

$$\mathbf{w}_{\mathrm{p}}=\frac{1}{\mathbf{\Delta}}\left[\left(\mathbf{X}\mu_{\mathrm{p}}-\mathbf{A}\right)\left(\mathbf{V}^{-1}\mathbf{R}\right)+\left(\mathbf{B}-\mathbf{A}\mu_{\mathrm{p}}\right)\left(\mathbf{V}^{-1}\right)\right] \tag{26E.2}$$

The short selling constraints are further considered in Markowitz model. The inequalities that represent non-negative portfolio weights are

$$\mathbf{w}_1\geq 0,\mathbf{w}_2\geq 0,\ldots\ldots,\mathbf{w}_N\geq 0,\ \text{where}\ \mathbf{w}=\left[\mathbf{w}_1,\mathbf{w}_2,\ldots\ldots,\mathbf{w}_{\mathrm{N}}\right] \tag{26E.3}$$

The solution of this constrained diversification is to incorporate Eq. (26.3) in Eq. (26.1). Although the constraints on portfolio weights ($\mathbf{w}^{\mathrm{T}}\mathbf{1}=1$ 1 and Eq. (26E.3)) are linear, the objective function, which is to minimize σ_p, is nonlinear as it contains the squares and cross-products of portfolio weights. There is no standard package to solve this quadratic programming issue because of the inequality restriction on each of elements in $\mathbf{w}$.[5]

Kuhn–Tucker conditions are frequently applied to solve the above problem. Consider the quadratic property in the objective function; the traditional Lagrangian techniques encounter difficulty to solve the global optimum. When inequalities are active, they will be treated as equalities and are included in the Lagrangian. Once the equalities are inactive, they should be left out. The advantage is that Kuhn–Tucker conditions require the objective function gradient to be expressible as a multiplier-weighted combination of the constrained gradient. The quadratic programming problem becomes complicated once there are more constraints included and can be solved by a computer program.[6] However, the results of such constrained optimization in asset allocation generate more feasible portfolios than unconstrained optimization.[7]

(1) A Numerical Example

Considering a three-asset case exemplified in the previous section, we have the following objective and constraint functions:

$$\begin{aligned}&\min\sigma_p^2=\sum_{j=1}^{3}\sum_{i=1}^{3}W_jW_i\mathrm{Cov}\left(R_j,R_i\right),\\&\text{s.t.}E(R_p)=\sum_{i=1}^{3}W_iE(R_i),\\&\sum_{i=1}^{3}W_i=1\\&W_1\geq 0,W_2\geq 0,W_3\geq 0.\end{aligned} \tag{26E.4}$$

Replacing the variables in Eq. (26E.4), we can rewrite Eq. (26E.1) as

$$\min\Xi=\left[W_1^2\sigma_{11}+W_2^2\sigma_{22}+W_3^2\sigma_{33}\right)+2(W_1W_2\sigma_{12}+W_1W_3\sigma_{13}+W_2W_3\sigma_{23})$$

[5] For detailed discussion, see Chaps. 6 and 9 in Elton et al. (2007).
[6] See Chap. 6 in Elton et al. (2007) and Chap. 14 in Rardin (1998).
[7] See Jagannathan and Ma (2003). Chiou et al. (2009) empirically investigate the impact of weighting constraints on international diversification and suggest that including upper and lower bounds in portfolio weights enhance feasibility of asset allocation.

$$
\begin{aligned}
&- \varphi(E(R_p) - \sum_{i=1}^{3} W_i E(R_i)) \\
&- \eta(\sum_{i=1}^{3} W_i - 1) \\
&- \lambda_1 W_1 - \lambda_2 W_2 - \lambda_3 W_3],
\end{aligned}
\tag{26E.5}
$$

where φ, η, and λ_i i = 1, 2, 3, are Lagrange multipliers (LMs). They represent the shadow prices, or the penalty, of the constraints. The corresponding complementary slackness conditions are

$$\varphi(E(R_p) - \sum_{i=1}^{3} W_i E(R_i)) = 0 \tag{26E.6}$$

$$\eta(\sum_{i=1}^{3} W_i - 1) = 0$$

$$-\lambda_1 W_1 = 0$$

$$-\lambda_2 W_2 = 0$$

$$-\lambda_3 W_3 = 0$$

The above complementary slackness conditions indicate that either inequality constraints should be active at a local optimum or the corresponding Lagrange variable should equal zero. For differentiable nonlinear program, solutions W_i, φ, φ, λ_i, i = 1, 2, 3, satisfy the Kuhn–Tucker conditions if they fulfill complementary slackness conditions, primal constraints, and gradient equation

$$\nabla \frac{\partial \sigma_p^2}{\partial W_i} = \sum \nabla \text{Constraint}(W_1, W_2, W_3) \cdot \text{LM} \tag{26E.7}$$

Any combination of W_i, i = 1, 2, 3, for which there exist a corresponding LMs satisfying these conditions is called a Kuhn–Tucker point.

Our portfolio model has the following objective function gradient

$$\nabla \frac{\partial \sigma_p^2}{\partial W_i} = \begin{pmatrix} 0.91W_1 + 0.0018W_2 + 0.0008W_3 \\ 0.0018W_1 + 0.1228W_2 + 0.002W_3 \\ 0.0008W_1 + 0.002W_2 + 0.105W_3 \end{pmatrix} \tag{26E.8}$$

and those of the five linear constraints are

$$\nabla \text{Constraint}_1(W_1, W_2, W_3) = (0.0053, 0.0055, 0.0126) \tag{26E.9}$$

$$\nabla \text{Constraint}_2(W_1, W_2, W_3) = (1, 1, 1)$$

$$\nabla \text{Constraint}_3(W_1, W_2, W_3) = (1, 0, 0)$$

$$\nabla \text{Constraint}_4(W_1, W_2, W_3) = (0, 1, 0)$$

$\nabla \text{Constraint}_5(W_1, W_2, W_3) = (0, 0, 1)$..

Therefore the gradient equation part of Kuhn–Tucker conditions is

$$0.0053\varphi + 1\eta + 1\lambda_1 = 0.91W_1 + 0.0018W_2 + 0.0008W_3 \tag{26E.10}$$

$$0.0055\varphi + 1\eta + 1\lambda_2 = 0.0018W_1 + 0.1228W_2 + 0.002W_3$$

$$0.0126\varphi + 1\eta + 1\lambda_3 = 0.0008W_1 + 0.002W_2 + 0.105W_3$$

plus the primal constraints as part of the conditions

$$E(R_p) = \sum_{i=1}^{3} W_i E(R_i) \tag{26E.10}$$

$$\sum_{i=1}^{3} W_i = 1$$

$$W_1 \geq 0$$

$$W_2 \geq 0$$

$$W_3 \geq 0$$

and complementary slackness conditions in Eq. (26E.6).

Notice that the weights are functions of five LMs and bounded by inequality constraints. Since the functions of solutions of W_i are multidimensional, we cannot show their relation on a graph. One may start from any feasible point and then search an improving feasible direction $\Delta(W_1, W_2, W_3)$ chased by implementations of feasible and small steps of LMs. The stop of an improving feasible search not necessarily represents the current Kuhn–Tucker point as the global optimum but suggests it is a local optimum. Since there is no close-form solution function for each weight, one may need to continue the search until no improvement can be found. In our case, if

$$\nabla \frac{\partial \sigma_p^2}{\partial W_i} \cdot \Delta(W_1, W_2, W_3) < 0, \tag{26E.12}$$

there is an improvement in objective function. The time-consuming calculation of the constrained portfolio selection can be speeded by applying computer software such as Microsoft Excel and MATLAB.[8]

We use the same data set to construct the nonconstrained (NC) efficient frontier and short selling constrained (SS) efficient frontier, which is shown in Fig. 26.7 When the portfolio weights have no limits, theoretically, the investor can expand the upper bound of efficient frontier unlimitedly by allocating assets in extreme long and short positions in different securities. The maximum return of NC efficient frontier can be infinite if the investor is willing to take the infinite risk. In Fig. 26.7, the SS optimal diversification is a subset of the NC portfolio.

The global minimum-variance (MV) for the two kinds of portfolio is identical in this case. The information is listed in Table 26.4 The conclusion of the same MV under different constraints does not always happen when the coefficients of correlation among securities and the relative magnitudes of return among assets change. Note that the optimal diversification strategies are sensitive to the variation in the first two moments of asset returns in the portfolio.

Notice that, theoretically, the maximum return of NC efficient frontier can be infinite if the investor is willing to take an infinite risk by allocating assets in extremely huge long and short positions. However, the effectiveness of such extreme-weight diversification strategies is questionable due to their low mean–variance efficiency. As the portfolios on efficient frontiers are shown in Table 26.5, assuming the long-term annual risk-free interest rate is 4%, the portfolios of corner solutions are less mean–variance efficient than the ones without negative and extremely positive weights.

[8] For example, the Matlab syntax frontcon allows users to enter a matrix which defines constraints related to minimum and maximum for assets. For more in-detail discussion, readers may refer the text of Matlab's finance function.

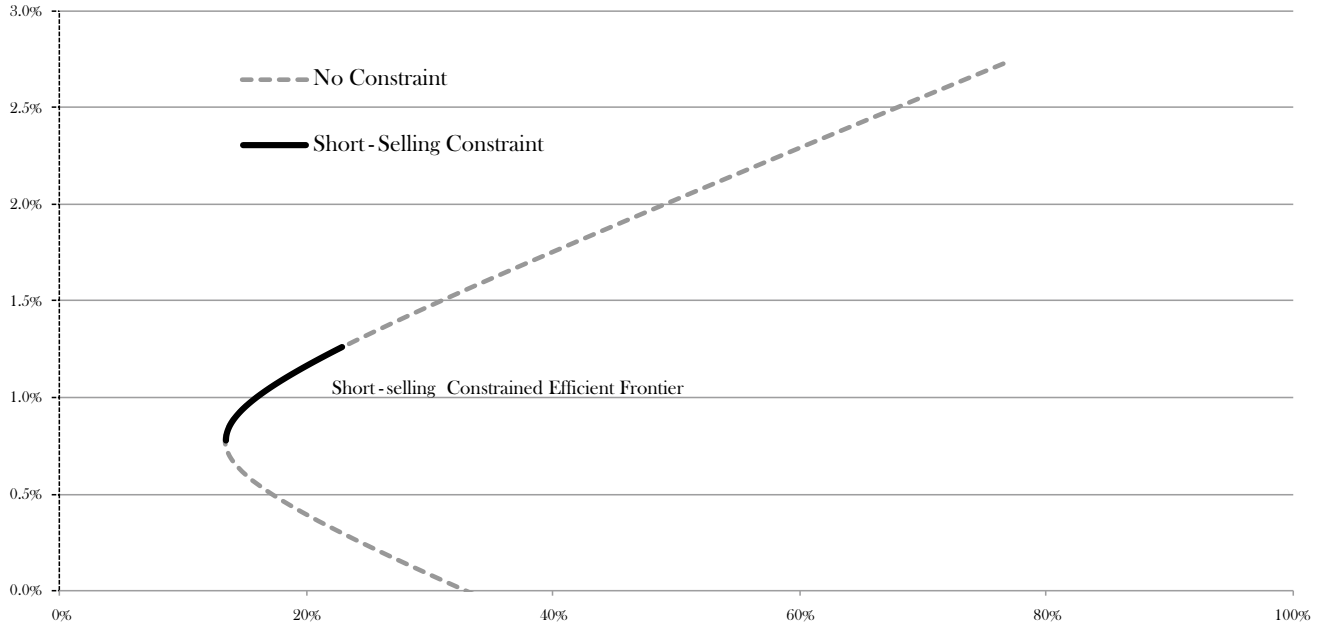


Fig. 26.7 Efficient frontiers

Table 26.4 Minimum-variance portfolio

E(R) (%)	σ (%)	Portfolio weight	w_1 (%)	w_2 (%)	w_3 (%)
0.78	13.39		38.57	28.10	33.33

Table 26.5 Portfolios on efficient frontier

E(R) (%)	σ(%)	Sharpe ratio	Portfolio weight	w_1 (%)	w_2 (%)	w_3 (%)
0.78	13.39	0.0333		38.57	28.10	33.33
1.04	16.87	0.0422		16.86	13.04	70.10
1.24	22.38	0.0406		−25.82	−16.90	142.72
2.63	72.96	0.0315		−100.53	−199.47	400.00
3.38	101.30	0.0300		−366.73	66.73	400.00

Bibliography

Alexander, G. J. and B. J. Resnick. "More on Estimation Risk and Single Rule for Optimal Portfolio Selection." *Journal of Finance*, v. 40 (March 1985), pp. 125–134.

Bodie, Z., A. Kane and A. Marcus. Investments, 7th ed. New York: McGraw-Hill Book Company,2006 .

Chen, S. N. and S. J. Brown. "Estimation Risk and Simple Rules for Optimal Portfolio Selection." *Journal of Finance*, v. 38 (September 1983), pp. 1087–1093.

Cheung, C. S. and C. C. Y. Kwan. "A Note on Simple Criteria for Optimal Portfolio Selection." *Journal of Finance*, v. 43 (March 1988), pp. 241–245.

Chiou, W. P., A. C. Lee and C. A. Chang. "Do Investors Still Benefit From International Diversification with Investment Constraints?" *Quarterly Review of Economics and Finance*, v. 49 (2009), pp. 449–483.

Chiou, W. P., A. C. Lee and Cheng F. Lee. "Stock Return, Risk and Legal Environment around the World." *International Review of Economics and Finance*, v. 19 (2010), pp. 95–105.

Elton, E. J and M. J. Gruber. `Profesionally Managed Publicly Traded Commodity Funds.' Journalf of Business, v. 60 (April 1987), pp. 175–199.

Elton, E. J., M. J. Gruber, S. J. Brown and W. N. Goetzmann. Modern Portfolio Theory and Investment Analysis, 7th ed. New York: John Wiley & Sons, 2006

Elton, E. J., M. J. Gruber and M. W. Padberg. "Simple Criteria for Optimal Portfolio Selection." *Journal of Finance*, v. 31 (December 1976), pp. 1341–1357.

Guerard, J. B. Handbook of Portfolio and Construction: Contemporary Applications of Markowitz Techniques. New York: Springer, 2010.

Jagannathan, R. and T. Ma. "Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps." *Journal of Finance*, v. 58 (2003), pp. 1651–1683.

Kwan, C. C. Y. "Portfolio Analysis Using Single Index, Multi-Index, and Constant Correlation Models: A Unified Treatment." *Journal of Finance*, v. 39 (December 1984), pp. 1469–1483.

Lee, C. F. and A. C. Lee. Encyclopedia of Finance. New York: Springer, 2006

Lee, C. F., A. C. Lee and J. C. Lee . *Handbook of Quantitative Finance and Risk Management*. New York: Springer, 2010.

Lee, C. F., J. C. Lee and A. C. Lee. Statistics for Business and Financial Economics. Singapore: World Scientific Publishing Co., 2000 .

Lintner, J. "The Valuation of Risk Asset on the Selection of Risky Investments in Stock Portfolio and Capital Budgets." *The Review of Economics and Statistics*, v. 57 (February 1965), pp. 13–37.

Maginn, J. L., D. L. Tuttle, J. E. Pinto and D. W. McLeavey. Managing Investment Portfolios: A Dynamic Process, CFA Institute Investment Series, 3rd ed. New York: John Wiley & Sons, 2007.

Markowitz, H. "Portfolio Selection." *Journal of Finance*, v. 7 (March 1952), pp. 77–91.

Markowitz, H. *Portfolio Selection: Efficient Diversification of Investments.* New York: John Wiley and Sons, 1959.

Pennacchi, G. *Theory of Asset Pricing*. Boston: Addison-Wesley, 2008.

Rardin, R. *Optimization in Operations Research*. New Jersey: Prentice Hall, 1998.

Sharpe, W. F. "A Simplified Model for Portfolio Analysis." *Management Science*, v. 9 (January 1963), pp. 277–293.

Sharpe, W. F. "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk." *Journal of Finance*, v. 19 (September 1964), pp.425–442.

Wackerly, D., W. Mendenhall and R. L. Scheaffer. Mathematical Statistics with Applications, 7 th ed.California: Duxbury Press, 2007.