

Eric Zivot

Introduction to Computational Finance and Financial Econometrics with R

– Monograph –

October 21, 2016

Springer

*This book is dedicated to my beautiful wife Nina and
our children Alexander (Sasha) and Lydia.*

Preface

Outline of preface (preliminary and incomplete). June 21, 2016.

- I started teaching the course Introduction to Financial Econometrics at UW in 1998. Motivation was to teach more statistics and quantitative methods to economics majors. I found that combining statistics topics with finance applications was very effective and popular.
- Early classes used Microsoft Excel as the main software tool (R was not around then). Experience with Excel was, and still is, in high demand by employers in the finance industry. However, Excel is not a good tool for doing statistics. In early 2000s I used Excel together with S-PLUS, as I was a consultant to Insightful at that time and I was developing the S+FinMetrics software and writing the book *Modeling Financial Time Series with S-PLUS*. This hybrid approach worked well but the difficulties of getting student licenses at the beginning of the quarter etc. was always a headache. In 2008, I made the switch to R and never looked back.
- Many economics students lack exposure to programming languages (programming classes are typically not required outside of computer science majors) and find R a difficult language to learn and use. My approach to dealing with this reality was to provide all of the R code to do calculations in class and on homework assignments. In this way students did not have to worry about the coding aspect of the class and could concentrate on the concepts. I found that students learned to use R quite well this way – learning by doing – and most students were fairly proficient in R by the end of the course.
- There are a lot of free resources on the web for learning R.
- In my Coursera course I partnered with DataCamp to provide a web-based learning environment for R tailored to the course material. I have found that this has substantially lowered the learning curve for many students.
- Student mostly use either a MAC or PC for their work (a few geeks use UNIX). I found that some things didn't always work well across platforms (usually issues with the file system)
- In 2014, I required students to use Rstudio as the environment for using R. This make learning and working with R even easier. Another advantage of using Rstudio is that my lecture notes and homework assignments can be created using Rmarkdown.
- Late 1990s was the beginning of the financial econometrics revolution. There was lots of interest in Garch and volatility modeling. Campbell, Lo and MacKinlay's book was

published. My course was one of the first undergraduate courses in the U.S. to focus on statistics and finance.

- (Longer print book) Book is based on course notes for my advanced undergraduate course *Introduction to Computational Finance and Financial Econometrics* that I have taught at the University of Washington over the last 15 years. This course is taken mainly by senior economics majors who are pursuing the more quantitative BS in economics. These students typically have at least one year of calculus, intermediate microeconomics with some calculus (so they are exposed to optimization under constraints using Lagrange multipliers), and an introductory course in probability and statistics for economics majors. Some students have taken a course in linear algebra, some have taken an introductory econometrics course, and some have taken a financial economics course. I find that the typical introductory probability and statistics course is at a lower level than is needed for much of the material in my course. Ideally, students should have an intermediate probability and statistics course that uses calculus. As a result, I spend about two weeks at the beginning of the course doing a review of random variables and probability that uses basic calculus. A key part of the review is the manipulation of random variables, computing expectations of functions of random variables, etc. so that the students have the necessary tools to work with simple probabilistic models for asset returns. I also do a quick review of basic matrix algebra. Most introductory textbook treatments of portfolio theory show calculations for the simple case of two assets and then talk about how the calculations extend to the general case of many assets. Hence, students are not capable of implementing the theory for real world portfolios. I emphasize that with knowledge of just a little bit of matrix algebra (essentially matrix addition, multiplication and inversion) the general analysis of portfolio theory can be easily learned and implemented. Also, the mathematical analysis of multivariate distributions is greatly simplified using matrix algebra. I do a short introduction to time series concepts because the statistical analysis of asset returns involves using historical time series data. I define time series/stochastic process and the concepts of stationary and nonstationary time series and give simple examples of such processes that are commonly used in the modeling of financial time series. After the review material, I jump into descriptive statistical analysis of historical financial time series data. I emphasize that my approach to building models for asset returns is based on first looking at the historical data and studying its properties. I start by telling students to pretend that observed returns on assets are realizations from some unknown covariance stationary stochastic process. I review common graphical and numerical descriptive statistics (e.g., histograms, QQ-plots, sample mean, variance, correlation, etc.). The goal of descriptive statistical analysis is to compile a collection of “stylized facts” that can be used as building blocks for probabilistic models for asset returns. For example, I show that monthly asset returns are typically uncorrelated over time, have empirical distributions (histograms) that look like normal bell curves, have means close to zero, are positively correlated with other assets, etc. I also emphasize that some “stylized facts” of asset returns appear to stay the same for different investment horizons (daily, weekly, and monthly returns are uncorrelated) and some do not (e.g., daily and weekly returns have fatter tails than the normal and show evidence of nonlinear time dependence in higher moments)
- (Shorter ebook). This book is based on my University of Washington sponsored Coursera course *Introduction to Computational Finance and Financial Econometrics* that has been

running every quarter on Coursera since 2013. This Coursera course is based on the Summer 2013 offering of my University of Washington advanced undergraduate economics course of the same name. At the time, my UW course was part of a three course summer certificate in *Fundamentals of Quantitative Finance* offered by the *Professional Masters Program in Computational Finance & Risk Management* that was video-recorded and available for online students. An edited version of this course became the Coursera course. The popularity of the course encouraged me to convert the class notes for the course into a short book.

- The book was typeset using Lyx with all of the R code examples embedded into the document and evaluated using **knitr**. I am extremely grateful to Yihui Xie for writing his wonderful book *Dynamics Documents with R and knitr* as this book would not have been possible without it. I am a firm believer in the “reproducible research paradigm”.
- The R package **IntroCompFinR** contains all of the financial data (downloaded from finance.yahoo.com) used for the examples presented in the book as well as a number of R functions for portfolio and risk analysis. The package is freely available on R-forge and can be installed using

```
> install.packages("IntroCompFinR", repos="http://R-Forge.R-project.org").
```

- There are now many good books written on the topic of computational finance and financial econometrics with examples in R at various levels of sophistication. Books written for advanced U.S. undergraduates include Carmona (2014) and Tsay (2012). Books written for Masters students in computational finance or financial engineering courses include Fan and Yao (20xx), Lai and Xing (2008), Ruppert and Matteson (2015) and Tsay (2010). All of these books assume the readers have a broader mathematical and statistical background than what I assume here. Hence, this book is truly an “introduction” to the methods of computational finance and financial econometrics and is appropriate for undergraduate economics and finance majors at Universities worldwide. Another feature that distinguishes this book from others is the extensive use of R and illustrated R code throughout the book. In this regard, this book is a “hand on” book in which the student can readily execute the R examples on their own computer while reading the book. This allows the student to also learn R as well as the concepts in the book.

Seattle, Washington
June 2016

Eric Zivot

Acknowledgements

Many people have helped with the development of this book. First and foremost, I would like to thank the students at UW who have taken my course over the last fifteen years. I have received countless comments, corrections, and suggestions that have greatly improved the exposition and material in the book.

A lot of how I think about the statistical analysis of financial data has been shaped by many discussions and interactions over the last fifteen years with my colleague and good friend R. Douglas Martin. My very patient wife Nina Sidneva carefully proof-read many early versions of the chapters as they were under development and caught many of my stupid errors. Xiao Yin proof-read most of the chapters after I converted my Latex notes to Lyx and incorporated the R examples with **knitr**. I could not have written the book without the expert technical help from Bethany Yollin and the wonderful book *Dynamics Documents with R and knitr* by Yihui Xie. Bethany helped with the conversion of my Latex notes to Lyx and the incorporation of the R examples in the text using **knitr**. She also helped to develop and maintain the book's R package **IntroComppfinR**. Any problem I found with Lyx and **knitr** was solved by referring to Yihui's book. My interactions with the R community, especially those involved with R/Finance, have been invaluable for learning and using R for the analysis of financial data. In particular, Brian Peterson and Peter Carl have been incredibly helpful and supportive of my book projects. I use their wonderful R package **PortfolioAnalytics** throughout the book. Working with time series data is made so much easier with Achim Zeileis' **zoo** package and Jeff Ryan's **xts** package. Many University of Washington (UW) econ PhD students (Brian Donhauser, Ming He, Wan-Jung Hsu, Kara Ng, Anthony Sanford, Galib Sultan) acted as TAs for my undergraduate course and gave constructive feedback. I also thank Coursera for helping me convert my undergraduate course into Coursera format.

Contents

1	Return Calculations	1
1.1	The Time Value of Money	1
1.1.1	Future value, present value and simple interest	1
1.1.2	Multiple compounding periods	3
1.1.3	Effective annual rate	4
1.2	Asset Return Calculations	6
1.2.1	Simple returns	6
1.2.2	Continuously compounded returns	16
1.3	Return Calculations in R	22
1.3.1	Brute force return calculations	25
1.3.2	R packages for return calculations	27
1.4	Further reading	31
1.5	Appendix: Properties of exponentials and logarithms	32
1.6	Exercises	33
	References	35
2	Review of Random Variables	37
2.1	Random Variables	37
2.1.1	Discrete random variables	38
2.1.2	Continuous random variables	40
2.1.3	The Cumulative Distribution Function	44
2.1.4	Quantiles of the distribution of a random variable	45
2.1.5	R functions for discrete and continuous distributions	47
2.1.6	Shape characteristics of probability distributions	50
2.1.7	Linear functions of a random variable	61
2.1.8	Value at Risk: An Introduction	64
2.2	Bivariate Distributions	67
2.2.1	Discrete random variables	67

2.2.2	Bivariate distributions for continuous random variables	73
2.2.3	Independence	75
2.2.4	Covariance and correlation	76
2.2.5	Bivariate normal distribution	79
2.2.6	Expectation and variance of the sum of two random variables	81
2.3	Multivariate Distributions	83
2.3.1	Discrete random variables	84
2.3.2	Continuous random variables	84
2.3.3	Independence	85
2.3.4	Dependence concepts	85
2.3.5	Linear combinations of N random variables	86
2.3.6	Covariance between linear combinations of random variables	88
2.4	Further Reading	88
2.5	Exercises	89
	References	91
3	Matrix Algebra Review	93
3.1	Matrices and Vectors	94
3.2	Basic Matrix Operations	98
3.2.1	Addition and subtraction	98
3.2.2	Scalar multiplication	99
3.2.3	Matrix multiplication	100
3.2.4	The identity matrix	102
3.2.5	Diagonal, lower triangular and upper triangular matrices	104
3.3	Representing Summation Using Matrix Notation	106
3.4	Systems of Linear Equations	108
3.4.1	Rank of a matrix	112
3.4.2	Partitioned matrices and partitioned inverses	113
3.5	Positive Definite Matrices	115
3.5.1	Matrix square root	115
3.6	Multivariate Probability Distributions Using Matrix Algebra	117
3.6.1	Random vectors	117
3.6.2	Covariance matrix	117
3.6.3	Correlation matrix	118
3.6.4	Variance of linear combination of random vectors	119
3.6.5	Covariance between linear combination of two random vectors	119
3.6.6	Variance of a vector of linear functions of a random vector	120
3.6.7	Bivariate normal distribution	120
3.6.8	Multivariate normal distribution	121

3.7	Portfolio Math Using Matrix Algebra	122
3.8	Derivatives of Simple Matrix Functions.....	124
3.9	Further Reading.....	127
3.10	Exercises	127
	References	128
4	Time Series Concepts.....	129
4.1	Stochastic Processes	130
4.1.1	Stationary stochastic processes	130
4.1.2	Non-Stationary processes	136
4.1.3	Ergodicity	139
4.2	Multivariate Time Series	140
4.2.1	Stationary and ergodic multivariate time series	140
4.3	Time Series Models	142
4.3.1	Moving average models	143
4.3.2	Autoregressive Models	149
4.3.3	Autoregressive Moving Average Models.....	153
4.3.4	Vector Autoregressive Models	154
4.4	Further Reading.....	155
4.5	Exercises	155
5	Descriptive Statistics for Financial Data	157
5.1	Univariate Descriptive Statistics	158
5.1.1	Example data	158
5.1.2	Time plots	160
5.1.3	Descriptive statistics for the distribution of returns	164
5.1.4	QQ-plots	173
5.1.5	Shape Characteristics of the Empirical Distribution	176
5.1.6	Outliers	179
5.1.7	Box plots.....	181
5.2	Time Series Descriptive Statistics.....	183
5.2.1	Sample Autocovariances and Autocorrelations.....	183
5.2.2	Time dependence in volatility	184
5.3	Bivariate Descriptive Statistics	186
5.3.1	Scatterplots	187
5.3.2	Sample covariance and correlation	188
5.3.3	Sample cross-lag covariances and correlations	191
5.4	Stylized facts for daily and monthly asset returns	194
5.5	Further Reading.....	195
5.6	Problems	195
	References	197

6 The Constant Expected Return Model	199
6.1 CER Model Assumptions	199
6.1.1 Regression model representation.....	200
6.1.2 Location-Scale model representation	202
6.1.3 The CER Model in matrix notation.....	202
6.1.4 The CER model for continuously compounded returns	203
6.1.5 CER model for simple returns	207
6.2 Monte Carlo Simulation of the CER Model	209
6.2.1 Simulating returns on more than one asset.....	214
6.3 Further Reading.....	218
6.4 Problems	219
References	220
7 Estimation of The CER Model	221
7.1 Estimators and Estimates	222
7.1.1 Properties of Estimators	223
7.1.2 Asymptotic Properties of Estimators	226
7.2 Estimators for the Parameters of the CER Model	229
7.3 Statistical Properties of the CER Model Estimates	233
7.3.1 Bias	233
7.3.2 Precision	234
7.3.3 Sampling Distributions and Confidence Intervals	238
7.4 Using Monte Carlo Simulation to Understand the Statistical Properties of Estimators	243
7.4.1 Evaluating the Statistical Properties of $\hat{\mu}$ Using Monte Carlo Simulation.....	243
7.4.2 Evaluating the Statistical Properties of $\hat{\sigma}_i^2$ and $\hat{\sigma}_i$ Using Monte Carlo simulation	246
7.4.3 Evaluating the Statistical Properties of $\hat{\rho}_{ij}$ by Monte Carlo simulation	249
7.5 Estimating Value-at-Risk in the CER Model	250
7.5.1 Statistical Properties of \hat{q}_α^R for Simple Returns	252
7.5.2 Statistical Properties of $\widehat{\text{VaR}}_\alpha$ for Simple Returns	254
7.5.3 Using Monte Carlo Simulation to Understand the Statistical Properties of \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ for Simple Returns	255
7.5.4 Statistical Properties of \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ for Continuously Compounded Returns	257
7.6 Further Reading.....	258
References	259

8 The Bootstrap	261
8.1 The Nonparametric Bootstrap	262
8.1.1 Procedure for the nonparametric bootstrap	263
8.1.2 Performing the Nonparametric Bootstrap in R	266
8.2 Bootstrapping Time Series Data	273
8.2.1 Block bootstrap	273
8.3 Parametric Bootstrap	274
8.4 Further Reading	274
8.5 Problems	274
References	274
9 Hypothesis Testing in the CER Model	275
9.1 Hypothesis Testing: General Principles	276
9.2 Hypothesis Testing in the CER Model	279
9.3 Tests for Individual Parameters: t-tests	281
9.3.1 Exact tests under normality of data	282
9.3.2 t-tests under asymptotic normality of estimators	284
9.3.3 Relationship between hypothesis tests and confidence intervals	287
9.3.4 Tests for no autocorrelation	293
9.3.5 Tests for constant parameters	295
9.3.6 Informal diagnostics based on rolling estimators	295
9.3.7 Informal diagnostics based on exponentially weighted moving averages	302
9.4 Further Reading	302
9.5 Problems	302
References	303
10 Modeling Daily Returns with the GARCH Model	305
10.1 Engle's ARCH Model	307
10.1.1 The ARCH(1) Model	307
10.1.2 The ARCH(p) Model	313
10.2 Bollerslev's GARCH Model	314
10.2.1 Statistical Properties of the GARCH(1,1) Model	315
10.3 Maximum Likelihood Estimation	316
10.3.1 The Likelihood Function	316
10.3.2 The Maximum Likelihood Estimator	319
10.3.3 Invariance Property of Maximum Likelihood Estimators	322
10.3.4 The Precision of the Maximum Likelihood Estimator	322
10.3.5 Asymptotic Properties of Maximum Likelihood Estimators	323
10.3.6 Computing the MLE Using Numerical Optimization Methods	324

10.4	Estimation of ARCH-GARCH Models in R Using rugarch	325
10.5	Forecasting Conditional Volatility from ARCH Models	325
10.5.1	Forecasting daily return volatility from the GARCH(1,1) model	325
10.5.2	Forecasting multi-day return volatility using a GARCH(1,1) model	327
10.6	Forecasting VaR from ARCH Models	327
10.7	Further Reading	327
10.8	Problems	327
	References	328
11	Introduction to Portfolio Theory	329
11.1	Portfolios of Two Risky Assets	330
11.1.1	Portfolio expected return and variance	332
11.1.2	Portfolio Value-at-Risk	335
11.2	Efficient portfolios with two risky assets	338
11.2.1	Computing the Global Minimum Variance Portfolio	339
11.2.2	Correlation and the Shape of the Efficient Frontier	341
11.2.3	Optimal Portfolios	343
11.3	Efficient portfolios with a risk-free asset	343
11.3.1	Efficient portfolios with one risky asset and one risk-free asset	343
11.4	Efficient portfolios with two risky assets and a risk-free asset	346
11.4.1	Solving for the Tangency Portfolio	346
11.4.2	Mutual Fund Separation	348
11.4.3	Interpreting Efficient Portfolios	350
11.4.4	Efficient Portfolios and Value-at-Risk	352
11.5	Application to Real World Portfolios	353
11.6	Further Reading	357
11.7	Appendix: Review of Optimization and Constrained Optimization	357
11.8	Problems	360
	References	361
12	Portfolio Theory with Matrix Algebra	363
12.1	Portfolios with N Risky Assets	363
12.1.1	Portfolio return and risk characteristics using matrix notation	364
12.1.2	Large portfolios and diversification	370
12.2	Determining the Global Minimum Variance Portfolio Using Matrix Algebra	373
12.2.1	Alternative derivation of global minimum variance portfolio	375
12.3	Determining Mean-Variance Efficient Portfolios Using Matrix Algebra	375
12.3.1	Alternative derivation of an efficient portfolio	380
12.4	Computing the Mean-Variance Efficient Frontier	381

12.4.1 Algorithm for computing efficient frontier	386
12.5 Computing Efficient Portfolios of N risky Assets and a Risk-Free Asset Using Matrix Algebra	388
12.5.1 Computing the tangency portfolio using matrix algebra	389
12.5.2 Alternative derivation of the tangency portfolio	391
12.5.3 Mutual fund separation theorem again	393
12.6 Computational Problems with Very Large Portfolios	396
12.7 Portfolio Analysis Functions in R	396
12.8 Further Reading	402
12.9 Exercises	402
12.10 References	402
13 Portfolio Theory with Short Sales Constraints	403
13.1 Portfolio Theory with Short Sales Constraints in a Simplified Setting	404
13.1.1 Two Risky Assets	404
13.1.2 One risky asset and a risk-free asset	408
13.1.3 Two risky assets and risk-free asset	409
13.2 Portfolio Theory with Short Sales Constraints in a General Setting	412
13.2.1 Multiple risky assets	412
13.2.2 Portfolio optimization imposing no-short sales constraints	415
13.2.3 Quadratic Programming Problems	416
13.2.4 Tangency portfolio with short sales restrictions	423
13.3 Further Reading	427
13.4 Problems	428
References	429
14 Portfolio Risk Budgeting	431
14.1 Risk Budgeting Using Portfolio Variance and Portfolio Standard Deviation .	431
14.2 Euler's Theorem and Risk Decompositions	433
14.2.1 Homogenous functions of degree one	433
14.2.2 Euler's theorem	434
14.2.3 Risk decomposition using Euler's theorem	436
14.3 Understanding Portfolio Volatility Risk Decompositions	443
14.3.1 $\sigma - \rho$ decomposition for MCR_i^σ	443
14.3.2 $\sigma - \beta$ decomposition for MCR_i^σ	446
14.4 Risk Budgeting for Optimized Portfolios	447
14.5 Reverse Optimization and Implied Returns	448
14.5.1 How to Use Implied Returns	449
14.6 Risk Parity Portfolios	449
14.7 Further Reading	449

15 Statistical Analysis of Portfolios	451
15.1 Statistical Analysis of Portfolios: Two Assets	451
15.1.1 Estimation error in the portfolio frontier	460
15.1.2 Statistical properties of the global minimum variance portfolio	464
15.1.3 Statistical properties of the Sharpe ratio and the tangency portfolio	469
15.2 Statistical Analysis of Portfolios: General Case	474
15.3 Rolling Analysis of Portfolios	474
15.4 Further Reading	474
15.5 Problems	474
References	475
16 Single Index Model	477
16.1 Motivation	477
16.2 William Sharpe's Single Index Model	481
16.2.1 Economic interpretation of the SI model	483
16.2.2 Statistical properties of returns in the SI model	484
16.2.3 SI model and portfolios	486
16.2.4 The SI model in matrix notation	488
16.3 Monte Carlo Simulation of the SI Model	490
16.4 Estimation of SI Model	491
16.4.1 Plug-in principle estimates	493
16.4.2 Least squares estimates	495
16.4.3 Simple linear regression in R	497
16.4.4 Maximum likelihood estimates	497
16.5 Statistical Properties of SI Model Estimates	498
16.5.1 Bias	499
16.5.2 Precision	499
16.5.3 Sampling distribution and confidence intervals	500
16.6 Problems	500
References	501
Glossary	503
Solutions	505
Index	507

Acronyms

List(s) of abbreviations or symbols

Lists of abbreviations, symbols and the like are easily formatted with the help of the Springer-enhanced **description** environment.

ABC	Spelled-out abbreviation and definition
ARCH	Autoregressive conditional heteroskedasticity
cdf	Cumulative distribution function
CER	Constant expected return
GARCH	Generalized autoregressive conditional heteroskedasticity
pd	Positive definite
pdf	Probability distribution function
psd	Positive semi-definite
SI	Single index
VaR	Value-at-risk

Chapter 1

Return Calculations

Updated: July 7, 2016

Copyright © Eric Zivot 2015

In this chapter we cover asset return calculations with an emphasis on equity returns. It is important to understand the different ways in which asset returns are calculated because most of the models presented in the book are for asset returns. Simple returns are most commonly used for money and portfolio calculations in practice. However, simple returns have some undesirable properties that make mathematical and statistical modeling difficult. Continuously compounded returns, in contrast, have nicer properties that make mathematical and statistical modeling easier. Section 1.1 covers basic time value of money calculations. Section 1.2 covers asset return calculations, including both simple and continuously compounded returns. Section 1.3 illustrates asset return calculations using R.

This chapter uses the following R packages: **IntroCompFinR**, **PerformanceAnalytics**, **quantmod**, **xts**, and **zoo**. Make sure these packages are installed and loaded before replicating the chapter examples.

1.1 The Time Value of Money

This section reviews basic time value of money calculations. The concepts of future value, present value and the compounding of interest are defined and discussed.

1.1.1 Future value, present value and simple interest

Consider an amount $\$V$ invested for n years at a *simple interest rate* of R per annum (where R is expressed as a decimal). If compounding takes place only at the end of the year, the *future value* after n years is:

$$FV_n = \$V(1 + R) \times \cdots \times (1 + R) = \$V \cdot (1 + R)^n. \quad (1.1)$$

Over the first year, $\$V$ grows to $\$V(1 + R) = \$V + \$V \times R$ which represents the initial principal $\$V$ plus the payment of simple interest $\$V \times R$ for the year. Over the second year, the new principal $\$V(1 + R)$ grows to $\$V(1 + R)(1 + R) = \$V(1 + R)^2$, and so on.

Example 1.1. Future value with simple interest.

Consider putting \$1000 in an interest checking account that pays a simple annual percentage rate of 3%. The future value after $n = 1, 5$ and 10 years is, respectively,

$$\begin{aligned} FV_1 &= \$1000 \cdot (1.03)^1 = \$1030, \\ FV_5 &= \$1000 \cdot (1.03)^5 = \$1159.27, \\ FV_{10} &= \$1000 \cdot (1.03)^{10} = \$1343.92. \end{aligned}$$

Over the first year, \$30 in interest is accrued; over three years, \$159.27 in interest is accrued; over five years, \$343.92 in interest is accrued. These calculations are easy to perform in R

```
V = 1000
R = 0.03
FV1 = V * (1 + R)
FV5 = V * (1 + R)^5
FV10 = V * (1 + R)^10
c(FV1, FV5, FV10)

## [1] 1030 1159 1344
```

The future value formula (1.1) defines a relationship between four variables: FV_n , V , R and n . Given three variables, the fourth variable can be determined. For example, given FV_n , R and n and solving for V gives the *present value* formula:

$$V = \frac{FV_n}{(1 + R)^n}. \quad (1.2)$$

Taking FV_n , n and V as given, the annual interest rate on the investment is defined as:

$$R = \left(\frac{FV_n}{V} \right)^{1/n} - 1. \quad (1.3)$$

Finally, given FV_n , V and R we can solve for n :

$$n = \frac{\ln(FV_n/V)}{\ln(1 + R)}. \quad (1.4)$$

The expression (1.4) can be used to determine the number years it takes for an investment of $\$V$ to double. Setting $FV_n = 2V$ in (1.4) gives:

$$n = \frac{\ln(2)}{\ln(1 + R)} \approx \frac{0.7}{R},$$

which uses the approximations $\ln(2) = 0.6931 \approx 0.7$ and $\ln(1 + R) \approx R$ for R close to zero. The approximation $n \approx 0.7/R$ is called the *rule of 70*.

Example 1.2. Using the rule of 70.

The rule of 70 gives a good approximation as long as the interest rate is not too high.

```
R = seq(0.01, 0.1, by = 0.01)
nExact = log(2)/log(1 + R)
nRule70 = 0.7/R
cbind(R, nExact, nRule70)

##           R   nExact   nRule70
## [1,] 0.01 69.66    70.00
## [2,] 0.02 35.00    35.00
## [3,] 0.03 23.45    23.33
## [4,] 0.04 17.67    17.50
## [5,] 0.05 14.21    14.00
## [6,] 0.06 11.90    11.67
## [7,] 0.07 10.24    10.00
## [8,] 0.08 9.01     8.75
## [9,] 0.09 8.04     7.78
## [10,] 0.10 7.27     7.00
```

For R values between 1% and 10% the approximation error in the rule of 70 is never more than half a year.



1.1.2 Multiple compounding periods

If interest is paid m times per year, then the future value after n years is:

$$FV_n^m = \$V \cdot \left(1 + \frac{R}{m}\right)^{m \cdot n}.$$

R/m is often referred to as the *periodic interest rate*. As m , the frequency of compounding, increases, the rate becomes continuously compounded and it can be shown that the future value becomes:

$$FV_n^c = \lim_{m \rightarrow \infty} \$V \cdot \left(1 + \frac{R}{m}\right)^{m \cdot n} = \$V \cdot e^{R \cdot n},$$

where $e^{(\cdot)}$ is the exponential function and $e^1 = 2.71828$.

Example 1.3. Future value with different compounding frequencies.

For a simple annual percentage rate of 10%, the value of \$1000 at the end of one year ($n = 1$) for different values of m is calculated below.

```
V = 1000
R = 0.1
m = c(1, 2, 4, 356, 10000)
FV = V * (1 + R/m)^(m)
print(cbind(m, FV), digits = 7)
```

```
##          m      FV
## [1,]    1 1100.000
## [2,]    2 1102.500
## [3,]    4 1103.813
## [4,] 356 1105.155
## [5,] 10000 1105.170
```

The result with continuous compounding is

```
print(V * exp(R), digits = 7)
## [1] 1105.171
```

■

The continuously compounded return analogues to the present value, annual return and horizon period formulas (1.2), (1.3) and (1.4) are:

$$\begin{aligned} V &= e^{-Rn} FV_n, \\ R &= \frac{1}{n} \ln \left(\frac{FV_n}{V} \right), \\ n &= \frac{1}{R} \ln \left(\frac{FV_n}{V} \right). \end{aligned}$$

1.1.3 Effective annual rate

We now consider the relationship between simple interest rates, periodic rates, effective annual rates and continuously compounded rates. Suppose an investment pays a periodic interest rate of 2% each quarter. This gives rise to a simple annual rate of 8% ($2\% \times 4$ quarters). At the end of the year, \$1000 invested accrues to:

$$\$1000 \cdot \left(1 + \frac{0.08}{4} \right)^{4 \cdot 1} = \$1082.40.$$

The *effective annual rate*, R_A , on the investment is determined by the relationship:

$$\$1000 \cdot (1 + R_A) = \$1082.40.$$

Solving for R_A gives:

$$R_A = \frac{\$1082.40}{\$1000} - 1 = 0.0824,$$

or $R_A = 8.24\%$. Here, the effective annual rate is the simple interest rate with annual compounding that gives the same future value that occurs with simple interest compounded four times per year. The effective annual rate is greater than the simple annual rate due to the payment of interest on interest.

The general relationship between the simple annual rate R with payments m times per year and the effective annual rate, R_A , is:

$$(1 + R_A) = \left(1 + \frac{R}{m}\right)^m.$$

Given the simple rate R , we can solve for the effective annual rate using:

$$R_A = \left(1 + \frac{R}{m}\right)^m - 1. \quad (1.5)$$

Given the effective annual rate R_A , we can solve for the simple rate using:

$$R = m \left[(1 + R_A)^{1/m} - 1 \right].$$

The relationship between the effective annual rate and the simple rate that is compounded continuously is:

$$(1 + R_A) = e^R.$$

Hence,

$$\begin{aligned} R_A &= e^R - 1, \\ R &= \ln(1 + R_A). \end{aligned}$$

Example 1.4. Determine effective annual rates.

The effective annual rates associated with the investments in example 1.2 are calculated below.

```
RA = FV/V - 1
print(cbind(m, FV, RA), digits = 7)

##          m      FV       RA
## [1,]    1 1100.000 0.1000000
## [2,]    2 1102.500 0.1025000
## [3,]    4 1103.813 0.1038129
## [4,]   356 1105.155 0.1051554
## [5,] 10000 1105.170 0.1051704
```

The effective annual rate with continuous compounding is

```
print(exp(R) - 1, digits = 7)

## [1] 0.1051709
```

■

Example 1.5. Determine continuously compounded rate from effective annual rate.

Suppose an investment pays a periodic interest rate of 5% every six months ($m = 2, R/2 = 0.05$). In the market this would be quoted as having an annual percentage rate, R_A , of 10%. An investment of \$100 yields $\$100 \cdot (1.05)^2 = \110.25 after one year. The effective annual rate, R_A , is then 10.25%. To find the continuously compounded simple rate that gives the same future value as investing at the effective annual rate we solve:

$$R = \ln(1.1025) = 0.09758.$$

That is, if interest is compounded continuously at a simple annual rate of 9.758% then \$100 invested today would grow to $\$100 \cdot e^{0.09758} = \110.25

■

1.2 Asset Return Calculations

In this section, we review asset return calculations given initial and future prices associated with an investment. We first cover simple return calculations, which are typically reported in practice but are often not convenient for statistical modeling purposes. We then describe continuously compounded return calculations, which are more convenient for statistical modeling purposes.

1.2.1 Simple returns

Consider purchasing an asset (e.g., stock, bond, ETF, mutual fund, option, etc.) at time t_0 for the price P_{t_0} , and then selling the asset at time t_1 for the price P_{t_1} . If there are no intermediate cash flows (e.g., dividends) between t_0 and t_1 , the rate of return over the period t_0 to t_1 is the percentage change in price:

$$R(t_0, t_1) = \frac{P_{t_1} - P_{t_0}}{P_{t_0}}. \quad (1.6)$$

The time between t_0 and t_1 is called the *holding period* and equation (1.6) is called the *holding period return*. In principle, the holding period can be any amount of time: one second; five minutes; eight hours; two days, six minutes and two seconds; fifteen years. To simplify matters, in this chapter we will assume that the holding period is some increment of calendar time; e.g., one day, one month or one year. In particular, we will assume a default holding period of one month in what follows.

Let P_t denote the price at the end of month t of an asset that pays no dividends and let P_{t-1} denote the price at the end of month $t-1$. Then the one-month *simple net return* on an investment in the asset between months $t-1$ and t is defined as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \% \Delta P_t. \quad (1.7)$$

Writing $(P_t - P_{t-1})/(P_{t-1}) = P_t/P_{t-1} - 1$, we can define the *simple gross return* as:

$$1 + R_t = \frac{P_t}{P_{t-1}}. \quad (1.8)$$

The one-month gross return has the interpretation of the future value of \$1 invested in the asset for one-month with simple interest rate R_t . Solving (1.8) for P_t gives

$$P_t = P_{t-1}(1 + R_t),$$

which shows P_t can be thought of as the future value of P_{t-1} invested for one month with simple return R_t . Unless otherwise stated, when we refer to returns we mean net returns. Since asset prices must always be non-negative (a long position in an asset is a limited liability investment), the smallest value for R_t is -1 or -100%.

Example 1.6. Simple return calculation.

Consider a one-month investment in Microsoft stock. Suppose you buy the stock in month $t - 1$ at $P_{t-1} = \$85$ and sell the stock the next month for $P_t = \$90$. Further assume that Microsoft does not pay a dividend between months $t - 1$ and t . The one-month simple net and gross returns are then:

$$R_t = \frac{\$90 - \$85}{\$85} = \frac{\$90}{\$85} - 1 = 1.0588 - 1 = 0.0588,$$

$$1 + R_t = 1.0588.$$

The one-month investment in Microsoft yielded a 5.88% per month return. Alternatively, \$1 invested in Microsoft stock in month $t - 1$ grew to \$1.0588 in month t . The calculations using R are

```
P0 = 90
Pm1 = 85
RO = (P0 - Pm1)/Pm1
c(RO, 1 + RO)

## [1] 0.0588 1.0588
```



Multi-period returns

The simple two-month return on an investment in an asset between months $t - 2$ and t is defined as:

$$R_t(2) = \frac{P_t - P_{t-2}}{P_{t-2}} = \frac{P_t}{P_{t-2}} - 1.$$

Writing $P/P_{t-2} = P_t/P_{t-1} \cdot P_{t-1}/P_{t-2}$ the two-month return can be expressed as:

$$R_t(2) = \frac{P_t}{P_{t-1}} \cdot \frac{P_{t-1}}{P_{t-2}} - 1$$

$$= (1 + R_t)(1 + R_{t-1}) - 1.$$

Then the simple two-month gross return becomes:

$$1 + R_t(2) = (1 + R_t)(1 + R_{t-1}) = 1 + R_{t-1} + R_t + R_{t-1}R_t,$$

which is a product of the two simple one-month gross returns and not one plus the sum of the two one-month returns. Hence,

$$R_t(2) = R_{t-1} + R_t + R_{t-1}R_t.$$

If, however, R_{t-1} and R_t are small then $R_{t-1}R_t \approx 0$ and $1 + R_t(2) \approx 1 + R_{t-1} + R_t$ so that $R_t(2) \approx R_{t-1} + R_t$.

Example 1.7. Computing two-period returns.

Continuing with the previous example, suppose that the price of Microsoft stock in month $t - 2$ is \$80 and no dividend is paid between months $t - 2$ and t . The two-month net return is:

$$R_t(2) = \frac{\$90 - \$80}{\$80} = \frac{\$90}{\$80} - 1 = 1.1250 - 1 = 0.1250,$$

or 12.50% per two months. The two one-month returns are:

$$R_{t-1} = \frac{\$85 - \$80}{\$80} = 1.0625 - 1 = 0.0625,$$

$$R_t = \frac{\$90 - \$85}{\$85} = 1.0588 - 1 = 0.0588,$$

and the geometric average of the two one-month gross returns is:

$$1 + R_t(2) = 1.0625 \times 1.0588 = 1.1250.$$

The calculations can be *vectorized* in R as follows:

```
Pm2 = 80
# create vector of prices
P = c(Pm2, Pm1, P0)
# calculate vector of 1-month returns from vector of prices
R = (P[2:3] - P[1:2])/P[1:2]
R

## [1] 0.0625 0.0588

# calculate 2-month return from 2 1-month returns
(1 + R[1]) * (1 + R[2]) - 1

## [1] 0.125

# same calculation vectorized using R function cumprod()
cumprod(1 + R) - 1

## [1] 0.0625 0.1250
```

In general, the k -month gross return is defined as the product of k one-month gross returns:

$$\begin{aligned} 1 + R_t(k) &= (1 + R_t)(1 + R_{t-1}) \cdots (1 + R_{t-k+1}) \\ &= \prod_{j=0}^{k-1} (1 + R_{t-j}). \end{aligned} \tag{1.9}$$

Portfolio returns

Consider an investment of $\$V$ in two assets, named asset A and asset B . Let x_A denote the fraction or share of wealth invested in asset A , and let x_B denote the remaining fraction invested in asset B . The dollar amounts invested in assets A and B are $\$V \times x_A$ and $\$V \times x_B$, respectively. We assume that the investment shares add up to 1, so that $x_A + x_B = 1$. The collection of investment shares (x_A, x_B) defines a portfolio. For example, one portfolio may be $(x_A = 0.5, x_B = 0.5)$ and another may be $(x_A = 0, x_B = 1)$. Negative values for x_A or x_B represent short sales. Let $R_{A,t}$ and $R_{B,t}$ denote the simple one-period returns on assets A and B . We wish to determine the simple one-period return on the portfolio defined by (x_A, x_B) . To do this, note that at the end of period t , the investments in assets A and B are worth $\$V \times x_A(1 + R_{A,t})$ and $\$V \times x_B(1 + R_{B,t})$, respectively. Hence, at the end of period t the portfolio is worth:

$$\$V \times [x_A(1 + R_{A,t}) + x_B(1 + R_{B,t})].$$

Hence, $x_A(1 + R_{A,t}) + x_B(1 + R_{B,t})$ defines the gross return on the portfolio. The portfolio gross return is equal to a weighted average of the gross returns on assets A and B , where the weights are the portfolio shares x_A and x_B .

To determine the portfolio rate of return, re-write the portfolio gross return as:

$$1 + R_{p,t} = x_A + x_B + x_A R_{A,t} + x_B R_{B,t} = 1 + x_A R_{A,t} + x_B R_{B,t},$$

since $x_A + x_B = 1$ by construction. Then the portfolio rate of return is:

$$R_{p,t} = x_A R_{A,t} + x_B R_{B,t},$$

which is equal to a weighted average of the simple returns on assets A and B , where the weights are the portfolio shares x_A and x_B .

Example 1.8. Compute portfolio return.

Consider a portfolio of Microsoft and Starbucks stock in which you initially purchase ten shares of each stock at the end of month $t - 1$ at the prices $P_{\text{msft},t-1} = \$85$ and $P_{\text{sbux},t-1} = \$30$, respectively. The initial value of the portfolio is $V_{t-1} = 10 \times \$85 + 10 \times 30 = \$1,150$. The portfolio shares are $x_{\text{msft}} = 850/1150 = 0.7391$ and $x_{\text{sbux}} = 30/1150 = 0.2609$. Suppose at the end of month t , $P_{\text{msft},t} = \$90$ and $P_{\text{sbux},t} = \$28$. Assuming that Microsoft and Starbucks do not pay a dividend between periods $t - 1$ and t , the one-period returns on the two stocks are:

$$R_{\text{msft},t} = \frac{\$90 - \$85}{\$85} = 0.0588,$$

$$R_{\text{sbux},t} = \frac{\$28 - \$30}{\$30} = -0.0667.$$

The one-month rate of return on the portfolio is then:

$$R_{p,t} = (0.7391)(0.0588) + (0.2609)(-0.0667) = 0.02609,$$

and the portfolio value at the end of month t is:

$$V_t = V_{t-1}(1 + R_{p,t}) = \$1,150 \times (1.02609) = \$1,180$$

In R, the portfolio calculations are

```
P.msft = c(85, 90)
P.sbux = c(30, 28)
V = P.msft[1] * 10 + P.sbux[1] * 10
x.msft = 10 * P.msft[1]/V
x.sbux = 10 * P.sbux[1]/V
R.msft = (P.msft[2] - P.msft[1])/P.msft[1]
R.sbux = (P.sbux[2] - P.sbux[1])/P.sbux[1]
R.p = x.msft * R.msft + x.sbux * R.sbux
R.p

## [1] 0.0261

# portfolio end-of-month value
V1 = V * (1 + R.p)
V1

## [1] 1180
```

In general, for a portfolio of n assets with investment shares x_i such that $x_1 + \dots + x_n = 1$, the one-period portfolio gross and simple returns are defined as:

$$1 + R_{p,t} = \sum_{i=1}^n x_i(1 + R_{i,t}) \quad (1.10)$$

$$R_{p,t} = \sum_{i=1}^n x_i R_{i,t}. \quad (1.11)$$

Adjusting for dividends

If an asset pays a dividend, D_t , sometime between months $t-1$ and t , the *total net return* calculation becomes:

$$R_t^{\text{total}} = \frac{P_t + D_t - P_{t-1}}{P_{t-1}} = \frac{P_t - P_{t-1}}{P_{t-1}} + \frac{D_t}{P_{t-1}}, \quad (1.12)$$

where $(P_t - P_{t-1})/P_{t-1}$ is referred as the *capital gain* and D_t/P_{t-1} is referred to as the *dividend yield*. The *total gross return* is:

$$1 + R_t^{\text{total}} = \frac{P_t + D_t}{P_{t-1}}. \quad (1.13)$$

The formula (1.9) for computing multi-period return remains the same except that one-period gross returns are computed using (1.13).

Example 1.9. Compute total return when dividends are paid.

Consider a one-month investment in Microsoft stock. Suppose you buy the stock in month $t - 1$ at $P_{t-1} = \$85$ and sell the stock the next month for $P_t = \$90$. Further assume that Microsoft pays a \$1 dividend between months $t - 1$ and t . The capital gain, dividend yield and total return are then:

$$\begin{aligned} R_t &= \frac{\$90 + \$1 - \$85}{\$85} = \frac{\$90 - \$85}{\$85} + \frac{\$1}{\$85} \\ &= 0.0588 + 0.0118 \\ &= 0.0707. \end{aligned}$$

The one-month investment in Microsoft yields a 7.07% per month total return. The capital gain component is 5.88% and the dividend yield component is 1.18%

■

Adjusting for inflation

The return calculations considered so far are based on the *nominal* or current prices of assets. Returns computed from nominal prices are nominal returns. The *real return* on an asset over a particular horizon takes into account the growth rate of the general price level over the horizon. If the nominal price of the asset grows faster than the general price level, then the nominal return will be greater than the inflation rate and the real return will be positive. Conversely, if the nominal price of the asset increases less than the general price level, then the nominal return will be less than the inflation rate and the real return will be negative.

The computation of real returns on an asset is a two-step process:

- Deflate the nominal price of the asset by the general price level
- Compute returns in the usual way using the deflated prices

To illustrate, consider computing the real simple one-period return on an asset. Let P_t denote the nominal price of the asset at time t and let CPI_t denote an index of the general price level (e.g. consumer price index) at time t ¹. The deflated or real price at time t is:

¹ The CPI is usually normalized to 1 or 100 in some base year. We assume that the CPI is normalized to 1 in the base year for simplicity.

$$P_t^{\text{Real}} = \frac{P_t}{CPI_t},$$

and the real one-period return is:

$$\begin{aligned} R_t^{\text{Real}} &= \frac{P_t^{\text{Real}} - P_{t-1}^{\text{Real}}}{P_{t-1}^{\text{Real}}} = \frac{\frac{P_t}{CPI_t} - \frac{P_{t-1}}{CPI_{t-1}}}{\frac{P_{t-1}}{CPI_{t-1}}} \\ &= \frac{P_t}{P_{t-1}} \cdot \frac{CPI_{t-1}}{CPI_t} - 1. \end{aligned} \quad (1.14)$$

The one-period gross real return is:

$$1 + R_t^{\text{Real}} = \frac{P_t}{P_{t-1}} \cdot \frac{CPI_{t-1}}{CPI_t}. \quad (1.15)$$

If we define inflation between periods $t - 1$ and t as:

$$\pi_t = \frac{CPI_t - CPI_{t-1}}{CPI_{t-1}} = \% \Delta CPI_t, \quad (1.16)$$

then $1 + \pi_t = CPI_t / CPI_{t-1}$ and (1.14) may be re-expressed as:

$$R_t^{\text{Real}} = \frac{1 + R_t}{1 + \pi_t} - 1. \quad (1.17)$$

Example 1.10. Compute real return.

Consider, again, a one-month investment in Microsoft stock. Suppose the CPI in months $t - 1$ and t is 1 and 1.01, respectively, representing a 1% monthly growth rate in the overall price level. The real prices of Microsoft stock are:

$$P_{t-1}^{\text{Real}} = \frac{\$85}{1} = \$85, \quad P_t^{\text{Real}} = \frac{\$90}{1.01} = \$89.1089,$$

and the real monthly return is:

$$R_t^{\text{Real}} = \frac{\$89.1089 - \$85}{\$85} = 0.0483.$$

The nominal return and inflation over the month are:

$$R_t = \frac{\$90 - \$85}{\$85} = 0.0588, \quad \pi_t = \frac{1.01 - 1}{1} = 0.01.$$

Then the real return computed using (1.17) is:

$$R_t^{\text{Real}} = \frac{1.0588}{1.01} - 1 = 0.0483.$$

Notice that simple real return is almost, but not quite, equal to the simple nominal return minus the inflation rate:

$$R_t^{\text{Real}} \approx R_t - \pi_t = 0.0588 - 0.01 = 0.0488.$$



Return on foreign investment

It is common for an investor in one country to buy an asset in another country as an investment. If the two countries have different currencies then the exchange rate between the two currencies influences the return on the asset. To illustrate, consider an investment in a foreign stock (e.g. a stock trading on the London stock exchange) by a U.S. national (domestic investor). The domestic investor takes U.S. dollars, converts them to the foreign currency (e.g. British Pound) via the exchange rate (price of foreign currency in U.S. dollars) and then purchases the foreign stock using the foreign currency. When the stock is sold, the proceeds in the foreign currency must then be converted back to the domestic currency.

To be more precise, consider the information in the table below:

Month	U.S. \$ Cost of 1 Pound £, $E_t^{US/UK}$	Value of British Shares, P_t^{UK}	Value in U.S. \$, P_t^{US}
$t - 1$	\$1.50	£40	$\$1.50 \times £40 = \60
t	\$1.30	£45	$\$1.30 \times £45 = \58.5

Let P_t^{US} , P_t^{UK} , and $E_t^{US/UK}$ denote the U.S. dollar price of the asset, British Pound price of the asset, and currency exchange rate between U.S. dollars and British Pounds, at the end of month t , respectively. The U.S. dollar cost of one share of British stock in month $t - 1$ is

$$P_{t-1}^{US} = E_{t-1}^{US/UK} \times P_{t-1}^{UK} = \$1.50 \times £40 = \$60,$$

and the U.S. dollar cost at time t is

$$P_t^{US} = E_t^{US/UK} \times P_t^{UK} = \$1.30 \times £45 = \$58.5.$$

The U.S. dollar rate of return is then

$$R_t^{US} = \frac{P_t^{US} - P_{t-1}^{US}}{P_{t-1}^{US}} = \frac{\$58.5 - \$60}{\$60} = -0.025.$$

Notice that

$$R_t^{US} = \frac{E_t^{US/UK} \times P_t^{UK} - E_{t-1}^{US/UK} \times P_{t-1}^{UK}}{E_{t-1}^{US/UK} \times P_{t-1}^{UK}} = \frac{E_t^{US/UK} \times P_t^{UK}}{E_{t-1}^{US/UK} \times P_{t-1}^{UK}} - 1 \quad (1.18)$$

$$= (1 + R_t^{US/UK})(1 + R_t^{UK}) - 1, \quad (1.19)$$

where

$$\frac{E_t^{US/UK}}{E_{t-1}^{US/UK}} = 1 + R_t^{US/UK}, \quad \frac{P_t^{UK}}{P_{t-1}^{UK}} = 1 + R_t^{UK}.$$

Here, $R_t^{US/UK}$ is the simple rate of return on foreign currency and R_t^{UK} is the simple rate of return on the British stock in British Pounds:

$$R_t^{US/UK} = \frac{E_t^{US/UK} - E_{t-1}^{US/UK}}{E_{t-1}^{US/UK}} = \frac{\$1.30 - \$1.50}{\$1.50} = -0.133,$$

$$R_t^{UK} = \frac{P_t^{UK} - P_{t-1}^{UK}}{P_{t-1}^{UK}} = \frac{\text{£}45 - \text{£}40}{\text{£}40} = 0.125.$$

Hence, the gross return, $1 + R_t^{US}$, for the U.S. investor has two parts: (1) the gross return from an investment in foreign currency, $1 + R_t^{US/UK}$; and (2) the gross return of the foreign asset in the foreign currency, $1 + R_t^{UK}$:

$$1 + R_t^{US} = (1 + R_t^{US/UK})(1 + R_t^{UK}) \\ = (0.867)(1.125) \\ = 0.975.$$

In this example, the U.S. investor loses money even though the British investment has a positive rate of return because of the large drop in the exchange rate.

Annualizing returns

Very often returns over different horizons are annualized, i.e., converted to an annual return, to facilitate comparisons with other investments. The annualization process depends on the holding period of the investment and an implicit assumption about compounding. We illustrate with several examples.

To start, if our investment horizon is one year then the annual gross and net returns are just:

$$1 + R_A = 1 + R_t(12) = \frac{P_t}{P_{t-12}} = (1 + R_t)(1 + R_{t-1}) \cdots (1 + R_{t-11}),$$

$$R_A = R_t(12).$$

In this case, no compounding is required to create an annual return.

Next, consider a one-month investment in an asset with return R_t . What is the annualized return on this investment? If we assume that we receive the same return $R = R_t$ every month for the year, then the gross annual return is:

$$1 + R_A = 1 + R_t(12) = (1 + R)^{12}.$$

That is, the annual gross return is defined as the monthly return compounded for 12 months. The net annual return is then:

$$R_A = (1 + R)^{12} - 1.$$

Example 1.11. Compute annualized return from one-month return.

In the first example, the one-month return, R_t , on Microsoft stock was 5.88%. If we assume that we can get this return for 12 months then the annualized return is:

$$R_A = (1.0588)^{12} - 1 = 1.9850 - 1 = 0.9850,$$

or 98.50% per year. Pretty good!

■

Now, consider a two-month investment with return $R_t(2)$. If we assume that we receive the same two-month return $R(2) = R_t(2)$ for the next six two-month periods, then the gross and net annual returns are:

$$\begin{aligned} 1 + R_A &= (1 + R(2))^6, \\ R_A &= (1 + R(2))^6 - 1. \end{aligned}$$

Here the annual gross return is defined as the two-month return compounded for 6 months.

Example 1.12. Compute annualized return from two-month return.

Suppose the two-month return, $R_t(2)$, on Microsoft stock is 12.5%. If we assume that we can get this two-month return for the next 6 two-month periods then the annualized return is:

$$R_A = (1.1250)^6 - 1 = 2.0273 - 1 = 1.0273,$$

or 102.73% per year.

■

Now suppose that our investment horizon is two years. That is, we start our investment at time $t - 24$ and cash out at time t . The two-year gross return is then $1 + R_t(24) = P_t/P_{t-24}$. What is the annual return on this two-year investment? The process is the same as computing the effective annual rate. To determine the annual return we solve the following relationship for R_A :

$$\begin{aligned} (1 + R_A)^2 &= 1 + R_t(24) \implies \\ R_A &= (1 + R_t(24))^{1/2} - 1. \end{aligned}$$

In this case, the annual return is compounded twice to get the two-year return and the relationship is then solved for the annual return.

Example 1.13. Compute annualized return from two-year return.

Suppose that the price of Microsoft stock 24 months ago is $P_{t-24} = \$50$ and the price today is $P_t = \$90$. The two-year gross return is $1 + R_t(24) = \$90/\$50 = 1.8000$ which yields a two-year net return of $R_t(24) = 0.80 = 80\%$. The annual return for this investment is defined as:

$$R_A = (1.800)^{1/2} - 1 = 1.3416 - 1 = 0.3416,$$

or 34.16% per year.

■

Average returns

For investments over a given horizon, it is often of interest to compute a measure of the average rate of return over the horizon. To illustrate, consider a sequence of monthly investments over T months with monthly returns R_1, R_2, \dots, R_T . The T -month return is

$$R(T) = (1 + R_1)(1 + R_2) \cdots (1 + R_T) - 1.$$

What is the average monthly return? There are two possibilities. The first is the *arithmetic average return*

$$\bar{R}^A = \frac{1}{T} (R_1 + R_2 + \cdots + R_T).$$

The second is the *geometric average return*

$$\bar{R}^G = [(1 + R_1)(1 + R_2) \cdots (1 + R_T)]^{1/T} - 1.$$

Notice that the geometric average return is the monthly return which compounded monthly for T months gives the gross T -month return

$$(1 + \bar{R}^G)^T = 1 + R(T).$$

As a measure of average investment performance, the geometric average return is preferred to the arithmetic average return. To see why, consider a two-month investment horizon with monthly returns $R_1 = 0.5$ and $R_2 = -0.5$. One dollar invested over these two months grows to $(1 + R_1)(1 + R_2) = (1.5)(0.5) = 0.75$, for a two-period return of $R(2) = 0.75 - 1 = -0.25$. Hence, the two-month investment loses 25%. The arithmetic average return is $\bar{R}^A = \frac{1}{2}(0.5 + (-0.5)) = 0$. As a measure of investment performance, this is misleading because the investment loses money over the two-month horizon. The average return indicates the investment breaks even. The geometric average return is $\bar{R}^G = [(1.5)(0.5)]^{1/2} - 1 = -0.134$. This is a more accurate measure of investment performance because it indicates that the investment eventually loses money. In addition, the compound two-month return using $\bar{R}^G = -0.134$ is equal to $R(2) = -0.25$.

1.2.2 Continuously compounded returns

In this section we define continuously compounded returns from simple returns, and describe their properties.

One-period returns

Let R_t denote the simple monthly return on an investment. The *continuously compounded monthly return*, r_t , is defined as:

$$r_t = \ln(1 + R_t) = \ln \left(\frac{P_t}{P_{t-1}} \right), \quad (1.20)$$

where $\ln(\cdot)$ is the natural log function². To see why r_t is called the continuously compounded return, take the exponential of both sides of (1.20) to give:

$$e^{r_t} = 1 + R_t = \frac{P_t}{P_{t-1}}.$$

Rearranging we get:

$$P_t = P_{t-1} e^{r_t},$$

so that r_t is the continuously compounded growth rate in prices between months $t-1$ and t . This is to be contrasted with R_t , which is the simple growth rate in prices between months $t-1$ and t without any compounding. As a result, r_t is always smaller than R_t . Furthermore, since $\ln(x/y) = \ln(x) - \ln(y)$ it follows that:

$$\begin{aligned} r_t &= \ln\left(\frac{P_t}{P_{t-1}}\right) \\ &= \ln(P_t) - \ln(P_{t-1}) \\ &= p_t - p_{t-1}, \end{aligned}$$

where $p_t = \ln(P_t)$. Hence, the continuously compounded monthly return, r_t , can be computed simply by taking the first difference of the natural logarithms of monthly prices.

If the simple return R_t is close to zero ($R_t \approx 0$) then $r_t = \ln(1+R_t) \approx R_t$. This result can be shown using a first order Taylor series approximation to the function $f(x) = \ln(1+x)$. Recall, for a continuous and differentiable function $f(x)$ the first order Taylor series approximation of $f(x)$ about the point $x = x_0$ is

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \text{remainder},$$

where $f'(x_0)$ denotes the derivative of $f(x)$ evaluated at x_0 . Let $f(x) = \ln(1+x)$ and set $x_0 = 0$. Note that

$$f(0) = \ln(1) = 0, \quad f'(x) = \frac{1}{1+x}, \quad f'(0) = 1.$$

Then, using the first order Taylor series approximation, $\ln(1+x) \approx x$ when $x \approx 0$.

Example 1.14. Compute continuously compounded returns.

Using the price and return data from Example 1, the continuously compounded monthly return on Microsoft stock can be computed in two ways:

$$\begin{aligned} r_t &= \ln(1.0588) = 0.0571 \\ r_t &= \ln(90) - \ln(85) = 4.4998 - 4.4427 = 0.0571 \end{aligned}$$

Notice that r_t is slightly smaller than R_t . Using R, the calculations are

² The continuously compounded return is always defined since asset prices, P_t , are always non-negative. Properties of logarithms and exponentials are discussed in the appendix to this chapter.

```
c(log(1 + R0), log(P0) - log(Pm1))
## [1] 0.0572 0.0572
```

Notice that the R function `log()` computes the natural logarithm.

Given a monthly continuously compounded return r_t , it is straightforward to solve back for the corresponding simple net return R_t :

$$R_t = e^{r_t} - 1. \quad (1.21)$$

Hence, nothing is lost by considering continuously compounded returns instead of simple returns. Continuously compounded returns are very similar to simple returns as long as the return is relatively small, which it generally will be for monthly or daily returns. Since R_t is bounded from below by -1, the smallest value for r_t is $-\infty$. This, however, does not mean that you could lose an infinite amount of money on an investment. The actual amount of money lost is determined by the simple return (1.21).

Example 1.15. Determine simple returns from continuously compounded returns.

In the previous example, the continuously compounded monthly return on Microsoft stock is $r_t = 5.71\%$. The simple net return is then:

$$R_t = e^{0.0571} - 1 = 0.0588.$$

For asset return modeling and statistical analysis it is often more convenient to use continuously compounded returns than simple returns due to the additivity property of multi-period continuously compounded returns discussed in the next sub-section.

Multi-period returns

The relationship between multi-period continuously compounded returns and one-period continuously compounded returns is more simple than the relationship between multi-period simple returns and one-period simple returns. To illustrate, consider the two-month continuously compounded return defined as:

$$r_t(2) = \ln(1 + R_t(2)) = \ln\left(\frac{P_t}{P_{t-2}}\right) = p_t - p_{t-2}.$$

Taking exponentials of both sides shows that:

$$P_t = P_{t-2}e^{r_t(2)}$$

so that $r_t(2)$ is the continuously compounded growth rate of prices between months $t - 2$ and t . Using $P_t/P_{t-2} = (P_t/P_{t-1}) \cdot (P_{t-1}/P_{t-2})$ and the fact that $\ln(x \cdot y) = \ln(x) + \ln(y)$

it follows that:

$$\begin{aligned} r_t(2) &= \ln \left(\frac{P_t}{P_{t-1}} \cdot \frac{P_{t-1}}{P_{t-2}} \right) \\ &= \ln \left(\frac{P_t}{P_{t-1}} \right) + \ln \left(\frac{P_{t-1}}{P_{t-2}} \right) \\ &= r_t + r_{t-1}. \end{aligned}$$

Hence the continuously compounded two-month return is just the sum of the two continuously compounded one-month returns. Recall, with simple returns the two-month return is a multiplicative (geometric) sum of two one-month returns.

Example 1.16. Compute two-month continuously compounded returns.

Using the data from Example 2, the continuously compounded two-month return on Microsoft stock can be computed in two equivalent ways. The first way uses the difference in the logs of P_t and P_{t-2} :

$$r_t(2) = \ln(90) - \ln(80) = 4.4998 - 4.3820 = 0.1178.$$

The second way uses the sum of the two continuously compounded one-month returns. Here $r_t = \ln(90) - \ln(85) = 0.0571$ and $r_{t-1} = \ln(85) - \ln(80) = 0.0607$ so that:

$$r_t(2) = 0.0571 + 0.0607 = 0.1178.$$

Notice that $r_t(2) = 0.1178 < R_t(2) = 0.1250$. The vectorized calculations in R are

```
p = log(P)
r = log(1 + R)
c(r[1] + r[2], sum(r), p[3] - p[1], diff(p, 2))

## [1] 0.118 0.118 0.118 0.118
```



The continuously compounded k -month return is defined by:

$$r_t(k) = \ln(1 + R_t(k)) = \ln \left(\frac{P_t}{P_{t-k}} \right) = p_t - p_{t-k}.$$

Using similar manipulations to the ones used for the continuously compounded two-month return, we can express the continuously compounded k -month return as the sum of k continuously compounded monthly returns:

$$r_t(k) = \sum_{j=0}^{k-1} r_{t-j}. \quad (1.22)$$

The additivity of continuously compounded returns to form multiperiod returns is an important property for statistical modeling purposes.

Portfolio returns

The continuously compounded portfolio return is defined by (1.20), where R_t is computed using the portfolio return (1.11). However, notice that:

$$r_{p,t} = \ln(1 + R_{p,t}) = \ln\left(1 + \sum_{i=1}^n x_i R_{i,t}\right) \neq \sum_{i=1}^n x_i r_{i,t}, \quad (1.23)$$

where $r_{i,t}$ denotes the continuously compounded one-period return on asset i . If the portfolio return $R_{p,t} = \sum_{i=1}^n x_i R_{i,t}$ is not too large then $r_{p,t} \approx R_{p,t}$ otherwise, $R_{p,t} > r_{p,t}$.

Example 1.17. Compute continuously compounded portfolio returns.

Consider a portfolio of Microsoft and Starbucks stock with $x_{msft} = 0.25$, $x_{sbux} = 0.75$, $R_{msft,t} = 0.0588$, $R_{sbux,t} = -0.0503$ and $R_{p,t} = -0.02302$. Using (1.23), the continuously compounded portfolio return is:

$$r_{p,t} = \ln(1 - 0.02302) = \ln(0.977) = -0.02329.$$

Using $r_{msft,t} = \ln(1 + 0.0588) = 0.0572$ and $r_{sbux,t} = \ln(1 - 0.0503) = -0.05161$, notice that:

$$x_{msft} r_{msft} + x_{sbux} r_{sbux} = -0.02442 \neq r_{p,t}$$

■

Adjusting for dividends

The continuously compounded one-period return adjusted for dividends is defined by (1.20), where R_t is computed using (1.12)³.

Example 1.18. Compute continuously compounded total return.

From example 1.9, the total simple return using (1.12) is $R_t = 0.0707$. The continuously compounded total return is then:

$$r_t = \ln(1 + R_t) = \ln(1.0707) = 0.0683.$$

■

Adjusting for inflation

Adjusting continuously compounded nominal returns for inflation is particularly simple. The continuously compounded one-period real return is defined as:

³ Show formula from Campbell, Lo and MacKinlay.

$$r_t^{\text{Real}} = \ln(1 + R_t^{\text{Real}}). \quad (1.24)$$

Using (1.15), it follows that:

$$\begin{aligned} r_t^{\text{Real}} &= \ln\left(\frac{P_t}{P_{t-1}} \cdot \frac{CPI_{t-1}}{CPI_t}\right) \\ &= \ln\left(\frac{P_t}{P_{t-1}}\right) + \ln\left(\frac{CPI_{t-1}}{CPI_t}\right) \\ &= \ln(P_t) - \ln(P_{t-1}) + \ln(CPI_{t-1}) - \ln(CPI_t) \\ &= \ln(P_t) - \ln(P_{t-1}) - (\ln(CPI_t) - \ln(CPI_{t-1})) \\ &= r_t - \pi_t^c, \end{aligned} \quad (1.25)$$

where $r_t = \ln(P_t) - \ln(P_{t-1}) = \ln(1 + R_t)$ is the nominal continuously compounded one-period return and $\pi_t^c = \ln(CPI_t) - \ln(CPI_{t-1}) = \ln(1 + \pi_t)$ is the one-period continuously compounded growth rate in the general price level (continuously compounded one-period inflation rate). Hence, the real continuously compounded return is simply the nominal continuously compounded return minus the the continuously compounded inflation rate.

Example 1.19. Compute continuously compounded real returns.

From example 1.10, the nominal simple return is $R_t = 0.0588$, the monthly inflation rate is $\pi_t = 0.01$, and the real simple return is $R_t^{\text{Real}} = 0.0483$. Using (1.24), the real continuously compounded return is:

$$r_t^{\text{Real}} = \ln(1 + R_t^{\text{Real}}) = \ln(1.0483) = 0.047.$$

Equivalently, using (1.25) the real return may also be computed as:

$$r_t^{\text{Real}} = r_t - \pi_t^c = \ln(1.0588) - \ln(1.01) = 0.047.$$

■

Annualizing continuously compounded returns

Just as we annualized simple monthly returns, we can also annualize continuously compounded monthly returns. For example, if our investment horizon is one year then the annual continuously compounded return is just the sum of the twelve monthly continuously compounded returns:

$$\begin{aligned} r_A &= r_t(12) = r_t + r_{t-1} + \cdots + r_{t-11} \\ &= \sum_{j=0}^{11} r_{t-j}. \end{aligned}$$

The average continuously compounded monthly return is defined as:

$$\bar{r}_m = \frac{1}{12} \sum_{j=0}^{11} r_{t-j}.$$

Notice that:

$$12 \cdot \bar{r}_m = \sum_{j=0}^{11} r_{t-j}$$

so that we may alternatively express r_A as:

$$r_A = 12 \cdot \bar{r}_m.$$

That is, the continuously compounded annual return is twelve times the average of the continuously compounded monthly returns.

As another example, consider a one-month investment in an asset with continuously compounded return r_t . What is the continuously compounded annual return on this investment? If we assume that we receive the same return $r = r_t$ every month for the year, then the annual continuously compounded return is just 12 times the monthly continuously compounded return:

$$r_A = r_t(12) = 12 \cdot r.$$

1.3 Return Calculations in R

This section discusses the calculation of returns from historical prices in R, as well as the graphical display of prices and returns. The examples in this section are based on the daily adjusted closing price data for Microsoft and Starbucks stock over the period January 4, 1993 through December 31, 2014.⁴ These data are available as the “xts” objects `msftDailyPrices` and `sbuxDailyPrices` in the R package `IntroCompFinR`.⁵

```
library(IntroCompFinR)
library(xts)
library(methods)
data(msftDailyPrices, sbuxDailyPrices)
str(msftDailyPrices)

## An 'xts' object on 1993-01-04/2014-12-31 containing:
##   Data: num [1:5541, 1] 1.89 1.92 1.98 1.94 1.94 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : NULL
##     ..$ : chr "MSFT"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##   NULL
```

⁴ These are prices that are adjusted for dividends and stock splits. That is, any dividend payments have been included in the prices and historical prices have been divided by the split ratio associated with any stock splits.

⁵ See the Appendix Working with Time Series Data in R for an overview of “xts” objects and Zivot (2016) for a comprehensive coverage.

End-of-month prices can be extracted from the daily prices using the `xts` function `to.monthly()`

```
msftMonthlyPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
sbuxMonthlyPrices = to.monthly(sbuxDailyPrices, OHLC = FALSE)
str(msftMonthlyPrices)

## An 'xts' object on Jan 1993/Dec 2014 containing:
##   Data: num [1:264, 1] 1.92 1.85 2.06 1.9 2.06 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : NULL
##     ..$ : chr "MSFT"
##   Indexed by objects of class: [yearmon] TZ: UTC
##   xts Attributes:
##   NULL
```

There are several advantages of using "`xts`" objects to represent financial time series data. You can easily extract observations between specific dates. For example, to extract daily prices between January 3, 2014 and January 7, 2014, use

```
msftDailyPrices["2014-01-03::2014-01-07"]

##           MSFT
## 2014-01-03 35.7
## 2014-01-06 34.9
## 2014-01-07 35.2
```

Or to extract monthly prices between January, 2014 and March, 2014 use

```
msftMonthlyPrices["2014-01::2014-03"]

##           MSFT
## Jan 2014 36.6
## Feb 2014 37.3
## Mar 2014 39.9
```

Two or more "`xts`" objects can be merged together and aligned to a common date index using the `xts` function `merge()`:

```
msftSbuxMonthlyPrices = merge(msftMonthlyPrices, sbuxMonthlyPrices)
head(msftSbuxMonthlyPrices, n = 3)

##           MSFT SBUX
## Jan 1993 1.92 1.12
## Feb 1993 1.85 1.05
## Mar 1993 2.06 1.10
```

Time plots of “`xts`” objects can be created with the generic `plot()` function as there is a method function in the `xts` package for objects of class “`xts`”. For example, Figure 1.3 shows a basic time plot of the monthly closing prices of Microsoft and Starbucks.⁶

⁶ In this book I use the development version of `xts` available on github at <https://github.com/joshuaulrich/xts>. This version has a substantially improved `plot.xts()` function.

```
plot(msftSbuxMonthlyPrices, legend.loc = "topleft")
```



Fig. 1.1 Single panel plot (default) with `plot.xts()`.

The default plot style in `plot.xts()` is a single-panel plot with multiple series. You can also create multi-panel plots, as in Figure 1.2, by setting the optional argument `multi.panel = TRUE` in the call to `plot.xts()`

```
plot(msftSbuxMonthlyPrices, multi.panel = TRUE)
```

Because the “`xts`” class inherits from the “`zoo`” class, the `zoo` method function `plot.zoo()` can also be used to plot “`xts`” objects. Figure 1.3 is created with

```
plot.zoo(msftSbuxMonthlyPrices, plot.type = "single", lwd = 2, col = c("black",
  "red"), ylab = "Price")
grid()
legend(x = "topleft", legend = colnames(msftSbuxMonthlyPrices), lwd = 2, col = c("black",
  "red"))
```

A muti-panel plot, shown in Figure 1.3, can be created using

```
plot.zoo(msftSbuxMonthlyPrices, plot.type = "multiple", main = "", lwd = 2, col = c("black",
  "red"))
```

More examples of manipulating and plotting “`xts`” objects are given in the end-of-chapter exercises.

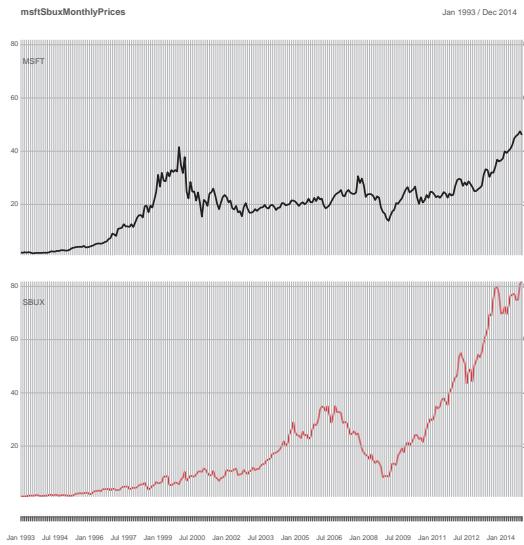


Fig. 1.2 Multi-panel time series plot with `plot.xts()`

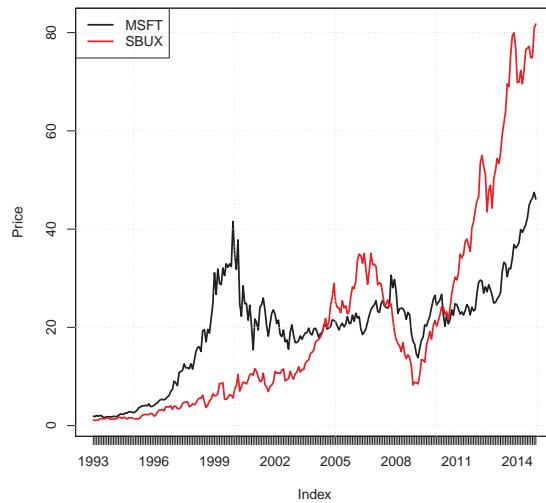


Fig. 1.3 Single panel time series plot (default) with `plot.zoo()`

1.3.1 Brute force return calculations

Consider computing simple monthly returns, $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$, from historical prices using the “xts” object `sbuxMonthlyPrices`. The R code for a brute force calculation is

```
sbuxMonthlyReturns = diff(sbuxMonthlyPrices)/lag(sbuxMonthlyPrices)
head(sbuxMonthlyReturns, n = 3)

##          SBUX
#> [1] 0.0000000
#> [2] 0.0000000
#> [3] 0.0000000
```

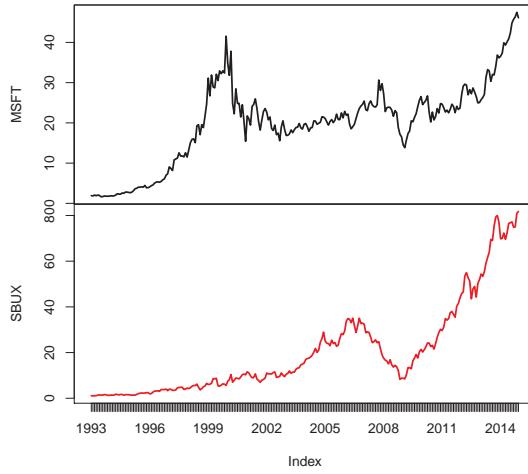


Fig. 1.4 Multi-panel time series plot with `plot.zoo()`.

```
## Jan 1993      NA
## Feb 1993 -0.0625
## Mar 1993  0.0476
```

Here, the `diff()` function computes the first difference in the prices, $P_t - P_{t-1}$, and the `lag()` function computes the lagged price, P_{t-1} .⁷ An equivalent calculation is $R_t = \frac{P_t}{P_{t-1}} - 1$:

```
head(sbuxMonthlyPrices/lag(sbuxMonthlyPrices) - 1, n = 3)

##                  SBUX
## Jan 1993      NA
## Feb 1993 -0.0625
## Mar 1993  0.0476
```

Notice that the return for January, 1993 is NA (missing value). To automatically remove this missing value, use the R function `na.omit()` or explicitly exclude the first observation:

```
head(na.omit(sbuxMonthlyReturns), n = 3)

##                  SBUX
## Feb 1993 -0.0625
## Mar 1993  0.0476
## Apr 1993  0.0182

head(sbuxMonthlyReturns[-1])

##                  SBUX
```

⁷ The method function `lag.xts()` works differently than generic `lag()` function and the method function `lag.zoo()`. In particular, `lag.xts()` with optional argument `k=1` gives the same result as `lag()` and `lag.zoo()` with `k=-1`.

```
## Feb 1993 -0.0625
## Mar 1993  0.0476
## Apr 1993  0.0182
## May 1993  0.2411
## Jun 1993  0.0216
## Jul 1993 -0.0282
```

To compute continuously compounded returns from simple returns, use:

```
sbuxMonthlyReturnsC = log(1 + sbuxMonthlyReturns)
head(sbxuMonthlyReturnsC, n = 3)

##           SBUX
## Jan 1993    NA
## Feb 1993 -0.0645
## Mar 1993  0.0465
```

Or, equivalently, to compute continuously compounded returns directly from prices, use:

```
sbuxMonthlyReturnsC = diff(log(sbxuMonthlyPrices))
head(sbxuMonthlyReturnsC, n = 3)

##           SBUX
## Jan 1993    NA
## Feb 1993 -0.0645
## Mar 1993  0.0465
```

1.3.2 R packages for return calculations

There are several R packages that contain functions for creating, manipulating and plotting returns. An up-to-date list of packages relevant for finance is given in the finance task view on the comprehensive R archive network (CRAN). This section briefly describes some of the functions in the **PerformanceAnalytics** and **quantmod** packages.

PerformanceAnalytics

The **PerformanceAnalytics** package, written by Brian Peterson and Peter Carl, contains functions for performance and risk analysis of financial portfolios. Table 1.1 summarizes the functions in the package for performing return calculations and for plotting financial data. These functions work best with the financial data being represented as “**xts**” or “**zoo**” objects.

Simple and continuously compounded returns can be computed using the **PerformanceAnalytics** function **Return.calculate()**:

Function	Description
<code>Return.calculate</code>	Calculate returns from prices
<code>Drawdowns</code>	Find the drawdowns and drawdown levels
<code>maxDrawdown</code>	Calculate the maximum drawdown from peak
<code>Return.annualized</code>	Calculate an annualized return
<code>Return.cumulative</code>	Calculate a compounded cumulative return
<code>Return.excess</code>	Calculate returns in excess of a risk free rate
<code>Return.Geltner</code>	Calculate Geltner liquidity-adjusted return
<code>Return.read</code>	Read returns data with different date formats
<code>chart.CumReturns</code>	Plot cumulative returns over time
<code>chart.Drawdown</code>	Plot drawdowns over time
<code>chart.RelativePerformance</code>	Plot relative performance among multiple assets
<code>chart.TimeSeries</code>	Plot time series
<code>charts.PerformanceSummary</code>	Combination of performance charts.

Table 1.1 PerformanceAnalytics return calculation and plotting functions

```
library(PerformanceAnalytics)
# Simple returns
sbuxMonthlyReturns = Return.calculate(sbuxMonthlyPrices)
head(sbuxMonthlyReturns, n = 3)

##          SBUX
## Jan 1993    NA
## Feb 1993 -0.0625
## Mar 1993  0.0476

# CC returns
sbuxMonthlyReturnsC = Return.calculate(sbuxMonthlyPrices, method = "log")
head(sbuxMonthlyReturnsC, n = 3)

##          SBUX
## Jan 1993    NA
## Feb 1993 -0.0645
## Mar 1993  0.0465
```

The **PerformanceAnalytics** package contains several useful charting functions specially designed for financial time series. Fancy time series plots with event labels and shading can be created with `chart.TimeSeries()`:

```
shading.dates = c("1998-01::2000:10")
label.dates = c("Jan 1998", "Oct 2000")
label.values = c("Start of Boom", "End of Boom")
chart.TimeSeries(msftMonthlyPrices["1993::2003"], main = "The rise and fall of Microsoft stock",
                 lwd = 2, col = "blue", ylab = "Price", date.format = "%b %Y", period.areas = shading.dates,
```

```
period.color = "lightblue", event.lines = label.dates, event.labels = label.values,
event.color = "red")
```

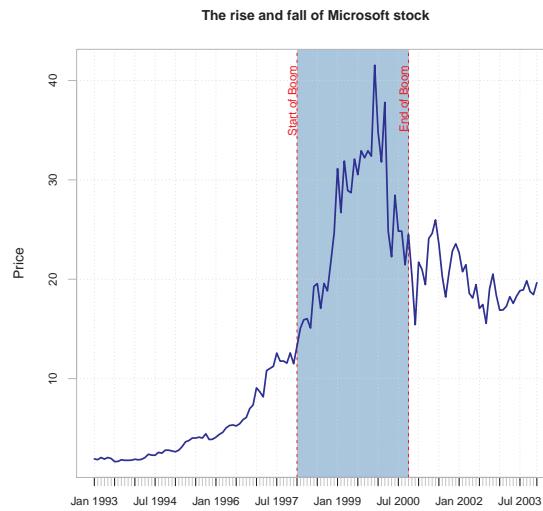


Fig. 1.5 Fancy time series chart created with `chart.TimeSeries()`.

Two or more investments can be compared by showing the growth of \$1 invested over time (i.e., the cumulative multi-period simple return) using the function `chart.CumReturns()`:

```
sbuxMsftReturns = Return.calculate(msftSbuxMonthlyPrices)
chart.CumReturns(sbuxMsftReturns, main = "Growth of $1", wealth.index = TRUE, lwd = 2,
legend.loc = "topleft")
```

Figure 1.3.2 clearly shows that Starbucks has outperformed Microsoft from 1993 through 2014. The plot showing the growth of a \$1 is sometimes called the *equity curve*.

More examples of using the **PerformanceAnalytics** functions are given in the end-of-chapter exercises.

quantmod

The **quantmod** package, written by Jeff Ryan, is designed to assist the quantitative trader in the development, testing, and deployment of statistically based trading models. See www.quantmod.com for more information about the **quantmod** package. Table 1.2 summarizes the functions in the package for retrieving data from the internet, performing return calculations, and plotting financial data.

Several functions in **quantmod** (i.e., those starting with `get`) automatically download specified data from the internet and import this data into R objects (typically “`xts`” objects). For example, to download from `finance.yahoo.com` all of the available daily data on Ya-

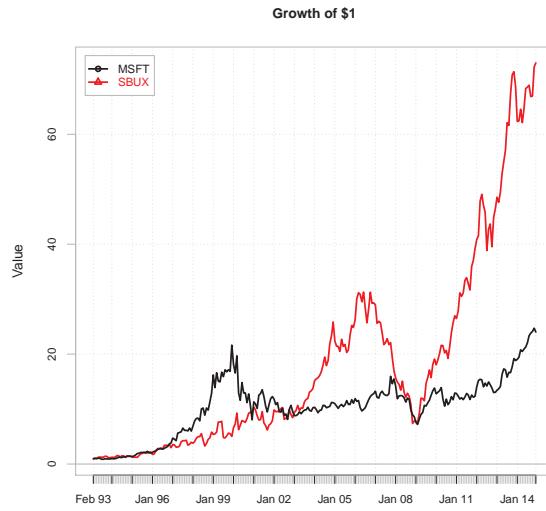


Fig. 1.6 Growth of \$1 invested in Microsoft and Starbucks stock.

Function	Description
<code>chartSeries</code>	Create financial charts
<code>getDividends</code>	Download dividend data from Yahoo!
<code>getFX</code>	Download exchange rates from Oanda
<code>getFinancials</code>	Download financial statements from google
<code>getMetals</code>	Download metals prices from Oanda
<code>getQuote</code>	Download quotes from various sources
<code>getSymbols</code>	Download data from various sources
<code>periodReturn</code>	Calculate returns from prices

Table 1.2 quantmod return calculation and plotting functions

hoo! stock (ticker symbol YHOO) and create the "xts" object YHOO use the `getSymbols()` function:

```
library(quantmod)
getSymbols("YHOO")

## [1] "YHOO"

class(YHOO)

## [1] "xts" "zoo"

colnames(YHOO)

## [1] "YHOO.Open"      "YHOO.High"       "YHOO.Low"        "YHOO.Close"
## [5] "YHOO.Volume"    "YHOO.Adjusted"
```

```
start(YHOO)

## [1] "2007-01-03"

end(YHOO)

## [1] "2016-10-20"
```

The “`xts`” object `YHOO` has six columns containing the daily open price, high price for the day, low price for the day, close price for the day, volume for the day, and (dividend and split) adjusted closing price for the day. To create the daily price-volume chart of the `YHOO` data shown in Figure 1.3.2, use the `chartSeries()` function:

```
chartSeries(YHOO, theme = chartTheme("white"))
```



Fig. 1.7 Daily price-volume plot for Yahoo! stock created with the `quantmod` function `chartSeries()`.

More examples of using the `quantmod` functions are given in the end-of-chapter exercises.

1.4 Further reading

This chapter describes basic asset return calculations with an emphasis on equity calculations. Similar material is covered in Chapter 2 of Ruppert and Matteson (2015). Comprehensive coverage of general return calculations is given in Bacon (2008) and Christopherson, Cariño and Ferson (2009).

1.5 Appendix: Properties of exponentials and logarithms

The computation of continuously compounded returns requires the use of natural logarithms. The natural logarithm function, $\ln(\cdot)$, is the inverse of the exponential function, $e^{(\cdot)} = \exp(\cdot)$, where $e^1 = 2.718$. That is, $\ln(x)$ is defined such that $x = \ln(e^x)$. Figure 1.5 plots e^x and $\ln(x)$. Notice that e^x is always positive and increasing in x . $\ln(x)$ is monotonically increasing in x and is only defined for $x > 0$. Also note that $\ln(1) = 0$ and $\ln(0) = -\infty$.

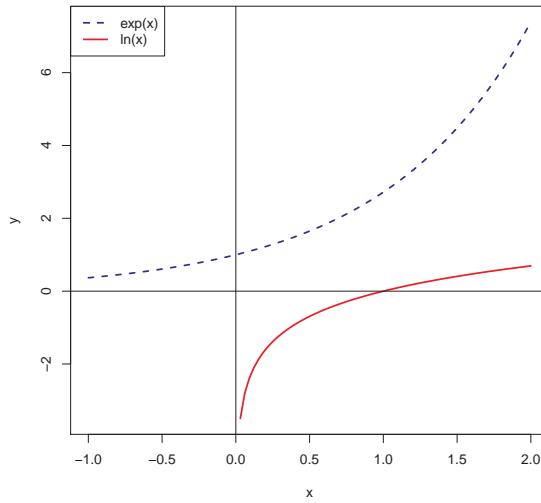


Fig. 1.8 Exponential and natural logarithm functions.

The exponential and natural logarithm functions have the following properties

1. $\ln(x \cdot y) = \ln(x) + \ln(y)$, $x, y > 0$
2. $\ln(x/y) = \ln(x) - \ln(y)$, $x, y > 0$
3. $\ln(x^y) = y \ln(x)$, $x > 0$
4. $\frac{d \ln(x)}{dx} = \frac{1}{x}$, $x > 0$
5. $\frac{d}{dx} \ln(f(x)) = \frac{1}{f(x)} \frac{d}{dx} f(x)$ (chain-rule)
6. $e^x e^y = e^{x+y}$
7. $e^x e^{-y} = e^{x-y}$
8. $(e^x)^y = e^{xy}$
9. $e^{\ln(x)} = x$
10. $\frac{d}{dx} e^x = e^x$
11. $\frac{d}{dx} e^{f(x)} = e^{f(x)} \frac{d}{dx} f(x)$ (chain-rule)

1.6 Exercises

Exercise 1.1. Consider the following (actual) monthly adjusted closing price data for Starbucks stock over the period December 2004 through December 2005

End of Month Price Data for Starbucks Stock	
December, 2004	\$31.18
January, 2005	\$27.00
February, 2005	\$25.91
March, 2005	\$25.83
April, 2005	\$24.76
May, 2005	\$27.40
June, 2005	\$25.83
July, 2005	\$26.27
August, 2005	\$24.51
September, 2005	\$25.05
October, 2005	\$28.28
November, 2005	\$30.45
December, 2005	\$30.51

1. Using the data in the table, what is the simple monthly return between the end of December 2004 and the end of January 2005? If you invested \$10,000 in Starbucks at the end of December 2004, how much would the investment be worth at the end of January 2005?
2. Using the data in the table, what is the continuously compounded monthly return between December 2004 and January 2005? Convert this continuously compounded return to a simple return (you should get the same answer as in part 1).
3. Assuming that the simple monthly return you computed in part (1) is the same for 12 months, what is the annual return with monthly compounding?
4. Assuming that the continuously compounded monthly return you computed in part (2) is the same for 12 months, what is the continuously compounded annual return?
5. Using the data in the table, compute the actual simple annual return between December 2004 and December 2005. If you invested \$10,000 in Starbucks at the end of December 2004, how much would the investment be worth at the end of December 2005? Compare with your result in part (3).
6. Using the data in the table, compute the actual annual continuously compounded return between December 2004 and December 2005. Compare with your result in part (4). Convert this continuously compounded return to a simple return (you should get the same answer as in part 5).

Exercise 1.2. Consider a one month investment in two Northwest stocks: Amazon and Costco. Suppose you buy Amazon and Costco at the end of September at $P_{A,t-1} = \$38.23$, $P_{C,t-1} = \$41.11$ and then sell at the end of the October for $P_{A,t} = \$41.29$ and $P_{C,t} = \$41.74$. (Note: these are actual closing prices for 2004 taken from Yahoo!)

1. What are the simple monthly returns for the two stocks?
2. What are the continuously compounded returns for the two stocks?
3. Suppose Costco paid a \$0.10 per share cash dividend at the end of October. What is the monthly simple total return on Costco? What is the monthly dividend yield?
4. Suppose the monthly returns on Amazon and Costco from question (1) above are the same every month for 1 year. Compute the simple annual returns as well as the continuously compounded annual returns for the two stocks.
5. At the end of September 2004, suppose you have \$10,000 to invest in Amazon and Costco over the next month. If you invest \$8000 in Amazon and \$2000 in Costco, what are your portfolio shares, x_A and x_C ?
6. Continuing with part 5, compute the monthly simple return and the monthly continuously compounded return on the portfolio. Assume that Costco does not pay a dividend.

Exercise 1.3. Consider a 60-month (5 year) investment in two assets: the Vanguard S&P 500 index (VFINX) and Apple stock (AAPL). Suppose you buy one share of the S&P 500 fund and one share of Apple stock at the end of January, 2010 for $P_{VFINF,t-60} = \$89.91$, $P_{AAPL,t-60} = \$25.88$, and then sell these shares at the end of January, 2015 for $P_{VFINX,t} = \$184.2$, $P_{AAPL,t} = \$116.7$. (Note: these are actual adjusted closing prices taken from Yahoo!). In this question, you will see how much money you could have made if you invested in these assets right after the financial crisis.

1. What are the simple 60-month (5-year) returns for the two investments?
2. What are the continuously compounded 60-month (5-year) returns for the two investments?
3. Suppose you invested \$1,000 in each asset at the end of January, 2010. How much would each investment be worth at the end of January, 2015?
4. What is the compound annual return on the two 5 year investments?
5. At the end of January, 2010, suppose you have \$1,000 to invest in VFINX and AAPL over the next 60 months (5 years). Suppose you purchase \$400 worth of VFINX and the remainder in AAPL. What are the portfolio weights in the two assets? Using the results from parts 1. and 2., compute the 5-year simple and continuously compounded portfolio returns.

Exercise 1.4. Consider an investment in a foreign stock (e.g., a stock trading on the London stock exchange) by a U.S. national (domestic investor). The domestic investor takes U.S. dollars, converts them to the foreign currency (e.g. British Pound) via the exchange rate (price of foreign currency in U.S. dollars) and then purchases the foreign stock using the foreign currency. When the stock is sold, the proceeds in the foreign currency must then be converted back to the domestic currency. To be more precise, consider the information in the table below:

Time	Cost of 1 Pound	Value of UK Shares	Value in U.S. \$
0	\$1.50	£40	$1.5 \times 40 = 60$
1	\$1.30	£45	$1.3 \times 45 = 58.5$

1. Compute the simple rate of return, R_e , from the prices of foreign currency. This is the return to the domestic investor of investing in the foreign currency.
2. Compute the simple rate of return, R_{UK} , from the UK stock prices.
3. Compute the simple rate of return, R_{US} , from the prices in US dollars.
4. What is the mathematical relationship between R_{US} , R_{UK} and R_e ?

Exercise 1.5. Add R examples here.

References

1. Bacon, C. (2008). *Practical Portfolio Performance Measurement and Attribution, Second Edition*. John Wiley & Sons, Chichester, England.
2. Christopherson, J.A., Cariño, D.R., and Ferson, W.E. (2009). Portfolio Performance Measurement and Benchmarking. McGraw-Hill.
3. Ruppert, D., and Matteson, D.S. (2015). *Statistics and Data Analysis for Financial Engineering with R examples*. Springer, New York.

Chapter 2

Review of Random Variables

Updated: July 14, 2016

Copyright © Eric Zivot 2015

This chapter reviews the probability concepts that are necessary for the modeling and statistical analysis of financial data presented in this book. This material is typically covered in an introductory probability and statistics course using calculus. In the course of the review, many examples related to finance will be presented and some important risk concepts, such as value-at-risk, will be introduced. The examples will also show how to use R for probability calculations and working with probability distributions.

The chapter is outlined as follows. Section 2.1 reviews univariate random variables and properties of univariate probability distributions. Important univariate distributions, such as the normal distribution, that are used throughout the book for modeling financial data, are described in detail. Section 2.2 covers bivariate probability distributions. The concepts of dependence and independence between two random variables are discussed. The measures of linear dependence between two random variables, covariance and correlation, are defined and the bivariate normal distribution is introduced. Properties of linear combinations of two random variables are given and illustrated using an example to describe the return and risk properties of a portfolio of two assets. Multivariate distributions are briefly discussed in section . Here, properties of linear combinations of N random variables are summarized.

The only R package used in this chapter is **mvtnorm**. Make sure this package is downloaded and installed prior to replicating the R examples in this chapter.

2.1 Random Variables

We start with the basic definition of a random variable:

Definition 2.1. A Random variable X is a variable that can take on a given set of values, called the sample space and denoted S_X , where the likelihood of the values in S_X is determined by X 's probability distribution function (pdf).

Example 2.1. Future price of Microsoft stock.

Consider the price of Microsoft stock next month. Since the price of Microsoft stock next month is not known with certainty today, we can consider it a random variable. The price next month must be positive and realistically it can't get too large. Therefore the sample space is the set of positive real numbers bounded above by some large number: $S_p = \{P : P \in [0, M], M > 0\}$. It is an open question as to what is the best characterization of the probability distribution of stock prices. The log-normal distribution is one possibility.¹

■

Example 2.2. Return on Microsoft stock.

Consider a one-month investment in Microsoft stock. That is, we buy one share of Microsoft stock at the end of month $t - 1$ (e.g., end of February) and plan to sell it at the end of month t (e.g., end of March). The return over month t on this investment, $R_t = (P_t - P_{t-1})/P_{t-1}$, is a random variable because we do not know what the price will be at the end of the month. In contrast to prices, returns can be positive or negative and are bounded from below by -100%. We can express the sample space as $S_{R_t} = \{R_t : R_t \in [-1, M], M > 0\}$. The normal distribution is often a good approximation to the distribution of simple monthly returns, and is a better approximation to the distribution of continuously compounded monthly returns.

■

Example 2.3. Up-down indicator variable.

As a final example, consider a variable X defined to be equal to one if the monthly price change on Microsoft stock, $P_t - P_{t-1}$, is positive, and is equal to zero if the price change is zero or negative. Here, the sample space is the set $S_X = \{0, 1\}$. If it is equally likely that the monthly price change is positive or negative (including zero) then the probability that $X = 1$ or $X = 0$ is 0.5. This is an example of a bernoulli random variable.

■

The next sub-sections define discrete and continuous random variables.

2.1.1 Discrete random variables

Consider a random variable generically denoted X and its set of possible values or sample space denoted S_X .

Definition 2.2. A discrete random variable X is one that can take on a finite number of n different values $S_X = \{x_1, x_2, \dots, x_n\}$ or, at most, a countably infinite number of different values $S_X = \{x_1, x_2, \dots\}$.

Definition 2.3. The pdf of a discrete random variable, denoted $p(x)$, is a function such that $p(x) = \Pr(X = x)$. The pdf must satisfy (i) $p(x) \geq 0$ for all $x \in S_X$; (ii) $p(x) = 0$ for all $x \notin S_X$; and (iii) $\sum_{x \in S_X} p(x) = 1$.

¹ If P is a positive random variable such that $\ln P$ is normally distributed then P has a log-normal distribution.

Example 2.4. Annual return on Microsoft stock.

State of Economy	Sample Space	$\Pr(X = x)$
Depression	-0.30	0.05
Recession	0.0	0.20
Normal	0.10	0.50
Mild Boom	0.20	0.20
Major Boom	0.50	0.05

Table 2.1 Probability distribution for the annual return on Microsoft.

Let X denote the annual return on Microsoft stock over the next year. We might hypothesize that the annual return will be influenced by the general state of the economy. Consider five possible states of the economy: depression, recession, normal, mild boom and major boom. A stock analyst might forecast different values of the return for each possible state. Hence, X is a discrete random variable that can take on five different values. Table 2.1 describes such a probability distribution of the return and a graphical representation of the probability distribution is presented in Figure 2.1.1 created with

```
r.msft = c(-0.3, 0, 0.1, 0.2, 0.5)
prob.vals = c(0.05, 0.2, 0.5, 0.2, 0.05)
plot(r.msft, prob.vals, type = "h", lwd = 4, xlab = "annual return", ylab = "probability",
      xaxt = "n")
points(r.msft, prob.vals, pch = 16, cex = 2, col = "blue")
axis(1, at = r.msft)
```

The Bernoulli Distribution

Let $X = 1$ if the price next month of Microsoft stock goes up and $X = 0$ if the price goes down (assuming it cannot stay the same). Then X is clearly a discrete random variable with sample space $S_X = \{0, 1\}$. If the probability of the stock price going up or down is the same then $p(0) = p(1) = 1/2$ and $p(0) + p(1) = 1$.

The probability distribution described above can be given an exact mathematical representation known as the *Bernoulli* distribution. Consider two mutually exclusive events generically called “success” and “failure”. For example, a success could be a stock price going up or a coin landing heads and a failure could be a stock price going down or a coin landing tails. The process creating the success or failure is called a *Bernoulli trial*. In general, let $X = 1$ if success occurs and let $X = 0$ if failure occurs. Let $\Pr(X = 1) = \pi$, where $0 < \pi < 1$, denote the probability of success. Then $\Pr(X = 0) = 1 - \pi$ is the probability of failure. A mathematical model describing this distribution is:

$$p(x) = \Pr(X = x) = \pi^x(1 - \pi)^{1-x}, \quad x = 0, 1. \quad (2.1)$$

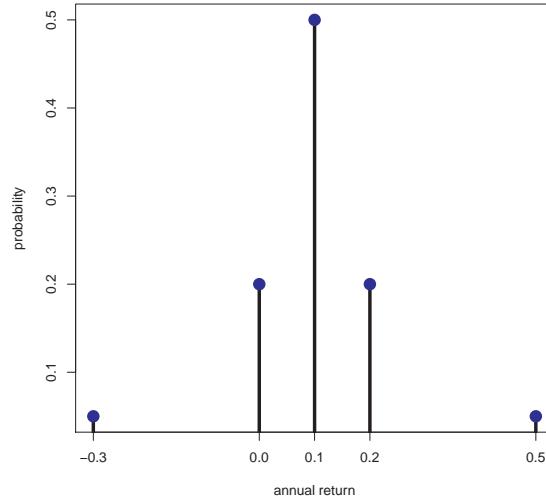


Fig. 2.1 Discrete distribution for Microsoft stock.

When $x = 0$, $p(0) = \pi^0(1 - \pi)^{1-0} = 1 - \pi$ and when $x = 1$, $p(1) = \pi^1(1 - \pi)^{1-1} = \pi$.

The Binomial Distribution

Consider a sequence of independent Bernoulli trials with success probability π generating a sequence of 0-1 variables indicating failures and successes. A *binomial* random variable X counts the number of successes in n Bernoulli trials, and is denoted $X \sim B(n, \pi)$. The sample space is $S_X = \{0, 1, \dots, n\}$ and:

$$\Pr(X = k) = \binom{n}{k} \pi^k (1 - \pi)^{n-k}.$$

The term $\binom{n}{k}$ is the binomial coefficient, and counts the number of ways k objects can be chosen from n distinct objects. It is defined by:

$$\binom{n}{k} = \frac{n!}{(n - k)!k!}$$

where $n!$ is the factorial of n , or $n(n - 1) \cdots 2 \cdot 1$.

2.1.2 Continuous random variables

Definition 2.4. A continuous random variable X is one that can take on any real value. That is, $S_X = \{x : x \in \mathbb{R}\}$.

Definition 2.5. The probability density function (pdf) of a continuous random variable X is a nonnegative function f , defined on the real line, such that for any interval A :

$$\Pr(X \in A) = \int_A f(x)dx.$$

That is, $\Pr(X \in A)$ is the “area under the probability curve over the interval A ”. The pdf $f(x)$ must satisfy (i) $f(x) \geq 0$; and (ii) $\int_{-\infty}^{\infty} f(x)dx = 1$.

A typical “bell-shaped” pdf is displayed in Figure 2.1.2 and the shaded area under the curve between -2 and 1 represents $\Pr(-2 \leq X \leq 1)$. For a continuous random variable, $f(x) \neq \Pr(X = x)$ but rather gives the height of the probability curve at x . In fact, $\Pr(X = x) = 0$ for all values of x . That is, probabilities are not defined over single points. They are only defined over intervals. As a result, for a continuous random variable X we have:

$$\Pr(a \leq X \leq b) = \Pr(a < X \leq b) = \Pr(a < X < b) = \Pr(a \leq X < b).$$

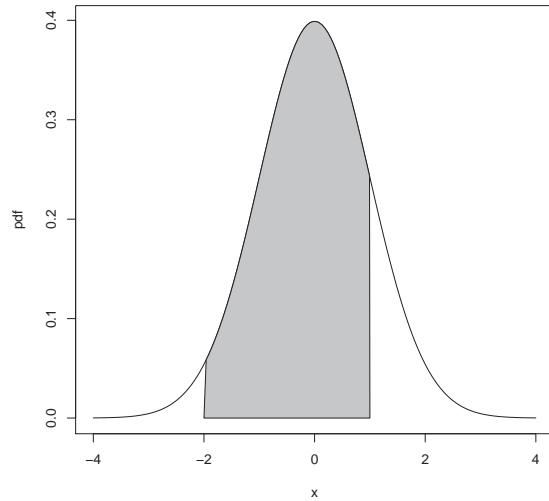


Fig. 2.2 $\Pr(-2 \leq X \leq 1)$ is represented by the area under the probability curve.

The Uniform Distribution on an interval

Let X denote the annual return on Microsoft stock and let a and b be two real numbers such that $a < b$. Suppose that the annual return on Microsoft stock can take on any value between a and b . That is, the sample space is restricted to the interval $S_X = \{x \in \mathcal{R} : a \leq x \leq b\}$. Further suppose that the probability that X will belong to any subinterval of S_X is proportional to the length of the interval. In this case, we say that X is *uniformly distributed* on the interval $[a, b]$. The pdf of X has a very simple mathematical form:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{otherwise,} \end{cases}$$

and is presented graphically as a rectangle. It is easy to see that the area under the curve over the interval $[a, b]$ (area of rectangle) integrates to one:

$$\int_a^b \frac{1}{b-a} dx = \frac{1}{b-a} \int_a^b dx = \frac{1}{b-a} [x]_a^b = \frac{1}{b-a} [b-a] = 1.$$

Example 2.5. Uniform distribution on $[-1, 1]$.

Let $a = -1$ and $b = 1$, so that $b - a = 2$. Consider computing the probability that X will be between -0.5 and 0.5. We solve,

$$\Pr(-0.5 \leq X \leq 0.5) = \int_{-0.5}^{0.5} \frac{1}{2} dx = \frac{1}{2} [x]_{-0.5}^{0.5} = \frac{1}{2} [0.5 - (-0.5)] = \frac{1}{2}.$$

Next, consider computing the probability that the return will fall in the interval $[0, \delta]$ where δ is some small number less than $b = 1$:

$$\Pr(0 \leq X \leq \delta) = \frac{1}{2} \int_0^\delta dx = \frac{1}{2} [\delta]_0^\delta = \frac{1}{2} \delta.$$

As $\delta \rightarrow 0$, $\Pr(0 \leq X \leq \delta) \rightarrow \Pr(X = 0)$.

Using the above result we see that

$$\lim_{\delta \rightarrow 0} \Pr(0 \leq X \leq \delta) = \Pr(X = 0) = \lim_{\delta \rightarrow 0} \frac{1}{2} \delta = 0.$$

Hence, for continuous random variables probabilities are defined on intervals but not at distinct points. ■

The Standard Normal Distribution

The normal or Gaussian distribution is perhaps the most famous and most useful continuous distribution in all of statistics. The shape of the normal distribution is the familiar “bell curve”. As we shall see, it can be used to describe the probabilistic behavior of stock returns although other distributions may be more appropriate.

If a random variable X follows a *standard normal distribution* then we often write $X \sim N(0, 1)$ as short-hand notation. This distribution is centered at zero and has inflection points at ± 1 .² The pdf of a normal random variable is given by:

$$f(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2}, \quad -\infty \leq x \leq \infty. \quad (2.2)$$

² An inflection point is a point where a curve goes from concave up to concave down, or vice versa.

It can be shown via the change of variables formula in calculus that the area under the standard normal curve is one:

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2} dx = 1.$$

The standard normal distribution is illustrated in Figure 2.1.2. Notice that the distribution is *symmetric* about zero; i.e., the distribution has exactly the same form to the left and right of zero. Because the standard normal pdf formula is used so often it is given its own special symbol $\phi(x)$.

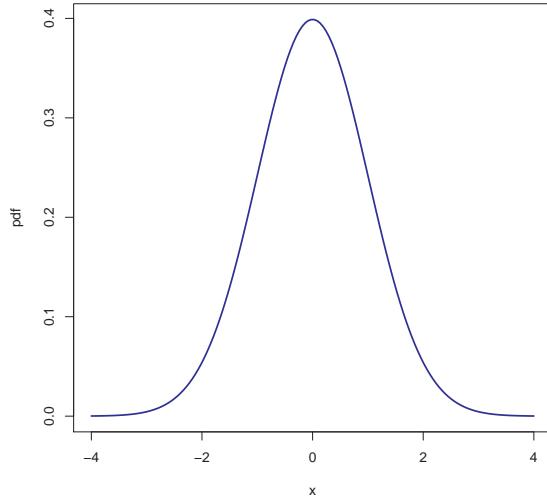


Fig. 2.3 Standard normal density.

The normal distribution has the annoying feature that the area under the normal curve cannot be evaluated analytically. That is:

$$\Pr(a \leq X \leq b) = \int_a^b \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2} dx,$$

does not have a closed form solution. The above integral must be computed by numerical approximation. Areas under the normal curve, in one form or another, are given in tables in almost every introductory statistics book and standard statistical software can be used to find these areas. Some useful approximate results are:

$$\Pr(-1 \leq X \leq 1) \approx 0.67,$$

$$\Pr(-2 \leq X \leq 2) \approx 0.95,$$

$$\Pr(-3 \leq X \leq 3) \approx 0.99.$$

2.1.3 The Cumulative Distribution Function

Definition 2.6. The cumulative distribution function (cdf) of a random variable X (discrete or continuous), denoted F_X , is the probability that $X \leq x$:

$$F_X(x) = \Pr(X \leq x), \quad -\infty \leq x \leq \infty.$$

The cdf has the following properties:

- (i) If $x_1 < x_2$ then $F_X(x_1) \leq F_X(x_2)$
- (ii) $F_X(-\infty) = 0$ and $F_X(\infty) = 1$
- (iii) $\Pr(X > x) = 1 - F_X(x)$
- (iv) $\Pr(x_1 < X \leq x_2) = F_X(x_2) - F_X(x_1)$
- (v) $F'_X(x) = \frac{d}{dx} F_X(x) = f(x)$ if X is a continuous random variable and $F_X(x)$ is continuous and differentiable.

Example 2.6. $F_X(x)$ for a discrete random variable.

The cdf for the discrete distribution of Microsoft from Table 2.1 is given by:

$$F_X(x) = \begin{cases} 0, & x < -0.3 \\ 0.05, & -0.3 \leq x < 0 \\ 0.25, & 0 \leq x < 0.1 \\ 0.75, & 0.1 \leq x < 0.2 \\ 0.95, & 0.2 \leq x < 0.5 \\ 1, & x > 0.5 \end{cases}$$

and is illustrated in Figure 2.1.3. ■

Example 2.7. $F_X(x)$ for a uniform random variable.

The cdf for the uniform distribution over $[a, b]$ can be determined analytically:

$$\begin{aligned} F_X(x) &= \Pr(X < x) = \int_{-\infty}^x f(t) dt \\ &= \frac{1}{b-a} \int_a^x dt = \frac{1}{b-a} [t]_a^x = \frac{x-a}{b-a}. \end{aligned}$$

We can determine the pdf of X directly from the cdf via,

$$f(x) = F'_X(x) = \frac{d}{dx} F_X(x) = \frac{1}{b-a}.$$
■

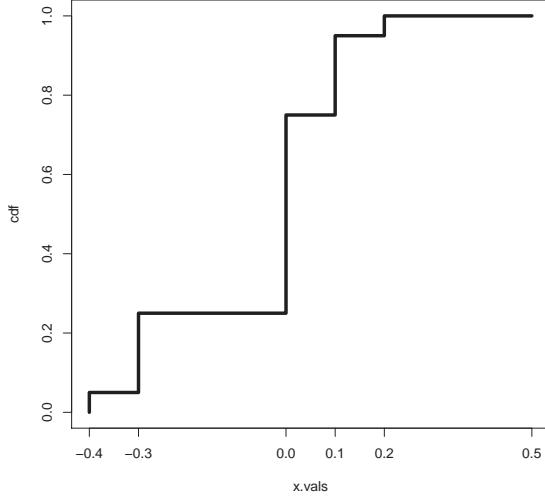


Fig. 2.4 CDF of Discrete Distribution for Microsoft Stock Return.

Example 2.8. $F_X(x)$ for a standard normal random variable.

The cdf of standard normal random variable X is used so often in statistics that it is given its own special symbol:

$$\Phi(x) = F_X(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz. \quad (2.3)$$

The cdf $\Phi(x)$, however, does not have an analytic representation like the cdf of the uniform distribution and so the integral in (2.3) must be approximated using numerical techniques. A graphical representation of $\Phi(x)$ is given in Figure 2.1.3. Because the standard normal pdf, $\phi(x)$, is symmetric about zero and bell-shaped the standard normal CDF, $\Phi(x)$, has an “S” shape where the middle of the “S” is at zero.

■

2.1.4 Quantiles of the distribution of a random variable

Consider a random variable X with continuous cdf $F_X(x)$. For $0 \leq \alpha \leq 1$, the $100 \cdot \alpha\%$ quantile of the distribution for X is the value q_α that satisfies:

$$F_X(q_\alpha) = \Pr(X \leq q_\alpha) = \alpha.$$

For example, the 5% quantile of X , $q_{0.05}$, satisfies,

$$F_X(q_{0.05}) = \Pr(X \leq q_{0.05}) = 0.05.$$

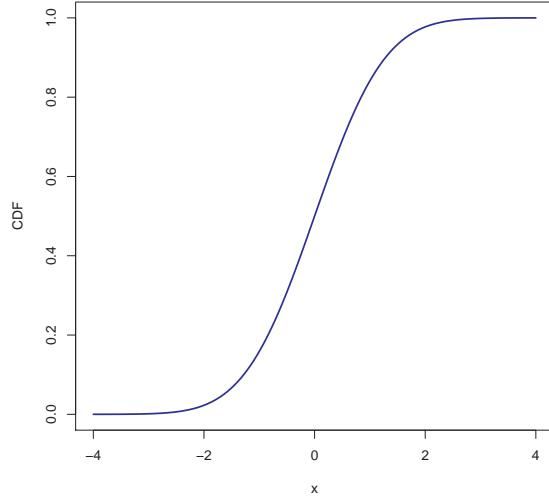


Fig. 2.5 Standard normal cdf $\Phi(x)$.

The *median* of the distribution is 50% quantile. That is, the median, $q_{0.5}$, satisfies,

$$F_X(q_{0.5}) = \Pr(X \leq q_{0.5}) = 0.5.$$

If F_X is invertible³ then q_α may be determined analytically as:

$$q_\alpha = F_X^{-1}(\alpha) \quad (2.4)$$

where F_X^{-1} denotes the inverse function of F_X . Hence, the 5% quantile and the median may be determined from

$$q_{0.05} = F_X^{-1}(0.05), q_{0.5} = F_X^{-1}(0.5).$$

In inverse cdf F_X^{-1} is sometimes called the *quantile function*.

Example 2.9. Quantiles from a uniform distribution.

Let $X \sim U[a, b]$ where $b > a$. Recall, the cdf of X is given by:

$$F_X(x) = \frac{x - a}{b - a}, \quad a \leq x \leq b,$$

which is continuous and strictly increasing. Given $\alpha \in [0, 1]$ such that $F_X(x) = \alpha$, solving for x gives the inverse cdf:

$$x = F_X^{-1}(\alpha) = \alpha(b - a) + a. \quad (2.5)$$

Using (2.5), the 5% quantile and median, for example, are given by:

³ The inverse function of F_X , denoted F_X^{-1} , has the property that $F_X^{-1}(F_X(x)) = x$. F_X^{-1} will exist if F_X is strictly increasing and is continuous.

$$\begin{aligned}q_{0.05} &= F_X^{-1}(.05) = .05(b-a) + a = .05b + .95a, \\q_{0.5} &= F_X^{-1}(.5) = .5(b-a) + a = .5(a+b).\end{aligned}$$

If $a = 0$ and $b = 1$, then $q_{0.05} = 0.05$ and $q_{0.5} = 0.5$. ■

Example 2.10. Quantiles from a standard normal distribution.

Let $X \sim N(0, 1)$. The quantiles of the standard normal distribution are determined by solving

$$q_\alpha = \Phi^{-1}(\alpha), \quad (2.6)$$

where Φ^{-1} denotes the inverse of the cdf Φ . This inverse function must be approximated numerically and is available in most spreadsheets and statistical software. Using the numerical approximation to the inverse function, the 1%, 2.5%, 5%, 10% quantiles and median are given by:

$$\begin{aligned}q_{0.01} &= \Phi^{-1}(.01) = -2.33, \quad q_{0.025} = \Phi^{-1}(.025) = -1.96, \\q_{0.05} &= \Phi^{-1}(.05) = -1.645, \quad q_{0.10} = \Phi^{-1}(.10) = -1.28, \\q_{0.5} &= \Phi^{-1}(.5) = 0.\end{aligned}$$

Often, the standard normal quantile is denoted z_α . ■

2.1.5 R functions for discrete and continuous distributions

R has built-in functions for a number of discrete and continuous distributions that are commonly used in probability modeling and statistical analysis. These are summarized in Table 2.2. For each distribution, there are four functions starting with `d`, `p`, `q` and `r` that compute density (pdf) values, cumulative probabilities (cdf), quantiles (inverse cdf) and random draws, respectively. Consider, for example, the functions associated with the normal distribution. The functions `dnorm()`, `pnorm()` and `qnorm()` evaluate the standard normal density (2.2), the cdf (2.3), and the inverse cdf or quantile function (2.6), respectively, with the default values `mean=0` and `sd = 1`. The function `rnorm()` returns a specified number of simulated values from the normal distribution.

Finding areas under the standard normal curve

Let X denote a standard normal random variable. Given that the total area under the normal curve is one and the distribution is symmetric about zero the following results hold:

- $\Pr(X \leq z) = 1 - \Pr(X \geq z)$ and $\Pr(X \geq z) = 1 - \Pr(X \leq z)$
- $\Pr(X \geq z) = \Pr(X \leq -z)$

Distribution	Function (root)	Parameters	Defaults
beta	beta	shape1, shape2	-, -
binomial	binom	size, prob	-, -
Cauchy	cauchy	location, scale	0, 1
chi-squared	chisq	df, ncp	-, 1
F	f	df1, df2	-, -
gamma	gamma	shape, rate, scale	_, 1, 1/rate
geometric	geom	prob	-
hyper-geometric	hyper	m, n, k	-, -, -
log-normal	lnorm	meanlog, sdlog	0, 1
logistic	logis	location, scale	0, 1
negative binomial	nbinom	size, prob, mu	-, -, -
normal	norm	mean, sd	0, 1
Poisson	pois	Lambda	1
Student's t	t	df, ncp	-, 1
uniform	unif	min, max	0, 1
Weibull	weibull	shape, scale	-, 1
Wilcoxon	wilcoxon	m, n	-, -

Table 2.2 Probability distributions in base R.

- $\Pr(X \geq 0) = \Pr(X \leq 0) = 0.5$

The following examples show how to do probability calculations with standard normal random variables.

Example 2.11. Finding areas under the normal curve using R.

First, consider finding $\Pr(X \geq 2)$. By the symmetry of the normal distribution, $\Pr(X \geq 2) = \Pr(X \leq -2) = \Phi(-2)$. In R use:

```
pnorm(-2)
## [1] 0.0228
```

Next, consider finding $\Pr(-1 \leq X \leq 2)$. Using the cdf, we compute $\Pr(-1 \leq X \leq 2) = \Pr(X \leq 2) - \Pr(X \leq -1) = \Phi(2) - \Phi(-1)$. In R use:

```
pnorm(2) - pnorm(-1)
## [1] 0.819
```

Finally, using R the exact values for $\Pr(-1 \leq X \leq 1)$, $\Pr(-2 \leq X \leq 2)$ and $\Pr(-3 \leq X \leq 3)$ are:

```
pnorm(1) - pnorm(-1)
## [1] 0.683

pnorm(2) - pnorm(-2)
## [1] 0.954

pnorm(3) - pnorm(-3)
## [1] 0.997
```



Plotting distributions

When working with a probability distribution, it is a good idea to make plots of the pdf or cdf to reveal important characteristics. The following examples illustrate plotting distributions using R.

Example 2.12. Plotting the standard normal curve.

The graphs of the standard normal pdf and cdf in Figures 2.1.2 and 2.1.3 were created using the following R code:

```
# plot pdf
x.vals = seq(-4, 4, length = 150)
plot(x.vals, dnorm(x.vals), type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "pdf")
# plot cdf
plot(x.vals, pnorm(x.vals), type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "CDF")
```



Example 2.13. Shading a region under the standard normal curve.

Figure 2.1.2 showing $\Pr(-2 \leq X \leq 1)$ as a red shaded area is created with the following code:

```
lb = -2
ub = 1
x.vals = seq(-4, 4, length = 150)
d.vals = dnorm(x.vals)
# plot normal density
plot(x.vals, d.vals, type = "l", xlab = "x", ylab = "pdf")
i = x.vals >= lb & x.vals <= ub
# add shaded region between -2 and 1
polygon(c(lb, x.vals[i], ub), c(0, d.vals[i], 0), col = "grey")
```



2.1.6 Shape characteristics of probability distributions

Very often we would like to know certain shape characteristics of a probability distribution. We might want to know where the distribution is centered, and how spread out the distribution is about the central value. We might want to know if the distribution is symmetric about the center or if the distribution has a long left or right tail. For stock returns we might want to know about the likelihood of observing extreme values for returns representing market crashes. This means that we would like to know about the amount of probability in the extreme tails of the distribution. In this section we discuss four important shape characteristics of a probability distribution:

1. expected value (mean): measures the center of mass of a distribution
2. variance and standard deviation: measures the spread about the mean
3. skewness: measures symmetry about the mean
4. kurtosis: measures "tail thickness"

Expected Value

The expected value of a random variable X , denoted $E[X]$ or μ_X , measures the center of mass of the pdf. For a discrete random variable X with sample space S_X , the expected value is defined as:

$$\mu_X = E[X] = \sum_{x \in S_X} x \cdot \Pr(X = x). \quad (2.7)$$

Equation (2.7) shows that $E[X]$ is a probability weighted average of the possible values of X .

Example 2.14. Expected value of discrete random variable.

Using the discrete distribution for the return on Microsoft stock in Table 2.1, the expected return is computed as:

$$\begin{aligned} E[X] &= (-0.3) \cdot (0.05) + (0.0) \cdot (0.20) + (0.1) \cdot (0.5) + (0.2) \cdot (0.2) + (0.5) \cdot (0.05) \\ &= 0.10. \end{aligned}$$

■

Example 2.15. Expected value of Bernoulli and binomial random variables.

Let X be a Bernoulli random variable with success probability π . Then,

$$E[X] = 0 \cdot (1 - \pi) + 1 \cdot \pi = \pi.$$

That is, the expected value of a Bernoulli random variable is its probability of success. Now, let $Y \sim B(n, \pi)$. It can be shown that:

$$E[Y] = \sum_{k=0}^n k \binom{n}{k} \pi^k (1-\pi)^{n-k} = n\pi. \quad (2.8)$$

■

For a continuous random variable X with pdf $f(x)$, the expected value is defined as:

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x \cdot f(x) dx. \quad (2.9)$$

Example 2.16. Expected value of a uniform random variable.

Suppose X has a uniform distribution over the interval $[a, b]$. Then,

$$\begin{aligned} E[X] &= \frac{1}{b-a} \int_a^b x dx = \frac{1}{b-a} \left[\frac{1}{2} x^2 \right]_a^b \\ &= \frac{1}{2(b-a)} [b^2 - a^2] \\ &= \frac{(b-a)(b+a)}{2(b-a)} = \frac{b+a}{2}. \end{aligned}$$

If $b = -1$ and $a = 1$, then $E[X] = 0$.

■

Example 2.17. Expected value of a standard normal random variable.

Let $X \sim N(0, 1)$. Then it can be shown that:

$$E[X] = \int_{-\infty}^{\infty} x \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx = 0.$$

Hence, the standard normal distribution is centered at zero.

■

Expectation of a function of a random variable

The other shape characteristics of the distribution of a random variable X are based on expectations of certain functions of X . Let $g(X)$ denote some function of the random variable X . If X is a discrete random variable with sample space S_X then:

$$E[g(X)] = \sum_{x \in S_X} g(x) \cdot \Pr(X = x), \quad (2.10)$$

and if X is a continuous random variable with pdf f then,

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) \cdot f(x) dx. \quad (2.11)$$

Variance and standard deviation

The variance of a random variable X , denoted $\text{var}(X)$ or σ_X^2 , measures the spread of the distribution about the mean using the function $g(X) = (X - \mu_X)^2$. If most values of X are close to μ_X then on average $(X - \mu_X)^2$ will be small. In contrast, if many values of X are far below and/or far above μ_X then on average $(X - \mu_X)^2$ will be large. Squaring the deviations about μ_X guarantees a positive value. The variance of X is defined as:

$$\sigma_X^2 = \text{var}(X) = E[(X - \mu_X)^2]. \quad (2.12)$$

Because σ_X^2 represents an average squared deviation, it is not in the same units as X . The standard deviation of X , denoted $\text{sd}(X)$ or σ_X , is the square root of the variance and is in the same units as X . For “bell-shaped” distributions, σ_X measures the typical size of a deviation from the mean value.

The computation of (2.12) can often be simplified by using the result:

$$\text{var}(X) = E[(X - \mu_X)^2] = E[X^2] - \mu_X^2 \quad (2.13)$$

Example 2.18. Variance and standard deviation for a discrete random variable.

Using the discrete distribution for the return on Microsoft stock in Table 2.1 and the result that $\mu_X = 0.1$, we have:

$$\begin{aligned} \text{var}(X) &= (-0.3 - 0.1)^2 \cdot (0.05) + (0.0 - 0.1)^2 \cdot (0.20) + (0.1 - 0.1)^2 \cdot (0.5) \\ &\quad + (0.2 - 0.1)^2 \cdot (0.2) + (0.5 - 0.1)^2 \cdot (0.05) \\ &= 0.020. \end{aligned}$$

Alternatively, we can compute $\text{var}(X)$ using (2.13):

$$\begin{aligned} E[X^2] - \mu_X^2 &= (-0.3)^2 \cdot (0.05) + (0.0)^2 \cdot (0.20) + (0.1)^2 \cdot (0.5) \\ &\quad + (0.2)^2 \cdot (0.2) + (0.5)^2 \cdot (0.05) - (0.1)^2 \\ &= 0.020. \end{aligned}$$

The standard deviation is $\text{sd}(X) = \sigma_X = \sqrt{0.020} = 0.141$. Given that the distribution is fairly bell-shaped we can say that typical values deviate from the mean value of 10% by about 14.1%. ■

Example 2.19. Variance and standard deviation of a Bernoulli and binomial random variables.

Let X be a Bernoulli random variable with success probability π . Given that $\mu_X = \pi$ it follows that:

$$\begin{aligned}
\text{var}(X) &= (0 - \pi)^2 \cdot (1 - \pi) + (1 - \pi)^2 \cdot \pi \\
&= \pi^2(1 - \pi) + (1 - \pi)^2\pi \\
&= \pi(1 - \pi)[\pi + (1 - \pi)] \\
&= \pi(1 - \pi), \\
\text{sd}(X) &= \sqrt{\pi(1 - \pi)}.
\end{aligned}$$

Now, let $Y \sim B(n, \pi)$: It can be shown that:

$$\text{var}(Y) = n\pi(1 - \pi)$$

and so $\text{sd}(Y) = \sqrt{n\pi(1 - \pi)}$. ■

Example 2.20. Variance and standard deviation of a uniform random variable.

Let $X \sim U[a, b]$. Using (2.13) and $\mu_X = \frac{a+b}{2}$, after some algebra, it can be shown that:

$$\text{var}(X) = E[X^2] - \mu_X^2 = \frac{1}{b-a} \int_a^b x^2 dx - \left(\frac{a+b}{2}\right)^2 = \frac{1}{12}(b-a)^2,$$

and $\text{sd}(X) = (b-a)/\sqrt{12}$. ■

Example 2.21. Variance and standard deviation of a standard normal random variable.

Let $X \sim N(0, 1)$. Here, $\mu_X = 0$ and it can be shown that:

$$\sigma_X^2 = \int_{-\infty}^{\infty} x^2 \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx = 1.$$

It follows that $\text{sd}(X) = 1$. ■

The General Normal Distribution

Recall, if X has a standard normal distribution then $E[X] = 0$, $\text{var}(X) = 1$. A general normal random variable X has $E[X] = \mu_X$ and $\text{var}(X) = \sigma_X^2$ and is denoted $X \sim N(\mu_X, \sigma_X^2)$. Its pdf is given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left\{-\frac{1}{2\sigma_X^2}(x - \mu_X)^2\right\}, \quad -\infty \leq x \leq \infty. \quad (2.14)$$

Showing that $E[X] = \mu_X$ and $\text{var}(X) = \sigma_X^2$ is a bit of work and is good calculus practice. As with the standard normal distribution, areas under the general normal curve cannot be computed analytically. Using numerical approximations, it can be shown that:

$$\begin{aligned}\Pr(\mu_X - \sigma_X < X < \mu_X + \sigma_X) &\approx 0.67, \\ \Pr(\mu_X - 2\sigma_X < X < \mu_X + 2\sigma_X) &\approx 0.95, \\ \Pr(\mu_X - 3\sigma_X < X < \mu_X + 3\sigma_X) &\approx 0.99.\end{aligned}$$

Hence, for a general normal random variable about 95% of the time we expect to see values within ± 2 standard deviations from its mean. Observations more than three standard deviations from the mean are very unlikely.

Example 2.22. Normal distribution for monthly returns.

Let R denote the monthly return on an investment in Microsoft stock, and assume that it is normally distributed with mean $\mu_R = 0.01$ and standard deviation $\sigma_R = 0.10$. That is, $R \sim N(0.01, (0.10)^2)$. Notice that $\sigma_R^2 = 0.01$ and is not in units of return per month. Figure 2.1.6 illustrates the distribution and is created using

```
mu.r = 0.01
sd.r = 0.1
x.vals = seq(-4, 4, length = 150) * sd.r + mu.r
plot(x.vals, dnorm(x.vals, mean = mu.r, sd = sd.r), type = "l", lwd = 2, col = "blue",
      xlab = "x", ylab = "pdf")
```

Notice that essentially all of the probability lies between -0.3 and 0.3 . Using the R function `pnorm()`, we can easily compute the probabilities $\Pr(R < -0.5)$, $\Pr(R < 0)$, $\Pr(R > 0.5)$ and $\Pr(R > 1)$:

```
pnorm(-0.5, mean = 0.01, sd = 0.1)
## [1] 1.7e-07

pnorm(0, mean = 0.01, sd = 0.1)
## [1] 0.46

1 - pnorm(0.5, mean = 0.01, sd = 0.1)
## [1] 4.79e-07

1 - pnorm(1, mean = 0.01, sd = 0.1)
## [1] 0
```

Using the R function `qnorm()`, we can find the quantiles $q_{0.01}$, $q_{0.05}$, $q_{0.95}$ and $q_{0.99}$:

```
a.vals = c(0.01, 0.05, 0.95, 0.99)
qnorm(a.vals, mean = 0.01, sd = 0.1)

## [1] -0.223 -0.154  0.174  0.243
```

Hence, over the next month, there are 1% and 5% chances of losing more than 22.2% and 15.5%, respectively. In addition, there are 5% and 1% chances of gaining more than 17.5% and 24.3%, respectively.

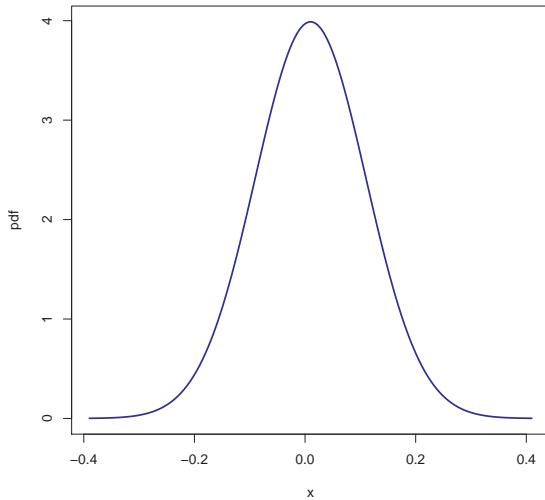


Fig. 2.6 Normal distribution for the monthly returns on Microsoft: $R \sim N(0.01, (0.10)^2)$.

Example 2.23. Risk-return tradeoff.

Consider the following investment problem. We can invest in two non-dividend paying stocks, Amazon and Boeing, over the next month. Let R_A denote the monthly return on Amazon and R_B denote the monthly return on Boeing. Assume that $R_A \sim N(0.02, (0.10)^2)$ and $R_B \sim N(0.01, (0.05)^2)$. Figure 2.1.6 shows the pdfs for the two returns. Notice that $\mu_A = 0.02 > \mu_B = 0.01$ but also that $\sigma_A = 0.10 > \sigma_B = 0.05$. The return we expect on Amazon is bigger than the return we expect on Boeing but the variability of Amazon's return is also greater than the variability of Boeing's return. The high return variability (volatility) of Amazon reflects the higher risk associated with investing in Amazon compared to investing in Boeing. If we invest in Boeing we get a lower expected return, but we also get less return variability or risk. This example illustrates the fundamental "no free lunch" principle of economics and finance: you can't get something for nothing. In general, to get a higher expected return you must be prepared to take on higher risk.

■

Example 2.24. Why the normal distribution may not be appropriate for simple returns.

Let R_t denote the simple annual return on an asset, and suppose that $R_t \sim N(0.05, (0.50)^2)$. Because asset prices must be non-negative, R_t must always be larger than -1 . However, the normal distribution is defined for $-\infty \leq R_t \leq \infty$ and based on the assumed normal distribution $\Pr(R_t < -1) = 0.018$. That is, there is a 1.8% chance that R_t is smaller than -1 . This implies that there is a 1.8% chance that the asset price at the end of the year will be negative! This is why the normal distribution may not be appropriate for simple returns.

■

Example 2.25. The normal distribution is more appropriate for continuously compounded returns.

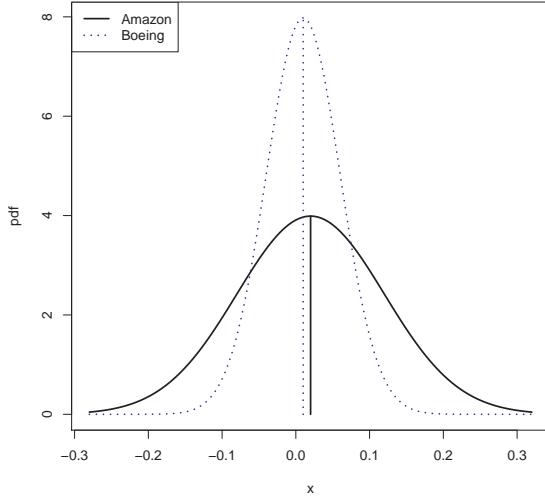


Fig. 2.7 Risk-return tradeoff between one-month investments in Amazon and Boeing stock.

Let $r_t = \ln(1 + R_t)$ denote the continuously compounded annual return on an asset, and suppose that $r_t \sim N(0.05, (0.50)^2)$. Unlike the simple return, the continuously compounded return can take on values less than -1 . In fact, r_t is defined for $-\infty \leq r_t \leq \infty$. For example, suppose $r_t = -2$. This implies a simple return of $R_t = e^{-2} - 1 = -0.865$.⁴ Then $\Pr(r_t \leq -2) = \Pr(R_t \leq -0.865) = 0.00002$. Although the normal distribution allows for values of r_t smaller than -1 , the implied simple return R_t will always be greater than -1 .

■

The Log-Normal Distribution

Let $X \sim N(\mu_X, \sigma_X^2)$, which is defined for $-\infty < X < \infty$. The log-normally distributed random variable Y is determined from the normally distributed random variable X using the transformation $Y = e^X$. In this case, we say that Y is log-normally distributed and write:

$$Y \sim \ln N(\mu_X, \sigma_X^2), \quad 0 < Y < \infty. \quad (2.15)$$

The pdf of the log-normal distribution for Y can be derived from the normal distribution for X using the change-of-variables formula from calculus and is given by:

$$f(y) = \frac{1}{y\sigma_X\sqrt{2\pi}} e^{-\frac{(\ln y - \mu_X)^2}{2\sigma_X^2}}. \quad (2.16)$$

Due to the exponential transformation, Y is only defined for non-negative values. It can be shown that:

⁴ If $r_t = -\infty$ then $R_t = e^{r_t} - 1 = e^{-\infty} - 1 = 0 - 1 = -1$.

$$\begin{aligned}\mu_Y &= E[Y] = e^{\mu_X + \sigma_X^2/2}, \\ \sigma_Y^2 &= \text{var}(Y) = e^{2\mu_X + \sigma_X^2} (e^{\sigma_X^2} - 1).\end{aligned}\tag{2.17}$$

Example 2.26. Log-normal distribution for simple returns.

Let $r_t = \ln(P_t/P_{t-1})$ denote the continuously compounded monthly return on an asset and assume that $r_t \sim N(0.05, (0.50)^2)$. That is, $\mu_r = 0.05$ and $\sigma_r = 0.50$. Let $R_t = (P_t - P_{t-1})/P_t$ denote the simple monthly return. The relationship between r_t and R_t is given by $r_t = \ln(1 + R_t)$ and $1 + R_t = e^{r_t}$. Since r_t is normally distributed, $1 + R_t$ is log-normally distributed. Notice that the distribution of $1 + R_t$ is only defined for positive values of $1 + R_t$. This is appropriate since the smallest value that R_t can take on is -1 . Using (2.17), the mean and variance for $1 + R_t$ are given by:

$$\begin{aligned}\mu_{1+R} &= e^{0.05+(0.5)^2/2} = 1.191 \\ \sigma_{1+R}^2 &= e^{2(0.05)+(0.5)^2} (e^{(0.5)^2} - 1) = 0.563\end{aligned}$$

The pdfs for r_t and $1 + R_t$ are shown in Figure 2.1.6. ■

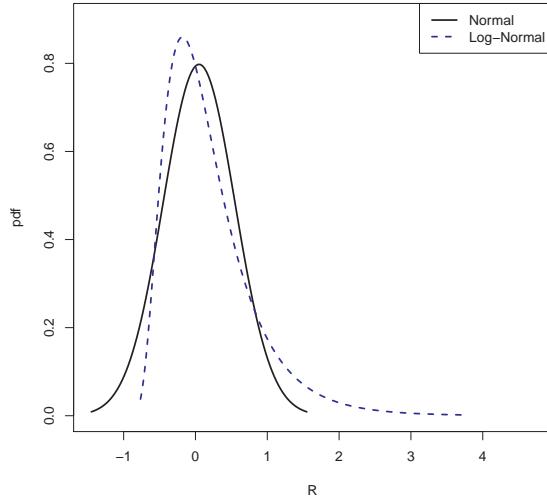


Fig. 2.8 Normal distribution for r_t and log-normal distribution for $1 + R_t = e^{r_t}$.

Skewness

The skewness of a random variable X , denoted $\text{skew}(X)$, measures the symmetry of a distribution about its mean value using the function $g(X) = (X - \mu_X)^3/\sigma_X^3$, where σ_X^3 is just $\text{sd}(X)$ raised to the third power:

$$\text{skew}(X) = \frac{E[(X - \mu_X)^3]}{\sigma_X^3} \quad (2.18)$$

When X is far below its mean $(X - \mu_X)^3$ is a big negative number, and when X is far above its mean $(X - \mu_X)^3$ is a big positive number. Hence, if there are more big values of X below μ_X then $\text{skew}(X) < 0$. Conversely, if there are more big values of X above μ_X then $\text{skew}(X) > 0$. If X has a symmetric distribution, then $\text{skew}(X) = 0$ since positive and negative values in (2.18) cancel out. If $\text{skew}(X) > 0$, then the distribution of X has a "long right tail" and if $\text{skew}(X) < 0$ the distribution of X has a "long left tail". These cases are illustrated in Figure 2.9.

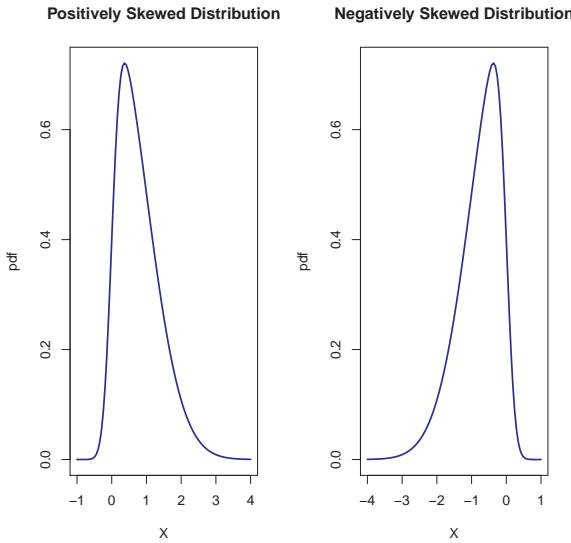


Fig. 2.9 Skewed distributions.

Example 2.27. Skewness for a discrete random variable.

Using the discrete distribution for the return on Microsoft stock in Table 2.1, the results that $\mu_X = 0.1$ and $\sigma_X = 0.141$, we have:

$$\begin{aligned} \text{skew}(X) &= [(-0.3 - 0.1)^3 \cdot (0.05) + (0.0 - 0.1)^3 \cdot (0.20) + (0.1 - 0.1)^3 \cdot (0.5) \\ &\quad + (0.2 - 0.1)^3 \cdot (0.2) + (0.5 - 0.1)^3 \cdot (0.05)] / (0.141)^3 \\ &= 0.0. \end{aligned}$$

■

Example 2.28. Skewness for a normal random variable.

Suppose X has a general normal distribution with mean μ_X and variance σ_X^2 . Then it can be shown that:

$$\text{skew}(X) = \int_{-\infty}^{\infty} \frac{(x - \mu_X)^3}{\sigma_X^3} \cdot \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{1}{2\sigma_X^2}(x - \mu_X)^2} dx = 0.$$

This result is expected since the normal distribution is symmetric about its mean value μ_X . ■

Example 2.29. Skewness for a log-Normal random variable.

Let $Y = e^X$, where $X \sim N(\mu_X, \sigma_X^2)$, be a log-normally distributed random variable with parameters μ_X and σ_X^2 . Then it can be shown that:

$$\text{skew}(Y) = (e^{\sigma_X^2} + 2) \sqrt{e^{\sigma_X^2} - 1} > 0.$$

Notice that $\text{skew}(Y)$ is always positive, indicating that the distribution of Y has a long right tail, and that it is an increasing function of σ_X^2 . This positive skewness is illustrated in Figure 2.1.6. ■

Kurtosis

The kurtosis of a random variable X , denoted $\text{kurt}(X)$, measures the thickness in the tails of a distribution and is based on $g(X) = (X - \mu_X)^4 / \sigma_X^4$:

$$\text{kurt}(X) = \frac{E[(X - \mu_X)^4]}{\sigma_X^4}, \quad (2.19)$$

where σ_X^4 is just $\text{sd}(X)$ raised to the fourth power. Since kurtosis is based on deviations from the mean raised to the fourth power, large deviations get lots of weight. Hence, distributions with large kurtosis values are ones where there is the possibility of extreme values. In contrast, if the kurtosis is small then most of the observations are tightly clustered around the mean and there is very little probability of observing extreme values.

Example 2.30. Kurtosis for a discrete random variable.

Using the discrete distribution for the return on Microsoft stock in Table 2.1, the results that $\mu_X = 0.1$ and $\sigma_X = 0.141$, we have:

$$\begin{aligned} \text{kurt}(X) &= [(-0.3 - 0.1)^4 \cdot (0.05) + (0.0 - 0.1)^4 \cdot (0.20) + (0.1 - 0.1)^4 \cdot (0.5) \\ &\quad + (0.2 - 0.1)^4 \cdot (0.2) + (0.5 - 0.1)^4 \cdot (0.05)] / (0.141)^4 \\ &= 6.5. \end{aligned}$$
■

Example 2.31. Kurtosis for a normal random variable.

Suppose X has a general normal distribution with mean μ_X and variance σ_X^2 . Then it can be shown that:

$$\text{kurt}(X) = \int_{-\infty}^{\infty} \frac{(x - \mu_X)^4}{\sigma_X^4} \cdot \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{1}{2}(\frac{x-\mu_X}{\sigma_X})^2} dx = 3.$$

Hence, a kurtosis of 3 is a benchmark value for tail thickness of bell-shaped distributions. If a distribution has a kurtosis greater than 3, then the distribution has thicker tails than the normal distribution. If a distribution has kurtosis less than 3, then the distribution has thinner tails than the normal. ■

Sometimes the kurtosis of a random variable is described relative to the kurtosis of a normal random variable. This relative value of kurtosis is referred to as *excess kurtosis* and is defined as:

$$\text{ekurt}(X) = \text{kurt}(X) - 3 \quad (2.20)$$

If the excess kurtosis of a random variable is equal to zero then the random variable has the same kurtosis as a normal random variable. If excess kurtosis is greater than zero, then kurtosis is larger than that for a normal; if excess kurtosis is less than zero, then kurtosis is less than that for a normal.

The Student's t Distribution

The kurtosis of a random variable gives information on the tail thickness of its distribution. The normal distribution, with kurtosis equal to three, gives a benchmark for the tail thickness of symmetric distributions. A distribution similar to the standard normal distribution but with fatter tails, and hence larger kurtosis, is the Student's t distribution. If X has a Student's t distribution with degrees of freedom parameter v , denoted $X \sim t_v$, then its pdf has the form:

$$f(x) = \frac{\Gamma(\frac{v+1}{2})}{\sqrt{v\pi}\Gamma(\frac{v}{2})} \left(1 + \frac{x^2}{v}\right)^{-\left(\frac{v+1}{2}\right)}, \quad -\infty < x < \infty, \quad v > 0. \quad (2.21)$$

where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ denotes the gamma function. It can be shown that:

$$\begin{aligned} E[X] &= 0, \quad v > 1 \\ \text{var}(X) &= \frac{v}{v-2}, \quad v > 2, \\ \text{skew}(X) &= 0, \quad v > 3, \\ \text{kurt}(X) &= \frac{6}{v-4} + 3, \quad v > 4. \end{aligned}$$

The parameter v controls the scale and tail thickness of the distribution. If v is close to four, then the kurtosis is large and the tails are thick. If $v < 4$, then $\text{kurt}(X) = \infty$. As $v \rightarrow \infty$ the Student's t pdf approaches that of a standard normal random variable and $\text{kurt}(X) = 3$. Figure 2.1.6 shows plots of the Student's t density for various values of v as well as the standard normal density.

Example 2.32. Computing tail probabilities and quantiles from the Student's t distribution.

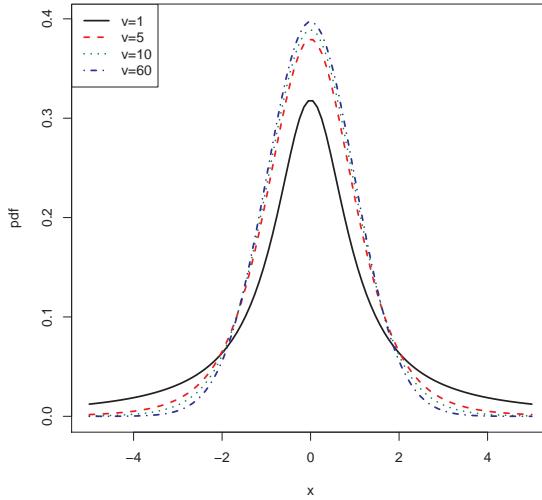


Fig. 2.10 Student's t density with $v = 1, 5, 10$ and 60 .

The R functions `pt()` and `qt()` can be used to compute the cdf and quantiles of a Student's t random variable. For $v = 1, 2, 5, 10, 60, 100$ and ∞ the 1% quantiles can be computed using:

```
v = c(1, 2, 5, 10, 60, 100, Inf)
qt(0.01, df = v)

## [1] -31.82 -6.96 -3.36 -2.76 -2.39 -2.36 -2.33
```

For $v = 1, 2, 5, 10, 60, 100$ and ∞ the values of $\Pr(X < -3)$ are:

```
pt(-3, df = v)

## [1] 0.10242 0.04773 0.01505 0.00667 0.00196 0.00170 0.00135
```

■

2.1.7 Linear functions of a random variable

Let X be a random variable either discrete or continuous with $E[X] = \mu_X$, $\text{var}(X) = \sigma_X^2$, and let a and b be known constants. Define a new random variable Y via the linear function of X :

$$Y = g(X) = aX + b.$$

Then the following results hold:

$$\begin{aligned}\mu_Y &= E[Y] = a \cdot E[X] + b = a \cdot \mu_X + b, \\ \sigma_Y^2 &= \text{var}(Y) = a^2 \text{var}(X) = a^2 \sigma_X^2, \\ \sigma_Y &= \text{sd}(Y) = a \cdot \text{sd}(X) = a \cdot \sigma_X.\end{aligned}$$

The first result shows that expectation is a linear operation. That is,

$$E[aX + b] = aE[X] + b.$$

The second result shows that adding a constant to X does not affect its variance, and that the effect of multiplying X by the constant a increases the variance of X by the square of a . These results will be used often enough that it is instructive to go through the derivations.

Consider the first result. Let X be a discrete random variable. By the definition of $E[g(X)]$, with $g(X) = b + aX$, we have:

$$\begin{aligned}E[Y] &= \sum_{x \in S_X} (ax + b) \cdot \Pr(X = x) \\ &= a \sum_{x \in S_X} x \cdot \Pr(X = x) + b \sum_{x \in S_X} \Pr(X = x) \\ &= a \cdot E[X] + b \cdot 1 \\ &= a \cdot \mu_X + b.\end{aligned}$$

If X is a continuous random variable then by the linearity of integration:

$$\begin{aligned}E[Y] &= \int_{-\infty}^{\infty} (ax + b)f(x) dx = a \int_{-\infty}^{\infty} xf(x) dx + b \int_{-\infty}^{\infty} f(x) dx \\ &= aE[X] + b.\end{aligned}$$

Next consider the second result. Since $\mu_Y = a\mu_X + b$ we have:

$$\begin{aligned}\text{var}(Y) &= E[(Y - \mu_Y)^2] \\ &= E[(aX + b - (a\mu_X + b))^2] \\ &= E[(a(X - \mu_X) + (b - b))^2] \\ &= E[a^2(X - \mu_X)^2] \\ &= a^2 E[(X - \mu_X)^2] \text{ (by the linearity of } E[\cdot]) \\ &= a^2 \text{var}(X)\end{aligned}$$

Notice that the derivation of the second result works for discrete and continuous random variables.

A normal random variable has the special property that a linear function of it is also a normal random variable. The following proposition establishes the result.

Proposition 2.1. *Let $X \sim N(\mu_x, \sigma_x^2)$ and let a and b be constants. Let $Y = aX + b$. Then $Y \sim N(a\mu_x + b, a^2\sigma_x^2)$.*

The above property is special to the normal distribution and may or may not hold for a random variable with a distribution that is not normal.

Example 2.33. Standardizing a random variable.

Let X be a random variable with $E[X] = \mu_X$ and $\text{var}(X) = \sigma_X^2$. Define a new random variable Z as:

$$Z = \frac{X - \mu_X}{\sigma_X} = \frac{1}{\sigma_X}X - \frac{\mu_X}{\sigma_X},$$

which is a linear function $aX + b$ where $a = \frac{1}{\sigma_X}$ and $b = -\frac{\mu_X}{\sigma_X}$. This transformation is called *standardizing* the random variable X since,

$$\begin{aligned} E[Z] &= \frac{1}{\sigma_X}E[X] - \frac{\mu_X}{\sigma_X} = \frac{1}{\sigma_X}\mu_X - \frac{\mu_X}{\sigma_X} = 0, \\ \text{var}(Z) &= \left(\frac{1}{\sigma_X}\right)^2 \text{var}(X) = \frac{\sigma_X^2}{\sigma_X^2} = 1. \end{aligned}$$

Hence, standardization creates a new random variable with mean zero and variance 1. In addition, if X is normally distributed then $Z \sim N(0, 1)$.

■

Example 2.34. Computing probabilities using standardized random variables.

Let $X \sim N(2, 4)$ and suppose we want to find $\Pr(X > 5)$ but we only know probabilities associated with a standard normal random variable $Z \sim N(0, 1)$. We solve the problem by standardizing X as follows:

$$\Pr(X > 5) = \Pr\left(\frac{X - 2}{\sqrt{4}} > \frac{5 - 2}{\sqrt{4}}\right) = \Pr\left(Z > \frac{3}{2}\right) = 0.06681.$$

■

Standardizing a random variable is often done in the construction of test statistics. For example, the so-called *t-statistic* or *t-ratio* used for testing simple hypotheses on coefficients is constructed by the standardization process.

A non-standard random variable X with mean μ_X and variance σ_X^2 can be created from a standard random variable via the linear transformation:

$$X = \mu_X + \sigma_X \cdot Z. \quad (2.22)$$

This result is useful for modeling purposes as illustrated in the next example.

Example 2.35. Quantile of general normal random variable.

Let $X \sim N(\mu_X, \sigma_X^2)$. Quantiles of X can be conveniently computed using:

$$q_\alpha = \mu_X + \sigma_X z_\alpha \quad (2.23)$$

where $\alpha \in (0, 1)$ and z_α is the $\alpha \times 100\%$ quantile of a standard normal random variable. This formula is derived as follows. By the definition z_α and (2.22)

$$\alpha = \Pr(Z \leq z_\alpha) = \Pr(\mu_X + \sigma_X \cdot Z \leq \mu_X + \sigma_X \cdot z_\alpha) = \Pr(X \leq \mu_X + \sigma_X \cdot z_\alpha),$$

which implies (2.23). ■

Example 2.36. Nonstandard Student's t distribution

The Student's t distribution defined in (2.21) has zero mean and variance equal to $v/(v-2)$ where v is the degrees of freedom parameter. We can define a Student's t random variable with non-zero mean μ and variance σ^2 from a Student's t random variable X as follows:

$$Y = \mu + \sigma \left(\frac{v-2}{v} \right)^{1/2} X = \mu + \sigma Z_v,$$

where $Z_v = \left(\frac{v-2}{v} \right)^{1/2} X$. The random variable Z_v is called a standardized Student's t random variable with v degrees of freedom since $E[Z_v] = 0$ and $\text{var}(Z_v) = \left(\frac{v-2}{v} \right) \text{var}(X) = \left(\frac{v-2}{v} \right) \left(\frac{v}{v-2} \right) = 1$. Then

$$\begin{aligned} E[Y] &= \mu + \sigma E[Z] = \mu, \\ \text{var}(Y) &= \sigma^2 \text{var}(Z) = \sigma^2. \end{aligned}$$
■

Example 2.37. Constant expected return model for asset returns.

Let r denote the monthly continuously compounded return on an asset, and assume that $r \sim N(\mu_r, \sigma_r^2)$. Then r can be expressed as:

$$r = \mu_r + \sigma_r \cdot \varepsilon, \quad \varepsilon \sim N(0, 1).$$

The random variable ε can be interpreted as representing the random news arriving in a given month that makes the observed return differ from its expected value μ . The fact that ε has mean zero means that news, on average, is neutral. The value of σ_r represents the typical size of a news shock. The bigger is σ_r , the larger is the impact of a news shock and vice-versa. ■

2.1.8 Value at Risk: An Introduction

As an example of working with linear functions of a normal random variable, and to illustrate the concept of *Value-at-Risk* (VaR), consider an investment of \$10,000 in Microsoft stock over the next month. Let R denote the monthly simple return on Microsoft stock and assume that $R \sim N(0.05, (0.10)^2)$. That is, $E[R] = \mu_R = 0.05$ and $\text{var}(R) = \sigma_R^2 = (0.10)^2$. Let W_0 denote the investment value at the beginning of the month and W_1 denote the investment value at the end of the month. In this example, $W_0 = \$10,000$. Consider the following questions:

- (i) What is the probability distribution of end of month wealth, W_1 ?
- (ii) What is the probability that end of month wealth is less than \$9,000, and what must the return on Microsoft be for this to happen?
- (iii) What is the loss in dollars that would occur if the return on Microsoft stock is equal to its 5% quantile, $q_{.05}$? That is, what is the monthly 5% VaR on the \$10,000 investment in Microsoft?

To answer (i), note that end of month wealth, W_1 , is related to initial wealth W_0 and the return on Microsoft stock R via the linear function:

$$W_1 = W_0(1 + R) = W_0 + W_0R = \$10,000 + \$10,000 \cdot R.$$

Using the properties of linear functions of a random variable we have:

$$E[W_1] = W_0 + W_0E[R] = \$10,000 + \$10,000(0.05) = \$10,500,$$

and,

$$\begin{aligned} \text{var}(W_1) &= (W_0)^2\text{var}(R) = (\$10,000)^2(0.10)^2, \\ \text{sd}(W_1) &= (\$10,000)(0.10) = \$1,000. \end{aligned}$$

Further, since R is assumed to be normally distributed it follows that W_1 is normally distributed:

$$W_1 \sim N(\$10,500, (\$1,000)^2).$$

To answer (ii), we use the above normal distribution for W_1 to get:

$$\Pr(W_w < \$9,000) = 0.067.$$

To find the return that produces end of month wealth of \$9,000, or a loss of \$10,000-\$9,000=\$1,000, we solve

$$R = \frac{\$9,000 - \$10,000}{\$10,000} = -0.10.$$

If the monthly return on Microsoft is -10% or less, then end of month wealth will be \$9,000 or less. Notice that $R = -0.10$ is the 6.7% quantile of the distribution of R :

$$\Pr(R < -0.10) = 0.067.$$

Question (iii) can be answered in two equivalent ways. First, we use $R \sim N(0.05, (0.10)^2)$ and solve for the 5% quantile of Microsoft Stock using (2.23)⁵:

$$\begin{aligned} \Pr(R < q_{.05}^R) &= 0.05 \\ \Rightarrow q_{.05}^R &= \mu_R + \sigma_R \cdot z_{.05} = 0.05 + 0.10 \cdot (-1.645) = -0.114. \end{aligned}$$

That is, with 5% probability the return on Microsoft stock is -11.4% or less. Now, if the return on Microsoft stock is -11.4% the loss in investment value is $\$10,000 \cdot (0.114) = \$1,144$. Hence, \$1,144 is the 5% VaR over the next month on the \$10,000 investment in Microsoft

⁵ Using R, $z_{.05} = qnorm(0.05) = -1.645$.

stock. For the second method, use $W_1 \sim N(\$10,500, (\$1,000)^2)$ and solve for the 5% quantile of end of month wealth directly:

$$\Pr(W_1 < q_{.05}^{W_1}) = 0.05 \Rightarrow q_{.05}^{W_1} = \$8,856.$$

This corresponds to a loss of investment value of $\$10,000 - \$8,856 = \$1,144$. Hence, if W_0 represents the initial wealth and $q_{.05}^{W_1}$ is the 5% quantile of the distribution of W_1 then the 5% VaR is:

$$\text{VaR}_{.05} = W_0 - q_{.05}^{W_1} = \$10,000 - \$8,856 = \$1,144.$$

In general if W_0 represents the initial wealth in dollars and q_α^R is the $\alpha \times 100\%$ quantile of distribution of the simple return R , then the $\alpha \times 100\%$ VaR is defined as:

$$\text{VaR}_\alpha = |W_0 \cdot q_\alpha^R|. \quad (2.24)$$

In words, VaR_α represents the dollar loss that could occur with probability α . By convention, it is reported as a positive number (hence the use of the absolute value function).

Value-at-Risk calculations for continuously compounded returns

The above calculations illustrate how to calculate value-at-risk using the normal distribution for simple returns. However, as argued in Example 31, the normal distribution may not be appropriate for characterizing the distribution of simple returns and may be more appropriate for characterizing the distribution of continuously compounded returns. Let R denote the simple monthly return, let $r = \ln(1 + R)$ denote the continuously compounded return and assume that $r \sim N(\mu_r, \sigma_r^2)$. The $\alpha \times 100\%$ monthly VaR on an investment of $\$W_0$ is computed using the following steps:

1. Compute the $\alpha \cdot 100\%$ quantile, q_α^r , from the normal distribution for the continuously compounded return r :

$$q_\alpha^r = \mu_r + \sigma_r z_\alpha,$$

where z_α is the $\alpha \cdot 100\%$ quantile of the standard normal distribution.

2. Convert the continuously compounded return quantile, q_α^r , to a simple return quantile using the transformation:

$$q_\alpha^R = e^{q_\alpha^r} - 1$$

3. Compute VaR using the simple return quantile (2.24).

Example 2.38. Computing VaR from simple and continuously compounded returns using R.

Let R denote the simple monthly return on Microsoft stock and assume that $R \sim N(0.05, (0.10)^2)$. Consider an initial investment of $W_0 = \$10,000$. To compute the 1% and 5% VaR values over the next month use:

```
mu.R = 0.05
sd.R = 0.1
w0 = 10000
q.01.R = mu.R + sd.R * qnorm(0.01)
```

```

q.05.R = mu.R + sd.R * qnorm(0.05)
VaR.01 = abs(q.01.R * w0)
VaR.05 = abs(q.05.R * w0)
VaR.01

## [1] 1826

VaR.05

## [1] 1145

```

Hence with 1% and 5% probability the loss over the next month is at least \$1,826 and \$1,145, respectively.

Let r denote the continuously compounded return on Microsoft stock and assume that $r \sim N(0.05, (0.10)^2)$. To compute the 1% and 5% VaR values over the next month use:

```

mu.r = 0.05
sd.r = 0.1
q.01.R = exp(mu.r + sd.r * qnorm(0.01)) - 1
q.05.R = exp(mu.r + sd.r * qnorm(0.05)) - 1
VaR.01 = abs(q.01.R * w0)
VaR.05 = abs(q.05.R * w0)
VaR.01

## [1] 1669

VaR.05

## [1] 1082

```

Notice that when $1 + R = e^r$ has a log-normal distribution, the 1% and 5% VaR values (losses) are slightly smaller than when R is normally distributed. This is due to the positive skewness of the log-normal distribution. ■

2.2 Bivariate Distributions

So far we have only considered probability distributions for a single random variable. In many situations, we want to be able to characterize the probabilistic behavior of two or more random variables simultaneously. In this section, we discuss bivariate distributions.

2.2.1 Discrete random variables

Let X and Y be discrete random variables with sample spaces S_X and S_Y , respectively. The likelihood that X and Y takes values in the joint sample space $S_{XY} = S_X \times S_Y$ is

determined by the joint probability distribution $p(x, y) = \Pr(X = x, Y = y)$. The function $p(x, y)$ satisfies:

- (i) $p(x, y) > 0$ for $x, y \in S_{XY}$;
- (ii) $p(x, y) = 0$ for $x, y \notin S_{XY}$;
- (iii) $\sum_{x,y \in S_{XY}} p(x, y) = \sum_{x \in S_X} \sum_{y \in S_Y} p(x, y) = 1$.

Example 2.39. Bivariate discrete distribution for stock returns.

Let X denote the monthly return (in percent) on Microsoft stock and let Y denote the monthly return on Apple stock. For simplicity suppose that the sample spaces for X and Y are $S_X = \{0, 1, 2, 3\}$ and $S_Y = \{0, 1\}$ so that the random variables X and Y are discrete. The joint sample space is the two dimensional grid $S_{XY} = \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1), (3, 0), (3, 1)\}$. Table 2.3 illustrates the joint distribution for X and Y . From the table, $p(0, 0) = \Pr(X = 0, Y = 0) = 1/8$. Notice that the sum of all the entries in the table sum to unity. ■

		Y		$\Pr(X)$
		% 0		
X	0	1/8	0	1/8
	1	2/8	1/8	3/8
	2	1/8	2/8	3/8
	3	0	1/8	1/8
		$\Pr(Y)$	4/8	4/8
				1

Table 2.3 Discrete bivariate distribution for Microsoft and Apple stock prices.

Marginal Distributions

The joint probability distribution tells the probability of X and Y occurring together. What if we only want to know about the probability of X occurring, or the probability of Y occurring?

Example 2.40. Find $\Pr(X = 0)$ and $\Pr(Y = 1)$ from joint distribution.

Consider the joint distribution in Table 2.3. What is $\Pr(X = 0)$ regardless of the value of Y ? Now X can occur if $Y = 0$ or if $Y = 1$ and since these two events are mutually exclusive we have that $\Pr(X = 0) = \Pr(X = 0, Y = 0) + \Pr(X = 0, Y = 1) = 0 + 1/8 = 1/8$. Notice that this probability is equal to the horizontal (row) sum of the probabilities in the table at $X = 0$. We can find $\Pr(Y = 1)$ in a similar fashion: $\Pr(Y = 1) = \Pr(X = 0, Y = 1) + \Pr(X = 1, Y = 1) + \Pr(X = 2, Y = 1) + \Pr(X = 3, Y = 1) = 0 + 1/8 + 2/8 + 1/8 = 4/8$. This probability is the vertical (column) sum of the probabilities in the table at $Y = 1$.

■

The probability $\Pr(X = x)$ is called the *marginal probability* of X and is given by:

$$\Pr(X = x) = \sum_{y \in S_Y} \Pr(X = x, Y = y). \quad (2.25)$$

Similarly, the marginal probability of $Y = y$ is given by:

$$\Pr(Y = y) = \sum_{x \in S_X} \Pr(X = x, Y = y). \quad (2.26)$$

Example 2.41. Marginal probabilities of discrete bivariate distribution.

The marginal probabilities of $X = x$ are given in the last column of Table 2.3, and the marginal probabilities of $Y = y$ are given in the last row of Table 2.3. Notice that these probabilities sum to 1. For future reference we note that $E[X] = 3/2$, $\text{var}(X) = 3/4$, $E[Y] = 1/2$, and $\text{var}(Y) = 1/4$.

■

Conditional Distributions

For random variables in Table 2.3, suppose we know that the random variable Y takes on the value $Y = 0$. How does this knowledge affect the likelihood that X takes on the values 0, 1, 2 or 3? For example, what is the probability that $X = 0$ *given that* we know $Y = 0$? To find this probability, we use Bayes' law and compute the *conditional probability*:

$$\Pr(X = 0|Y = 0) = \frac{\Pr(X = 0, Y = 0)}{\Pr(Y = 0)} = \frac{1/8}{4/8} = 1/4.$$

The notation $\Pr(X = 0|Y = 0)$ is read as "the probability that $X = 0$ given that $Y = 0$ ". Notice that $\Pr(X = 0|Y = 0) = 1/4 > \Pr(X = 0) = 1/8$. Hence, knowledge that $Y = 0$ increases the likelihood that $X = 0$. Clearly, X depends on Y .

Now suppose that we know that $X = 0$. How does this knowledge affect the probability that $Y = 0$? To find out we compute:

$$\Pr(Y = 0|X = 0) = \frac{\Pr(X = 0, Y = 0)}{\Pr(X = 0)} = \frac{1/8}{1/8} = 1.$$

Notice that $\Pr(Y = 0|X = 0) = 1 > \Pr(Y = 0) = 1/2$. That is, knowledge that $X = 0$ makes it certain that $Y = 0$.

In general, the conditional probability that $X = x$ given that $Y = y$ (provided $\Pr(Y = y) \neq 0$) is,

$$\Pr(X = x|Y = y) = \frac{\Pr(X = x, Y = y)}{\Pr(Y = y)}, \quad (2.27)$$

and the conditional probability that $Y = y$ given that $X = x$ (provided $\Pr(X = x) \neq 0$) is,

$$\Pr(Y = y|X = x) = \frac{\Pr(X = x, Y = y)}{\Pr(X = x)}. \quad (2.28)$$

Example 2.42. Conditional distributions.

For the bivariate distribution in Table 2.3, the conditional probabilities along with marginal probabilities are summarized in Tables 2.4 and 2.5. Notice that the marginal distribution of X is centered at $x = 3/2$ whereas the conditional distribution of $X|Y = 0$ is centered at $x = 1$ and the conditional distribution of $X|Y = 1$ is centered at $x = 2$.

■

x	$\Pr(X = x)$	$\Pr(X Y = 0)$	$\Pr(X Y = 1)$
0	1/8	2/8	0
1	3/8	4/8	2/8
2	3/8	2/8	4/8
3	1/8	0	2/8

Table 2.4 Conditional probability distribution of X from bivariate discrete distribution.

y	$\Pr(Y = y)$	$\Pr(Y X = 0)$	$\Pr(Y X = 1)$	$\Pr(Y X = 2)$	$\Pr(Y X = 3)$
0	1/2	1	2/3	1/3	0
1	1/2	0	1/3	2/3	1

Table 2.5 Conditional distribution of Y from bivariate discrete distribution.

Conditional expectation and conditional variance

Just as we defined shape characteristics of the marginal distributions of X and Y , we can also define shape characteristics of the conditional distributions of $X|Y = y$ and $Y|X = x$. The most important shape characteristics are the *conditional expectation (conditional mean)* and the *conditional variance*. The conditional mean of $X|Y = y$ is denoted by $\mu_{X|Y=y} = E[X|Y = y]$, and the conditional mean of $Y|X = x$ is denoted by $\mu_{Y|X=x} = E[Y|X = x]$. These means are computed as:

$$\mu_{X|Y=y} = E[X|Y = y] = \sum_{x \in S_X} x \cdot \Pr(X = x|Y = y), \quad (2.29)$$

$$\mu_{Y|X=x} = E[Y|X = x] = \sum_{y \in S_Y} y \cdot \Pr(Y = y|X = x). \quad (2.30)$$

Similarly, the conditional variance of $X|Y = y$ is denoted by $\sigma_{X|Y=y}^2 = \text{var}(X|Y = y)$ and the conditional variance of $Y|X = x$ is denoted by $\sigma_{Y|X=x}^2 = \text{var}(Y|X = x)$. These variances are computed as:

$$\sigma_{X|Y=y}^2 = \text{var}(X|Y = y) = \sum_{x \in S_X} (x - \mu_{X|Y=y})^2 \cdot \Pr(X = x|Y = y), \quad (2.31)$$

$$\sigma_{Y|X=x}^2 = \text{var}(Y|X = x) = \sum_{y \in S_Y} (y - \mu_{Y|X=x})^2 \cdot \Pr(Y = y|X = x). \quad (2.32)$$

Example 2.43. Compute conditional expectation and conditional variance.

For the random variables in Table 2.3, we have the following conditional moments for X :

$$E[X|Y = 0] = 0 \cdot 1/4 + 1 \cdot 1/2 + 2 \cdot 1/4 + 3 \cdot 0 = 1,$$

$$E[X|Y = 1] = 0 \cdot 0 + 1 \cdot 1/4 + 2 \cdot 1/2 + 3 \cdot 1/4 = 2,$$

$$\text{var}(X|Y = 0) = (0 - 1)^2 \cdot 1/4 + (1 - 1)^2 \cdot 1/2 + (2 - 1)^2 \cdot 1/2 + (3 - 1)^2 \cdot 0 = 1/2,$$

$$\text{var}(X|Y = 1) = (0 - 2)^2 \cdot 0 + (1 - 2)^2 \cdot 1/4 + (2 - 2)^2 \cdot 1/2 + (3 - 2)^2 \cdot 1/4 = 1/2.$$

Compare these values to $E[X] = 3/2$ and $\text{var}(X) = 3/4$. Notice that as y increases, $E[X|Y = y]$ increases.

For Y , similar calculations gives:

$$E[Y|X = 0] = 0, E[Y|X = 1] = 1/3, E[Y|X = 2] = 2/3, E[Y|X = 3] = 1,$$

$$\text{var}(Y|X = 0) = 0, \text{var}(Y|X = 1) = 0.2222, \text{var}(Y|X = 2) = 0.2222, \text{var}(Y|X = 3) = 0.$$

Compare these values to $E[Y] = 1/2$ and $\text{var}(Y) = 1/4$. Notice that as x increases $E[Y|X = x]$ increases. ■

Conditional expectation and the regression function

Consider the problem of predicting the value Y given that we know $X = x$. A natural predictor to use is the conditional expectation $E[Y|X = x]$. In this prediction context, the conditional expectation $E[Y|X = x]$ is called the *regression function*. The graph with $E[Y|X = x]$ on the vertical axis and x on the horizontal axis gives the *regression line*. The relationship between Y and the regression function may be expressed using the trivial identity:

$$\begin{aligned} Y &= E[Y|X = x] + Y - E[Y|X = x] \\ &= E[Y|X = x] + \varepsilon, \end{aligned} \quad (2.33)$$

where $\varepsilon = Y - E[Y|X]$ is called the *regression error*.

Example 2.44. Regression line for bivariate discrete distribution.

For the random variables in Table 2.3, the regression line is plotted in Figure 2.2.1. Notice that there is a linear relationship between $E[Y|X = x]$ and x . When such a linear relationship exists we call the regression function a *linear regression*. Linearity of the regression function, however, is not guaranteed. It may be the case that there is a non-linear (e.g., quadratic) relationship between $E[Y|X = x]$ and x . In this case, we call the regression function a *non-linear regression*.

■

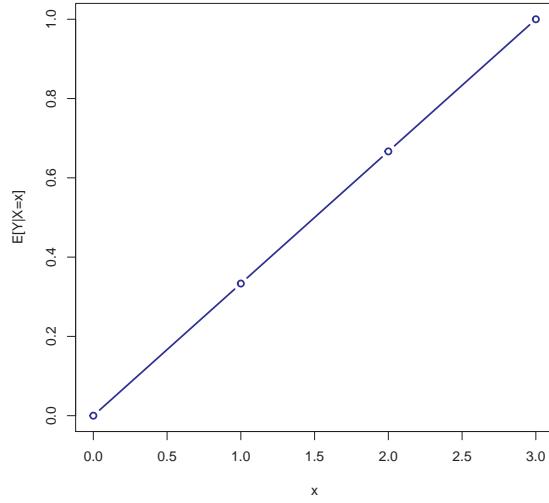


Fig. 2.11 Regression function $E[Y|X = x]$ from discrete bivariate distribution.

Law of Total Expectations

For the random variables in Table 2.3 notice that:

$$\begin{aligned} E[X] &= E[X|Y = 0] \cdot \Pr(Y = 0) + E[X|Y = 1] \cdot \Pr(Y = 1) \\ &= 1 \cdot 1/2 + 2 \cdot 1/2 = 3/2, \end{aligned}$$

and,

$$\begin{aligned} E[Y] &= E[Y|X = 0] \cdot \Pr(X = 0) + E[Y|X = 1] \cdot \Pr(X = 1) \\ &\quad + E[Y|X = 2] \cdot \Pr(X = 2) + E[Y|X = 3] \cdot \Pr(X = 3) = 1/2. \end{aligned}$$

This result is known as the *law of total expectations*. In general, for two random variables X and Y (discrete or continuous) we have,

$$\begin{aligned} E[X] &= E[E[X|Y]], \\ E[Y] &= E[E[Y|X]], \end{aligned} \tag{2.34}$$

where the first expectation is taken with respect to Y and the second expectation is taken with respect to X .

2.2.2 Bivariate distributions for continuous random variables

Let X and Y be continuous random variables defined over the real line. We characterize the joint probability distribution of X and Y using the joint probability function (pdf) $f(x, y)$ such that $f(x, y) \geq 0$ and,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1.$$

The three-dimensional plot of the joint probability distribution gives a probability surface whose total volume is unity. To compute joint probabilities of $x_1 \leq X \leq x_2$ and $y_1 \leq Y \leq y_2$, we need to find the volume under the probability surface over the grid where the intervals $[x_1, x_2]$ and $[y_1, y_2]$ overlap. Finding this volume requires solving the double integral:

$$\Pr(x_1 \leq X \leq x_2, y_1 \leq Y \leq y_2) = \int_{x_1}^{x_2} \int_{y_1}^{y_2} f(x, y) dx dy.$$

Example 2.45. Bivariate standard normal distribution.

A standard bivariate normal pdf for X and Y has the form:

$$f(x, y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}, \quad -\infty \leq x, y \leq \infty \quad (2.35)$$

and has the shape of a symmetric bell (think Liberty Bell) centered at $x = 0$ and $y = 0$. To find $\Pr(-1 < X < 1, -1 < Y < 1)$ we must solve:

$$\int_{-1}^1 \int_{-1}^1 \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} dx dy,$$

which, unfortunately, does not have an analytical solution. Numerical approximation methods are required to evaluate the above integral. The function `pmvnorm()` in the R package `mvtnorm` can be used to evaluate areas under the bivariate standard normal surface. To compute $\Pr(-1 < X < 1, -1 < Y < 1)$ use:

```
library(mvtnorm)
pmvnorm(lower = c(-1, -1), upper = c(1, 1))

## [1] 0.466
## attr(,"error")
## [1] 1e-15
## attr(,"msg")
## [1] "Normal Completion"
```

Here, $\Pr(-1 < X < 1, -1 < Y < 1) = 0.4661$. The attribute `error` gives the estimated absolute error of the approximation, and the attribute `message` tells the status of the algorithm used for the approximation. See the online help for `pmvnorm` for more details.



Marginal and conditional distributions

The marginal pdf of X is found by integrating y out of the joint pdf $f(x, y)$ and the marginal pdf of Y is found by integrating x out of the joint pdf $f(x, y)$:

$$f(x) = \int_{-\infty}^{\infty} f(x, y) dy, \quad (2.36)$$

$$f(y) = \int_{-\infty}^{\infty} f(x, y) dx. \quad (2.37)$$

The conditional pdf of X given that $Y = y$, denoted $f(x|y)$, is computed as,

$$f(x|y) = \frac{f(x, y)}{f(y)}, \quad (2.38)$$

and the conditional pdf of Y given that $X = x$ is computed as,

$$f(y|x) = \frac{f(x, y)}{f(x)}. \quad (2.39)$$

The conditional means are computed as,

$$\mu_{X|Y=y} = E[X|Y = y] = \int x \cdot p(x|y) dx, \quad (2.40)$$

$$\mu_{Y|X=x} = E[Y|X = x] = \int y \cdot p(y|x) dy, \quad (2.41)$$

and the conditional variances are computed as,

$$\sigma_{X|Y=y}^2 = \text{var}(X|Y = y) = \int (x - \mu_{X|Y=y})^2 p(x|y) dx, \quad (2.42)$$

$$\sigma_{Y|X=x}^2 = \text{var}(Y|X = x) = \int (y - \mu_{Y|X=x})^2 p(y|x) dy. \quad (2.43)$$

Example 2.46. Conditional and marginal distributions from bivariate standard normal.

Suppose X and Y are distributed bivariate standard normal. To find the marginal distribution of X we use (2.36) and solve:

$$f(x) = \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)} dy = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}.$$

Hence, the marginal distribution of X is standard normal. Similar calculations show that the marginal distribution of Y is also standard normal. To find the conditional distribution of $X|Y = y$ we use (2.38) and solve:

$$\begin{aligned} f(x|y) &= \frac{f(x, y)}{f(x)} = \frac{\frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}}{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}} \\ &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x^2+y^2)+\frac{1}{2}y^2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \\ &= f(x) \end{aligned}$$

So, for the standard bivariate normal distribution $f(x|y) = f(x)$ which does not depend on y . Similar calculations show that $f(y|x) = f(y)$.

■

2.2.3 Independence

Let X and Y be two discrete random variables. Intuitively, X is independent of Y if knowledge about Y does not influence the likelihood that $X = x$ for all possible values of $x \in S_X$ and $y \in S_Y$. Similarly, Y is independent of X if knowledge about X does not influence the likelihood that $Y = y$ for all values of $y \in S_Y$. We represent this intuition formally for discrete random variables as follows.

Definition 2.7. Let X and Y be discrete random variables with sample spaces S_X and S_Y , respectively. X and Y are independent random variables iff:

$$\begin{aligned}\Pr(X = x|Y = y) &= \Pr(X = x), \text{ for all } x \in S_X, y \in S_Y \\ \Pr(Y = y|X = x) &= \Pr(Y = y), \text{ for all } x \in S_X, y \in S_Y\end{aligned}$$

Example 2.47. Check independence of bivariate discrete random variables.

For the data in Table 2.3, we know that $\Pr(X = 0|Y = 0) = 1/4 \neq \Pr(X = 0) = 1/8$ so X and Y are not independent.

■

Proposition 2.2. Let X and Y be discrete random variables with sample spaces S_X and S_Y , respectively. X and Y are independent if and only if (iff):

$$\Pr(X = x, Y = y) = \Pr(X = x) \cdot \Pr(Y = y), \text{ for all } x \in S_X, y \in S_Y.$$

Intuition for the above result follows from:

$$\begin{aligned}\Pr(X = x|Y = y) &= \frac{\Pr(X = x, Y = y)}{\Pr(Y = y)} = \frac{\Pr(X = x) \cdot \Pr(Y = y)}{\Pr(Y = y)} = \Pr(X = x), \\ \Pr(Y = y|X = x) &= \frac{\Pr(X = x, Y = y)}{\Pr(X = x)} = \frac{\Pr(X = x) \cdot \Pr(Y = y)}{\Pr(X = x)} = \Pr(Y = y),\end{aligned}$$

which shows that X and Y are independent.

For continuous random variables, we have the following definition of independence.

Definition 2.8. Let X and Y be continuous random variables. X and Y are independent iff:

$$\begin{aligned}f(x|y) &= f(x), \text{ for } -\infty < x, y < \infty, \\ f(y|x) &= f(y), \text{ for } -\infty < x, y < \infty.\end{aligned}$$

As with discrete random variables, we have the following result for continuous random variables.

Proposition 2.3. *Let X and Y be continuous random variables . X and Y are independent iff:*

$$f(x, y) = f(x)f(y)$$

The result in the above proposition is extremely useful in practice because it gives us an easy way to compute the joint pdf for two independent random variables: we simply compute the product of the marginal distributions.

Example 2.48. Constructing the bivariate standard normal distribution.

Let $X \sim N(0, 1)$, $Y \sim N(0, 1)$ and let X and Y be independent. Then,

$$f(x, y) = f(x)f(y) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2} = \frac{1}{2\pi}e^{-\frac{1}{2}(x^2+y^2)}.$$

This result is a special case of the bivariate normal distribution⁶.

■

A useful property of the independence between two random variables is the following.

Result: If X and Y are independent then $g(X)$ and $h(Y)$ are independent for any functions $g(\cdot)$ and $h(\cdot)$.

For example, if X and Y are independent then X^2 and Y^2 are also independent.

2.2.4 Covariance and correlation

Let X and Y be two discrete random variables. Figure 2.2.4 displays several bivariate probability scatterplots (where equal probabilities are given on the dots). In panel (a) we see no linear relationship between X and Y . In panel (b) we see a perfect positive linear relationship between X and Y and in panel (c) we see a perfect negative linear relationship. In panel (d) we see a positive, but not perfect, linear relationship; in panel (e) we see a negative, but not perfect, linear relationship. Finally, in panel (f) we see no systematic linear relationship but we see a strong nonlinear (parabolic) relationship. The *covariance* between X and Y measures the *direction* of the linear relationship between the two random variables. The *correlation* between X and Y measures the *direction* and the *strength* of the linear relationship between the two random variables.

Let X and Y be two random variables with $E[X] = \mu_X$, $\text{var}(X) = \sigma_X^2$, $E[Y] = \mu_Y$ and $\text{var}(Y) = \sigma_Y^2$.

⁶ stuff to add: if X and Y are independent then $f(X)$ and $g(Y)$ are independent for any functions $f(\cdot)$ and $g(\cdot)$.

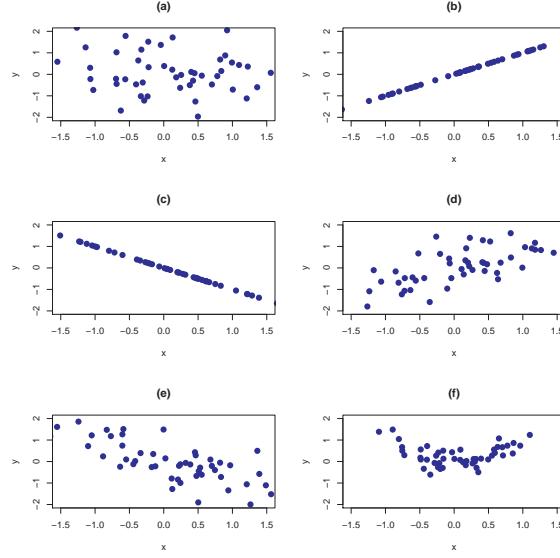


Fig. 2.12 Probability scatterplots illustrating dependence between X and Y .

Definition 2.9. The covariance between two random variables X and Y is given by:

$$\begin{aligned}\sigma_{XY} &= \text{cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \\ &= \sum_{x \in S_X} \sum_{y \in S_Y} (x - \mu_X)(y - \mu_Y) \Pr(X = x, Y = y) \text{ for discrete } X \text{ and } Y, \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_X)(y - \mu_Y) p(x, y) dx dy \text{ for continuous } X \text{ and } Y.\end{aligned}$$

Definition 2.10. The correlation between two random variables X and Y is given by:

$$\rho_{XY} = \text{cor}(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\text{var}(Y)}} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}.$$

The correlation coefficient, ρ_{XY} , is a scaled version of the covariance.

To see how covariance measures the direction of linear association, consider the probability scatterplot in Figure 2.2.4. In the figure, each pair of points occurs with equal probability. The plot is separated into quadrants (right to left, top to bottom). In the first quadrant (black circles), the realized values satisfy $x < \mu_X, y > \mu_Y$ so that the product $(x - \mu_X)(y - \mu_Y) < 0$. In the second quadrant (blue squares), the values satisfy $x > \mu_X$ and $y > \mu_Y$ so that the product $(x - \mu_X)(y - \mu_Y) > 0$. In the third quadrant (red triangles), the values satisfy $x < \mu_X$ and $y < \mu_Y$ so that the product $(x - \mu_X)(y - \mu_Y) > 0$. Finally, in the fourth quadrant (green diamonds), $x > \mu_X$ but $y < \mu_Y$ so that the product $(x - \mu_X)(y - \mu_Y) < 0$. Covariance is then a probability weighted average all of the product terms in the four quadrants. For the values in Figure 2.2.4, this weighted average is positive because most of the values are in the second and third quadrants.

Example 2.49. Calculate covariance and correlation for discrete random variables.

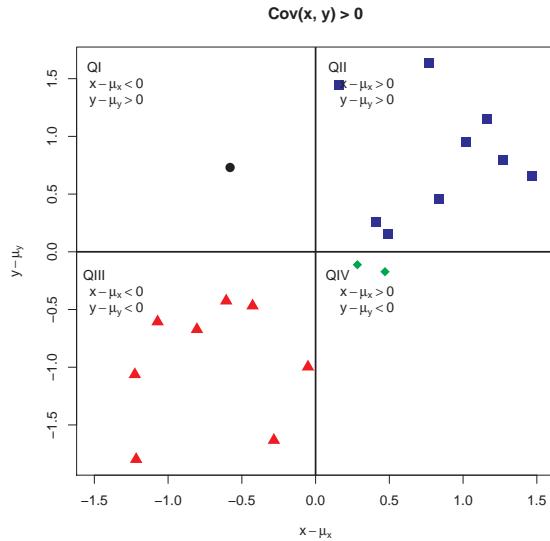


Fig. 2.13 Probability scatterplot of discrete distribution with positive covariance. Each pair (X, Y) occurs with equal probability.

For the data in Table 2.3, we have:

$$\begin{aligned}\sigma_{XY} = \text{cov}(X, Y) &= (0 - 3/2)(0 - 1/2) \cdot 1/8 + (0 - 3/2)(1 - 1/2) \cdot 0 \\ &\quad + \dots + (3 - 3/2)(1 - 1/2) \cdot 1/8 = 1/4 \\ \rho_{XY} = \text{cor}(X, Y) &= \frac{1/4}{\sqrt{(3/4) \cdot (1/2)}} = 0.577\end{aligned}$$

■

Properties of covariance and correlation

Let X and Y be random variables and let a and b be constants. Some important properties of $\text{cov}(X, Y)$ are:

1. $\text{cov}(X, X) = \text{var}(X)$
2. $\text{cov}(X, Y) = \text{cov}(Y, X)$
3. $\text{cov}(X, Y) = E[XY] - E[X]E[Y] = E[XY] - \mu_X\mu_Y$
4. $\text{cov}(aX, bY) = a \cdot b \cdot \text{cov}(X, Y)$
5. If X and Y are independent then $\text{cov}(X, Y) = 0$ (no association \implies no linear association). However, if $\text{cov}(X, Y) = 0$ then X and Y are not necessarily independent (no linear association $\not\implies$ no association).
6. If X and Y are jointly normally distributed and $\text{cov}(X, Y) = 0$, then X and Y are independent.

The first two properties are intuitive. The third property results from expanding the definition of covariance:

$$\begin{aligned}\text{cov}(X, Y) &= E[(X - \mu_X)(Y - \mu_Y)] \\ &= E[XY - X\mu_Y - \mu_X Y + \mu_X\mu_Y] \\ &= E[XY] - E[X]\mu_Y - \mu_X E[Y] + \mu_X\mu_Y \\ &= E[XY] - 2\mu_X\mu_Y + \mu_X\mu_Y \\ &= E[XY] - \mu_X\mu_Y\end{aligned}$$

The fourth property follows from the linearity of expectations:

$$\begin{aligned}\text{cov}(aX, bY) &= E[(aX - a\mu_X)(bY - b\mu_Y)] \\ &= a \cdot b \cdot E[(X - \mu_X)(Y - \mu_Y)] \\ &= a \cdot b \cdot \text{cov}(X, Y)\end{aligned}$$

The fourth property shows that the value of $\text{cov}(X, Y)$ depends on the scaling of the random variables X and Y . By simply changing the scale of X or Y we can make $\text{cov}(X, Y)$ equal to any value that we want. Consequently, the numerical value of $\text{cov}(X, Y)$ is not informative about the strength of the linear association between X and Y . However, the sign of $\text{cov}(X, Y)$ is informative about the direction of linear association between X and Y . The fifth property should be intuitive. Independence between the random variables X and Y means that there is no relationship, linear or nonlinear, between X and Y . However, the lack of a linear relationship between X and Y does not preclude a nonlinear relationship. The last result illustrates an important property of the normal distribution: lack of covariance implies independence.

Some important properties of $\text{cor}(X, Y)$ are:

1. $-1 \leq \rho_{XY} \leq 1$
2. If $\rho_{XY} = 1$ then X and Y are perfectly positively linearly related. That is, $Y = aX + b$ where $a > 0$.
3. If $\rho_{XY} = -1$ then X and Y are perfectly negatively linearly related. That is, $Y = aX + b$ where $a < 0$.
4. If $\rho_{XY} = 0$ then X and Y are not linearly related but may be nonlinearly related.
5. $\text{cor}(aX, bY) = \text{cor}(X, Y)$ if $a > 0$ and $b > 0$; $\text{cor}(X, Y) = -\text{cor}(X, Y)$ if $a > 0, b < 0$ or $a < 0, b > 0$.

2.2.5 Bivariate normal distribution

Let X and Y be distributed bivariate normal. The joint pdf is given by:

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \times \exp\left\{-\frac{1}{2(1-\rho_{XY}^2)} \left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - \frac{2\rho_{XY}(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y} \right]\right\} \quad (2.44)$$

where $E[X] = \mu_X$, $E[Y] = \mu_Y$, $\text{sd}(X) = \sigma_X$, $\text{sd}(Y) = \sigma_Y$, and $\rho_{XY} = \text{cor}(X, Y)$. The correlation coefficient ρ_{XY} describes the dependence between X and Y . If $\rho_{XY} = 0$ then the pdf collapses to the pdf of the standard bivariate normal distribution.

It can be shown that the marginal distributions of X and Y are normal: $X \sim N(\mu_X, \sigma_X^2)$, $Y \sim N(\mu_Y, \sigma_Y^2)$. In addition, it can be shown that the conditional distributions $f(x|y)$ and $f(y|x)$ are also normal with means given by:

$$\mu_{X|Y=y} = \alpha_X + \beta_X \cdot y, \quad (2.45)$$

$$\mu_{Y|X=x} = \alpha_Y + \beta_Y \cdot x, \quad (2.46)$$

where,

$$\begin{aligned} \alpha_X &= \mu_X - \beta_X \mu_X, \quad \beta_X = \sigma_{XY}/\sigma_Y^2, \\ \alpha_Y &= \mu_Y - \beta_Y \mu_X, \quad \beta_Y = \sigma_{XY}/\sigma_X^2, \end{aligned}$$

and variances given by,

$$\begin{aligned} \sigma_{X|Y=y}^2 &= \sigma_X^2 - \sigma_{XY}^2/\sigma_Y^2, \\ \sigma_{Y|X=x}^2 &= \sigma_Y^2 - \sigma_{XY}^2/\sigma_X^2. \end{aligned}$$

Notice that the conditional means (regression functions) (2.29) and (2.30) are linear functions of x and y , respectively.

Example 2.50. Expressing the bivariate normal distribution using matrix algebra.

The formula for the bivariate normal distribution (3.4) is a bit messy. We can greatly simplify the formula by using matrix algebra. Define the 2×1 vectors $\mathbf{x} = (x, y)'$ and $\boldsymbol{\mu} = (\mu_X, \mu_Y)'$, and the 2×2 matrix:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}$$

Then the bivariate normal distribution (3.4) may be compactly expressed as:

$$f(\mathbf{x}) = \frac{1}{2\pi \det(\boldsymbol{\Sigma})^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

where,

$$\det(\boldsymbol{\Sigma}) = \sigma_X^2 \sigma_Y^2 - \sigma_{XY}^2 = \sigma_X^2 \sigma_Y^2 - \sigma_X^2 \sigma_Y^2 \rho_{XY}^2 = \sigma_X^2 \sigma_Y^2 (1 - \rho_{XY}^2).$$

■

Example 2.51. Plotting the bivariate normal distribution.

The R package **mvtnorm** contains the functions `dmvnorm()`, `pmvnorm()`, and `qmvnorm()` which can be used to compute the bivariate normal pdf, cdf and quantiles, respectively. Plotting the bivariate normal distribution over a specified grid of x and y values in R can be done with the `persp()` function. First, we specify the parameter values for the joint distribution. Here, we choose $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$ and $\rho = 0.5$. We will use the `dmvnorm()` function to evaluate the joint pdf at these values. To do so, we must specify a covariance matrix of the form:

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

In R this matrix can be created using:

```
sigma = matrix(c(1, 0.5, 0.5, 1), 2, 2)
```

Next we specify a grid of x and y values between -3 and 3 :

```
x = seq(-3, 3, length = 25)
y = seq(-3, 3, length = 25)
```

To evaluate the joint pdf over the two-dimensional grid we can use the `outer()` function:

```
# function to evaluate bivariate normal pdf on grid
bv.norm <- function(x, y, sigma) {
  z = cbind(x, y)
  return(dmvnorm(z, sigma = sigma))
}
# use outer function to evaluate pdf on 2D grid of x-y values
fxy = outer(x, y, bv.norm, sigma)
```

To create the 3D plot of the joint pdf, use the `persp()` function:

```
persp(x, y, fxy, theta = 60, phi = 30, expand = 0.5, ticktype = "detailed", zlab = "f(x,y)",
      col = "grey")
```

The resulting plot is given in Figure 2.2.5.



2.2.6 Expectation and variance of the sum of two random variables

Let X and Y be two random variables with well defined means, variances and covariance and let a and b be constants. Then the following results hold:

$$\begin{aligned} E[aX + bY] &= aE[X] + bE[Y] = a\mu_X + b\mu_Y \\ \text{var}(aX + bY) &= a^2\text{var}(X) + b^2\text{var}(Y) + 2 \cdot a \cdot b \cdot \text{cov}(X, Y) \\ &= a^2\sigma_X^2 + b^2\sigma_Y^2 + 2 \cdot a \cdot b \cdot \sigma_{XY} \end{aligned}$$

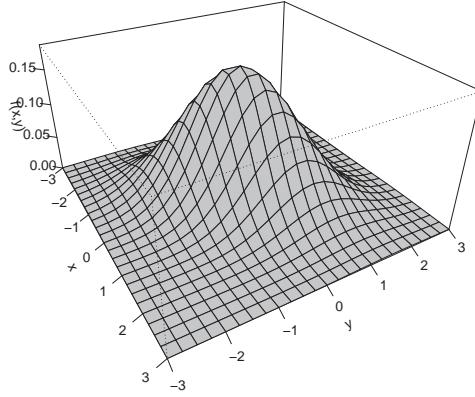


Fig. 2.14 Bivariate normal pdf with $\mu_X = \mu_Y = 0$, $\sigma_X\sigma_Y = 1$ and $\rho = 0.5$.

The first result states that the expected value of a linear combination of two random variables is equal to a linear combination of the expected values of the random variables. This result indicates that the expectation operator is a linear operator. In other words, expectation is additive. The second result states that variance of a linear combination of random variables is not a linear combination of the variances of the random variables. In particular, notice that covariance comes up as a term when computing the variance of the sum of two (not independent) random variables. Hence, the variance operator is not, in general, a linear operator. That is, variance, in general, is not additive.

It is instructive to go through the derivation of these results. Let X and Y be discrete random variables. Then,

$$\begin{aligned}
 E[aX + bY] &= \sum_{x \in S_X} \sum_{y \in S_y} (ax + by) \Pr(X = x, Y = y) \\
 &= \sum_{x \in S_X} \sum_{y \in S_y} ax \Pr(X = x, Y = y) + \sum_{x \in S_X} \sum_{y \in S_y} bx \Pr(X = x, Y = y) \\
 &= a \sum_{x \in S_X} x \sum_{y \in S_y} \Pr(X = x, Y = y) + b \sum_{y \in S_y} y \sum_{x \in S_X} \Pr(X = x, Y = y) \\
 &= a \sum_{x \in S_X} x \Pr(X = x) + b \sum_{y \in S_y} y \Pr(Y = y) \\
 &= aE[X] + bE[Y] = a\mu_X + b\mu_Y.
 \end{aligned}$$

The result for continuous random variables is similar. Effectively, the summations are replaced by integrals and the joint probabilities are replaced by the joint pdf. Next, let X and Y be discrete or continuous random variables. Then,

$$\begin{aligned}
\text{var}(aX + bY) &= E[(aX + bY - E[aX + bY])^2] \\
&= E[(aX + bY - a\mu_X - b\mu_Y)^2] \\
&= E[(a(X - \mu_X) + b(Y - \mu_Y))^2] \\
&= a^2E[(X - \mu_X)^2] + b^2E[(Y - \mu_Y)^2] + 2 \cdot a \cdot b \cdot E[(X - \mu_X)(Y - \mu_Y)] \\
&= a^2\text{var}(X) + b^2\text{var}(Y) + 2 \cdot a \cdot b \cdot \text{cov}(X, Y).
\end{aligned}$$

Linear combination of two normal random variables

The following proposition gives an important result concerning a linear combination of normal random variables.

Proposition 2.4. *Let $X \sim N(\mu_X, \sigma_X^2)$, $Y \sim N(\mu_Y, \sigma_Y^2)$, $\sigma_{XY} = \text{cov}(X, Y)$ and a and b be constants. Define the new random variable Z as:*

$$Z = aX + bY.$$

Then,

$$Z \sim N(\mu_Z, \sigma_Z^2),$$

where $\mu_Z = a\mu_X + b\mu_Y$ and $\sigma_Z^2 = a^2\sigma_X^2 + b^2\sigma_Y^2 + 2ab\sigma_{XY}$.

This important result states that a linear combination of two normally distributed random variables is itself a normally distributed random variable. The proof of the result relies on the change of variables theorem from calculus and is omitted. Not all random variables have the nice property that their distributions are closed under addition.

Example 2.52. Portfolio of two assets.

Consider a portfolio of two stocks A (Amazon) and B (Boeing) with investment shares x_A and x_B with $x_A + x_B = 1$. Let R_A and R_B denote the simple monthly returns on these assets, and assume that $R_A \sim N(\mu_A, \sigma_A^2)$ and $R_B \sim N(\mu_B, \sigma_B^2)$. Furthermore, let $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B = \text{cov}(R_A, R_B)$. The portfolio return is $R_p = x_A R_A + x_B R_B$, which is a linear function of two random variables. Using the properties of linear functions of random variables, we have:

$$\begin{aligned}
\mu_p &= E[R_p] = x_A E[R_A] + x_B E[R_B] = x_A \mu_A + x_B \mu_B \\
\sigma_p^2 &= \text{var}(R_p) = x_A^2 \text{var}(R_A) + x_B^2 \text{var}(R_B) + 2x_A x_B \text{cov}(R_A, R_B) \\
&= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}.
\end{aligned}$$

■

2.3 Multivariate Distributions

Multivariate distributions are used to characterize the joint distribution of a collection of N random variables X_1, X_2, \dots, X_N for $N > 1$. The mathematical formulation of this joint

distribution can be quite complex and typically makes use of matrix algebra. Here, we summarize some basic properties of multivariate distributions without the use of matrix algebra. In chapter 3, we show how matrix algebra can greatly simplify the description of multivariate distributions.

2.3.1 Discrete random variables

Let X_1, X_2, \dots, X_N be N discrete random variables with sample spaces $S_{X_1}, S_{X_2}, \dots, S_{X_N}$. The likelihood that these random variables take values in the joint sample space $S_{X_1} \times S_{X_2} \times \dots \times S_{X_N}$ is given by the joint probability function:

$$p(x_1, x_2, \dots, x_N) = \Pr(X_1 = x_1, X_2 = x_2, \dots, X_N = x_N).$$

For $N > 2$ it is not easy to represent the joint probabilities in a table like Table 2.3 or to visualize the distribution.

Marginal distributions for each variable X_i can be derived from the joint distribution as in (2.25) by summing the joint probabilities over the other variables $j \neq i$. For example,

$$p(x_1) = \sum_{x_2 \in S_{X_2}, \dots, x_N \in S_{X_N}} p(x_1, x_2, \dots, x_N).$$

With N random variables, there are numerous conditional distributions that can be formed. For example, the distribution of X_1 given $X_2 = x_2, \dots, X_N = x_N$ is determined using:

$$\Pr(X_1 = x_1 | X_2 = x_2, \dots, X_N = x_N) = \frac{\Pr(X_1 = x_1, X_2 = x_2, \dots, X_N = x_N)}{\Pr(X_2 = x_2, \dots, X_N = x_N)}.$$

Similarly, the joint distribution of X_1 and X_2 given $X_3 = x_3, \dots, X_N = x_N$ is given by:

$$\Pr(X_1 = x_1, X_2 = x_2 | X_3 = x_3, \dots, X_N = x_N) = \frac{\Pr(X_1 = x_1, X_2 = x_2, \dots, X_N = x_N)}{\Pr(X_3 = x_3, \dots, X_N = x_N)}.$$

2.3.2 Continuous random variables

Let X_1, X_2, \dots, X_N be N continuous random variables each taking values on the real line. The joint pdf is a function $f(x_1, x_2, \dots, x_N) \geq 0$ such that:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, x_2, \dots, x_N) dx_1 dx_2 \cdots dx_N = 1.$$

Joint probabilities of $x_{11} \leq X_1 \leq x_{12}, x_{21} \leq X_2 \leq x_{22}, \dots, x_{N1} \leq X_N \leq x_{N2}$ are computed by solving the integral equation:

$$\int_{x_{11}}^{x_{12}} \int_{x_{21}}^{x_{22}} \cdots \int_{x_{N1}}^{x_{N2}} f(x_1, x_2, \dots, x_N) dx_1 dx_2 \cdots dx_N. \quad (2.47)$$

For most multivariate distributions, the integral in (2.47) cannot be solved analytically and must be approximated numerically.

The marginal pdf for x_i is found by integrating the joint pdf with respect to the other variables. For example, the marginal pdf for x_1 is found by solving:

$$f(x_1) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, x_2, \dots, x_N) dx_2 \cdots dx_N.$$

Conditional pdf for a single random variable or a collection of random variables are defined in the obvious way.

2.3.3 Independence

A collection of N random variables are independent if their joint distribution factors into the product of all of the marginal distributions:

$$\begin{aligned} p(x_1, x_2, \dots, x_N) &= p(x_1)p(x_2) \cdots p(x_N) \text{ for } X_i \text{ discrete,} \\ f(x_1, x_2, \dots, x_N) &= f(x_1)f(x_2) \cdots f(x_N) \text{ for } X_i \text{ continuous.} \end{aligned}$$

In addition, if N random variables are independent then any functions of these random variables are also independent.

2.3.4 Dependence concepts

In general, it is difficult to define dependence concepts for collections of more than two random variables. Dependence is typically only defined between pairwise random variables. Hence, covariance and correlation are also useful concepts when dealing with more than two random variables.

For N random variables X_1, X_2, \dots, X_N , with mean values $\mu_i = E[X_i]$ and variances $\sigma_i^2 = \text{var}(X_i)$, the pairwise covariances and correlations are defined as:

$$\begin{aligned} \text{cov}(X_i, X_j) &= \sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)], \\ \text{cov}(X_i, X_j) &= \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}, \end{aligned}$$

for $i \neq j$. There are $N(N - 1)/2$ pairwise covariances and correlations. Often, these values are summarized using matrix algebra in an $N \times N$ covariance matrix Σ and an $N \times N$ correlation matrix \mathbf{C} :

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}, \quad (2.48)$$

$$\mathbf{C} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1N} \\ \rho_{12} & 1 & \cdots & \rho_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1N} & \rho_{2N} & \cdots & 1 \end{pmatrix}. \quad (2.49)$$

These matrices are formally defined in chapter 3.

2.3.5 Linear combinations of N random variables

Many of the results for manipulating a collection of random variables generalize in a straightforward way to the case of more than two random variables. The details of the generalizations are not important for our purposes. However, the following results will be used repeatedly throughout the book.

Let X_1, X_2, \dots, X_N denote a collection of N random variables (discrete or continuous) with means μ_i , variances σ_i^2 and covariances σ_{ij} . Define the new random variable Z as a linear combination:

$$Z = a_1X_1 + a_2X_2 + \cdots + a_NX_N$$

where a_1, a_2, \dots, a_N are constants. Then the following results hold:

$$\begin{aligned} \mu_Z &= E[Z] = a_1E[X_1] + a_2E[X_2] + \cdots + a_NE[X_N] \\ &= \sum_{i=1}^N a_iE[X_i] = \sum_{i=1}^N a_i\mu_i. \end{aligned} \quad (2.50)$$

$$\begin{aligned} \sigma_Z^2 &= \text{var}(Z) = a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + \cdots + a_N^2\sigma_N^2 \\ &\quad + 2a_1a_2\sigma_{12} + 2a_1a_3\sigma_{13} + \cdots + a_1a_N\sigma_{1N} \\ &\quad + 2a_2a_3\sigma_{23} + 2a_2a_4\sigma_{24} + \cdots + a_2a_N\sigma_{2N} \\ &\quad + \cdots + \\ &\quad + 2a_{N-1}a_N\sigma_{(N-1)N} \\ &= \sum_{i=1}^N a_i^2\sigma_i^2 + 2 \sum_{i=1}^N \sum_{j \neq i} a_i a_j \sigma_{ij}. \end{aligned} \quad (2.51)$$

The derivation of these results is very similar to the bivariate case and so is omitted. In addition, if all of the X_i are normally distributed then Z is also normally distributed with mean μ_Z and variance σ_Z^2 as described above.

The variance of a linear combination of N random variables contains N variance terms and $N(N - 1)$ covariance terms. For $N = 2, 5, 10$ and 100 the number of covariance terms in $\text{var}(Z)$ is $2, 20, 90$ and $9,900$, respectively. Notice that when N is large there are many more covariance terms than variance terms in $\text{var}(Z)$.

The expression for $\text{var}(Z)$ is messy. It can be simplified using matrix algebra notation, as explained in detail in chapter 3. To preview, define the $N \times 1$ vectors $\mathbf{X} = (X_1, \dots, X_N)'$ and $\mathbf{a} = (a_1, \dots, a_N)'$. Then $Z = \mathbf{a}'\mathbf{X}$ and $\text{var}(Z) = \text{var}(\mathbf{a}'\mathbf{X}) = \mathbf{a}'\Sigma\mathbf{a}$, where Σ is the $N \times N$ covariance matrix, which is much more compact than (2.51).

Example 2.53. Square-root-of-time rule for multi-period continuously compounded returns.

Let r_t denote the continuously compounded monthly return on an asset at times $t = 1, \dots, 12$. Assume that r_1, \dots, r_{12} are independent and identically distributed (iid) $N(\mu, \sigma^2)$. Recall, the annual continuously compounded return is equal the sum of twelve monthly continuously compounded returns: $r_A = r(12) = \sum_{t=1}^{12} r_t$. Since each monthly return is normally distributed, the annual return is also normally distributed. The mean of $r(12)$ is:

$$\begin{aligned} E[r(12)] &= E\left[\sum_{t=1}^{12} r_t\right] \\ &= \sum_{t=1}^{12} E[r_t] \quad (\text{by linearity of expectation}), \\ &= \sum_{t=1}^{12} \mu \quad (\text{by identical distributions}), \\ &= 12 \cdot \mu. \end{aligned}$$

Hence, the expected 12-month (annual) return is equal to 12 times the expected monthly return. The variance of $r(12)$ is:

$$\begin{aligned} \text{var}(r(12)) &= \text{var}\left(\sum_{t=1}^{12} r_t\right) \\ &= \sum_{t=1}^{12} \text{var}(r_t) \quad (\text{by independence}), \\ &= \sum_{j=0}^{11} \sigma^2 \quad (\text{by identical distributions}), \\ &= 12 \cdot \sigma^2, \end{aligned}$$

so that the annual variance is also equal to 12 times the monthly variance⁷. Hence, the annual standard deviation is $\sqrt{12}$ times the monthly standard deviation: $\text{sd}(r(12)) = \sqrt{12}\sigma$ (this result is known as the *square-root-of-time rule*). Therefore, $r(12) \sim N(12\mu, 12\sigma^2)$.

■

2.3.6 Covariance between linear combinations of random variables

Consider the linear combinations of two random variables:

$$\begin{aligned} Y &= aX_1 + bX_2, \\ Z &= cX_3 + dX_4, \end{aligned}$$

where a, b, c and d are constants. The covariance between Y and Z is,

$$\begin{aligned} \text{cov}(Y, Z) &= \text{cov}(aX_1 + bX_2, cX_3 + dX_4) \\ &= E[((aX_1 + bX_2) - (a\mu_1 + b\mu_2))((cX_3 + dX_4) - (c\mu_3 + d\mu_4))] \\ &= E[(a(X_1 - \mu_1) + b(X_2 - \mu_2))(c(X_3 - \mu_3) + d(X_4 - \mu_4))] \\ &= acE[(X_1 - \mu_1)(X_3 - \mu_3)] + adE[(X_1 - \mu_1)(X_4 - \mu_4)] \\ &\quad + bcE[(X_2 - \mu_2)(X_3 - \mu_3)] + bdE[(X_2 - \mu_2)(X_4 - \mu_4)] \\ &= accov(X_1, X_3) + adcov(X_1, X_4) + bccov(X_2, X_3) + bdcov(X_2, X_4). \end{aligned}$$

Hence, covariance is additive for linear combinations of random variables. The result above extends in an obvious way to arbitrary linear combinations of random variables.

2.4 Further Reading

This material in this chapter is a selected survey of topics, with applications to finance, typically covered in an introductory probability and statistics course using calculus. A good textbook for such a course is DeGroot and Schervish (2011). Chapter one of Carmona (2014) and the appendix of Ruppert and Matteson (2016) give a slightly more advanced review than is presented here. Two slightly more advanced books, oriented towards economics students, are Amemiya (1994) and Goldberger (1991). Good introductory treatments of probability and statistics using R include Dalgaard (2002) and Verzani (2005). Pfaff (2013) discusses some useful financial asset return distributions and their implementation in R.

⁷ This result often causes some confusion. It is easy to make the mistake and say that the annual variance is $(12)^2 = 144$ time the monthly variance. This result would occur if $r_A = 12r_t$, so that $\text{var}(r_A) = (12)^2\text{var}(r_t) = 144\text{var}(r_t)$.

2.5 Exercises

Exercise 2.1. Suppose X is a normally distributed random variable with mean 0.05 and variance $(0.10)^2$. That is, $X \sim N(0.05, (0.10)^2)$. Use the `pnorm()` and `qnorm()` functions in R to compute the following:

1. $Pr(X \geq 0.10)$, $Pr(X \leq -0.10)$.
2. $Pr(-0.05 \leq X \leq 0.15)$.
3. 1% and 5% quantiles, $q_{0.01}^R$ and $q_{0.05}^R$.
4. 95% and 99% quantiles, $q_{0.95}^R$ and $q_{0.99}^R$.

Exercise 2.2. Let X denote the monthly return on Microsoft Stock and let Y denote the monthly return on Starbucks stock. Assume that $X \sim N(0.05, (0.10)^2)$ and $Y \sim N(0.025, (0.05)^2)$.

1. Using a grid of values between -0.25 and 0.35 , plot the normal curves for X and Y . Make sure that both normal curves are on the same plot.
2. Plot the points $(\sigma_X, \mu_X) = (0.10, 0.05)$ and $(\sigma_Y, \mu_Y) = (0.05, 0.025)$ in an x-y plot.
3. Comment on the risk-return tradeoffs for the two stocks.

Exercise 2.3. Let X_1 , X_2 , Y_1 , and Y_2 be random variables.

1. Show that $\text{Cov}(X_1+X_2, Y_1+Y_2) = \text{Cov}(X_1, Y_1) + \text{Cov}(X_1, Y_2) + \text{Cov}(X_2, Y_1) + \text{Cov}(X_2, Y_2)$.

Exercise 2.4. Let X be a random variable with $\text{var}(X) < \infty$.

1. Show that $\text{var}(X) = E[(X - \mu_X)^2] = E[X^2] - \mu_X^2$.

Exercise 2.5. Let X and Y be a random variables such that $\text{cov}(X, Y) < \infty$.

1. If $Y = aX + b$ where $a > 0$ show that $\rho_{XY} = \text{cor}(X, Y) = 1$.
2. If $Y = aX + b$ where $a < 0$ show that $\rho_{XY} = \text{cor}(X, Y) = -1$.

Exercise 2.6. Let R denote the simple monthly return on Microsoft stock and let W_0 denote initial wealth to be invested over the month. Assume that $R \sim N(0.04, (0.09)^2)$ and that $W_0 = \$100,000$.

1. Determine the 1% and 5% Value-at-Risk (VaR) over the month on the investment. That is, determine the loss in investment value that may occur over the next month with 1% probability and with 5% probability.

Exercise 2.7. Let R_t denote the simple monthly return and assume $R_t \sim \text{iid } N(\mu, \sigma^2)$. Consider the 2-period return $R_t(2) = (1 + R_t)(1 + R_{t-1}) - 1$.

1. Assuming that $\text{cov}(R_t, R_{t-1}) = 0$, show that $E[R_t R_{t-1}] = \mu^2$. Hint: use $\text{cov}(R_t, R_{t-1}) = E[R_t R_{t-1}] - E[R_t]E[R_{t-1}]$.
2. Show that $E[R_t(2)] = (1 + \mu^2) - 1$.

3. Is $R_t(2)$ normally distributed? Justify your answer.

Exercise 2.8. Let r denote the continuously compounded monthly return on Microsoft stock and let W_0 denote initial wealth to be invested over the month. Assume that $r \sim iid N(0.04, (0.09)^2)$ and that $W_0 = \$100,000$.

1. Determine the 1% and 5% value-at-risk (VaR) over the month on the investment. That is, determine the loss in investment value that may occur over the next month with 1% probability and with 5% probability. (Hint: compute the 1% and 5% quantile from the Normal distribution for r and then convert the continuously compounded return quantile to a simple return quantile using the transformation $R = e^r - 1$.)
2. Let $r(12) = r_1 + r_2 + \dots + r_{12}$ denote the 12-month (annual) continuously compounded return return. Compute $E[r(12)]$, $\text{var}(r(12))$ and $\text{sd}(r(12))$. What is the probability distribution of $r(12)$?
3. Using the probability distribution for $r(12)$, determine the 1% and 5% value-at-risk (VaR) over the year on the investment.

Exercise 2.9. Let R_{VFINX} and R_{AAPL} denote the monthly simple returns on VFINX (Vanguard S&P 500 index) and AAPL (Apple stock). Suppose that $R_{VFINX} \sim i.i.d. N(0.013, (0.037)^2)$, $R_{AAPL} \sim i.i.d. N(0.028, (0.073)^2)$.

1. Sketch the normal distributions for the two assets on the same graph. Show the mean values and the ranges within 2 standard deviations of the mean. Which asset appears to be more risky?
2. Plot the risk-return tradeoff for the two assets. That is, plot the mean values of each asset on the y-axis and the standard deviations on the x-axis. Comment on the relationship.
3. Let $W_0 = \$1,000$ be the initial wealth invested in each asset. Compute the 1% monthly Value-at-Risk values for each asset. (Hint: $q_{0.01}^Z = -2.326$).
4. Continue with the above question, state in words what the 1% Value-at-Risk numbers represent (i.e., explain what 1% Value-at-Risk for a one month \\$1,000 investment means).
5. The normal distribution can be used to characterize the probability distribution of monthly simple returns or monthly continuously compounded returns. What are two problems with using the normal distribution for simple returns? Given these two problems, why might it be better to use the normal distribution for continuously compounded returns?

Exercise 2.10. In this question, you will examine the chi-square and Student's t distributions. A chi-square random variable with n degrees of freedom, denoted $X \sim \chi_n^2$, is defined as $X = Z_1^2 + Z_2^2 + \dots + Z_n^2$ where Z_1, Z_2, \dots, Z_n are $iid N(0, 1)$ random variables. Notice that X only takes positive values. A Student's t random variable with n degrees of freedom, denoted $t \sim t_n$, is defined as $t = Z / \sqrt{X/n}$, where $Z \sim N(0, 1)$, $X \sim \chi_n^2$ and Z is independent of X . The Student's t distribution is similar to the standard normal except that it has fatter tails.

1. On the same graph, plot the probability curves of chi-squared distributed random variables with 1, 2, 5 and 10 degrees of freedom. Use different colors and line styles for each curve. Hint: In R the density of the chi-square distribution is computed using the function `dchisq()`.

2. On the same graph, plot the probability curves of Student's t distributed random variables with 1, 2, 5 and 10 degrees of freedom. Also include the probability curve for the standard normal distribution. Use different colors and line styles for each curve. Hint: In R the density of the chi-square distribution is computed using the function `dt()`.

Exercise 2.11. Consider the following joint distribution of X and Y :

X/Y	1	2	3
1	0.1	0.2	0
2	0.1	0	0.2
3	0	0.1	0.3

1. Find the marginal distributions of X and Y . Using these distributions, compute $E[X]$, $\text{var}(X)$, $\text{sd}(X)$, $E[Y]$, $\text{var}(Y)$ and $\text{sd}(Y)$.
2. Compute $\text{cov}(X, Y)$ and $\text{cor}(X, Y)$.
3. Find the conditional distributions $\Pr(X|Y = y)$ for $y = 1, 2, 3$ and the conditional distributions $\Pr(Y|X = x)$ for $x = 1, 2, 3$.
4. Using the conditional distributions $\Pr(X|Y = y)$ for $y = 1, 2, 3$ and $\Pr(Y|X = x)$ for $x = 1, 2, 3$ compute $E[X|Y = y]$ and $E[Y|X = x]$.
5. Using the conditional distributions $\Pr(X|Y = y)$ for $y = 1, 2, 3$ and $\Pr(Y|X = x)$ for $x = 1, 2, 3$ compute $\text{var}[X|Y = y]$ and $\text{var}[Y|X = x]$.
6. Are X and Y independent? Fully justify your answer.

Exercise 2.12. Let X and Y be distributed bivariate normal with $\mu_X = 0.05$, $\mu_Y = 0.025$, $\sigma_X = 0.10$, and $\sigma_Y = 0.05$.

1. Using R package **mvtnorm** function `rmvnorm()`, simulate 100 observations from the bivariate normal distribution with $\rho_{XY} = 0.9$. Using the `plot()` function create a scatterplot of the observations and comment on the direction and strength of the linear association. Using the function `pmvnorm()`, compute the joint probability $Pr(X \leq 0, Y \leq 0)$.
2. Using R package **mvtnorm** function `rmvnorm()`, simulate 100 observations from the bivariate normal distribution with $\rho_{XY} = -0.9$. Using the `plot()` function create a scatterplot of the observations and comment on the direction and strength of the linear association. Using the function `pmvnorm()`, compute the joint probability $Pr(X \leq 0, Y \leq 0)$.
3. Using R package **mvtnorm** function `rmvnorm()`, simulate 100 observations from the bivariate normal distribution with $\rho_{XY} = 0$. Using the `plot()` function create a scatterplot of the observations and comment on the direction and strength of the linear association. Using the function `pmvnorm()`, compute the joint probability $Pr(X \leq 0, Y \leq 0)$.

References

1. Amemiya, T. (1994). *Introduction to Statistics and Econometrics*. Harvard University Press, Cambridge, MA.

2. Carmona, R. (2014). *Statistical Analysis of Financial Data in R, Second Edition*. Springer, New York.
3. Dalgaard, P. (2002). *Introductory Statistics with R*. Springer.
4. DeGroot, M. H., and Schervish, M. J. (2011). *Probability and Statistics*, 4th Edition. Pearson.
5. Goldberger, A.S. (1991). *A Course in Econometrics*. Harvard University Press, Cambridge, MA.
6. Pfaff, B. (2013). *Financial Risk Modelling and Portfolio Optimization with R*. Wiley, Chichester, UK.
7. Ruppert, D., and Matteson, D.S. (2015). *Statistics and Data Analysis for Financial Engineering with R examples*. Springer, New York.
8. Verzani, J. (2005). *Using R for Introductory Statistics*. Chapman & Hall/CRC.

Chapter 3

Matrix Algebra Review

Updated: July 7, 2016

Copyright © Eric Zivot 2015

This chapter reviews some basic matrix algebra concepts that we will use throughout the book. Many of the formulas that involve multiple variables that we derive in the book can be simplified using matrix algebra. This aids in understanding general concepts. For example, portfolio risk and return calculations for portfolios with more than two assets is easy to express using matrix algebra. If returns are jointly normally distributed, then this multivariate normal distribution can be expressed concisely using matrix algebra. The first order conditions involving the portfolio weights for mean-variance optimized portfolios is a system of linear equations that can be set-up and solved using matrix algebra. In addition, R is a matrix programming language. Pretty much anything that you can represent in matrix algebra you can evaluate in R using matrix calculations that have a syntax very much like the matrix algebra representation. Lastly, many R calculations can be efficiently evaluated if they are “vectorized” - that is, if they operate on vectors of elements instead of looping over individual elements.

This chapter is organized as follows. Section 1 gives definitions of matrices and vectors and shows how to represent these objects in R. Section 2 reviews basic matrix operations (addition, subtraction, multiplication), and section 3 shows how common summation calculations can be represented using matrix notation. Section 4 covers systems of linear equations and shows how the inverse of a matrix is related to the solutions to a system of linear equations. Section 5 discusses positive definite matrices and Cholesky decomposition. The use of matrix algebra for representing multivariate probability distributions is presented in section 6, and portfolio calculations using matrix algebra is presented in section 7. Section 8 concludes with a discussion of derivatives of some simple matrix functions.

The R packages used in this chapter are **Matrix** and **mvtnorm**. Make sure these packages are downloaded and installed prior to replicating the R examples in this chapter.

3.1 Matrices and Vectors

A *matrix* is just an array of numbers. The *dimension* of a matrix is determined by the number of its rows and columns. For example, a matrix \mathbf{A} with n rows and m columns is illustrated below:

$$\underset{(n \times m)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

where a_{ij} denotes the i^{th} row and j^{th} column element of \mathbf{A} .

A *vector* is simply a matrix with 1 column. For example,

$$\underset{(n \times 1)}{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

is an $n \times 1$ vector with elements x_1, x_2, \dots, x_n . Vectors and matrices are often written in bold type (or underlined) to distinguish them from scalars (single elements of vectors or matrices).

Example 3.1. Matrix creation in R.

In R, matrix objects are created using the `matrix()` function. For example, to create the 2×3 matrix:

$$\underset{(2 \times 3)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

use,

```
matA = matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
matA

##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6

class(matA)

## [1] "matrix"
```

The optional argument `byrow=TRUE` fills the matrix row by row.¹ The default is `byrow=FALSE` which fills the matrix column by column:

```
matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Matrix objects (i.e., objects of class “`matrix`”) have row and column dimension attributes which can be examined with the `dim()` function:

```
dim(matA)

## [1] 2 3
```

The rows and columns can be given names using:

```
dimnames(matA) = list(c("row1", "row2"), c("col1", "col2", "col3"))

matA

##      col1 col2 col3
## row1    1    2    3
## row2    4    5    6
```

or,

```
colnames(matA) = c("col1", "col2", "col3")
rownames(matA) = c("row1", "row2")

matA

##      col1 col2 col3
## row1    1    2    3
## row2    4    5    6
```

The elements of a matrix can be extracted or subsetted as follows:

```
matA[1, 2]

## [1] 2

matA["row1", "col1"]

## [1] 1

matA[1, ]

## col1 col2 col3
##    1    2    3

matA[, 2]

## row1 row2
##    2    5
```

¹ When specifying logical variables in R always spell out `TRUE` and `FALSE` instead of using `T` and `F`. Upon startup R defines the variables `T=TRUE` and `F=FALSE` so that `T` and `F` can be used as substitutes for `TRUE` and `FALSE`, respectively. However, this shortcut is not recommended because the variables `T` and `F` could be reassigned during subsequent programming.

To preserve the dimension attributes when subsetting, use the `drop=FALSE` option:

```
matA[1, , drop = FALSE]

##      col1 col2 col3
## row1    1    2    3

matA[, 2, drop = FALSE]

##      col2
## row1    2
## row2    5
```



Example 3.2. Creating vectors in R.

Vectors can be created in R using a variety of methods. The easiest way is to use the combine function `c()`:

```
xvec = c(1, 2, 3)
xvec

## [1] 1 2 3

xvec = 1:3
xvec

## [1] 1 2 3

xvec = seq(from = 1, to = 3, by = 1)
xvec

## [1] 1 2 3
```

Vectors of numbers in R are of class “`numeric`” and do not have a dimension attribute:

```
class(xvec)

## [1] "numeric"

dim(xvec)

## NULL
```

The elements of a vector can be assigned names using the `names()` function:

```
names(xvec) = c("x1", "x2", "x3")
xvec

## x1 x2 x3
## 1 2 3
```

To force a dimension attribute onto a vector, coerce it to a “`matrix`” object using `as.matrix()`:

```
as.matrix(xvec)

##      [,1]
## x1     1
## x2     2
## x3     3
```

■

The *transpose* of an $n \times m$ matrix \mathbf{A} is a new matrix with the rows and columns of \mathbf{A} interchanged, and is denoted \mathbf{A}' or \mathbf{A}^\top . For example,

$$\mathbf{A}_{(2 \times 3)} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad \mathbf{A}'_{(3 \times 2)} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

$$\mathbf{x}_{(3 \times 1)} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{x}'_{(1 \times 3)} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}.$$

A *symmetric* matrix \mathbf{A} is such that $\mathbf{A} = \mathbf{A}'$. Obviously, this can only occur if \mathbf{A} is a *square* matrix; i.e., the number of rows of \mathbf{A} is equal to the number of columns. For example, consider the 2×2 matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Then,

$$\mathbf{A}' = \mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}.$$

Example 3.3. Transpose of a matrix in R.

To take the transpose of a matrix or vector, use the `t()` function:

```
matA = matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
matA

##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6

t(matA)

##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```
xvec = c(1, 2, 3)
t(xvec)

##      [,1] [,2] [,3]
## [1,]    1    2    3
```

Notice that, when applied to a “*numeric*” vector with n elements, the `t()` function returns a “*matrix*” object with dimension $1 \times n$.

■

3.2 Basic Matrix Operations

In this section we review the basic matrix operations of addition, subtraction, scalar multiplication and multiplication.

3.2.1 Addition and subtraction

Matrix addition and subtraction are element by element operations and only apply to matrices of the same dimension. For example, let,

$$\mathbf{A} = \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix}.$$

Then,

$$\begin{aligned} \mathbf{A} + \mathbf{B} &= \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 4+2 & 9+0 \\ 2+0 & 1+7 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ 2 & 8 \end{bmatrix}, \\ \mathbf{A} - \mathbf{B} &= \begin{bmatrix} 4 & 9 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 4-2 & 9-0 \\ 2-0 & 1-7 \end{bmatrix} = \begin{bmatrix} 2 & 9 \\ 2 & -6 \end{bmatrix}. \end{aligned}$$

Example 3.4. Matrix addition and subtraction in R.

Matrix addition and subtraction is straightforward in R:

```
matA = matrix(c(4, 9, 2, 1), 2, 2, byrow = TRUE)
matB = matrix(c(2, 0, 0, 7), 2, 2, byrow = TRUE)
matA

##      [,1] [,2]
## [1,]    4    9
## [2,]    2    1
```

```

matB

##      [,1] [,2]
## [1,]    2    0
## [2,]    0    7

# matrix addition
matC = matA + matB
matC

##      [,1] [,2]
## [1,]    6    9
## [2,]    2    8

# matrix subtraction
matD = matA - matB
matD

##      [,1] [,2]
## [1,]    2    9
## [2,]    2   -6

```



3.2.2 Scalar multiplication

Here we refer to the multiplication of a matrix by a scalar number. This is also an element-by-element operation. For example, let $c = 2$ and,

$$\mathbf{A} = \begin{bmatrix} 3 & -1 \\ 0 & 5 \end{bmatrix}.$$

Then,

$$c \cdot \mathbf{A} = \begin{bmatrix} 2 \cdot 3 & 2 \cdot (-1) \\ 2 \cdot (0) & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ 0 & 10 \end{bmatrix}.$$

Example 3.5. Scalar multiplication in R.

```

matA = matrix(c(3, -1, 0, 5), 2, 2, byrow = TRUE)
matC = 2 * matA
matC

##      [,1] [,2]
## [1,]    6   -2
## [2,]    0   10

```



3.2.3 Matrix multiplication

Matrix multiplication only applies to *conformable* matrices. \mathbf{A} and \mathbf{B} are conformable matrices if the number of columns in \mathbf{A} is equal to the number of rows in \mathbf{B} . For example, if \mathbf{A} is $n \times m$ and \mathbf{B} is $m \times p$ then \mathbf{A} and \mathbf{B} are conformable and the matrix product of \mathbf{A} and \mathbf{B} has dimension $n \times p$. The mechanics of matrix multiplication is best explained by example. Let,

$$\underset{(2 \times 2)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } \underset{(2 \times 3)}{\mathbf{B}} = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 2 \end{bmatrix}.$$

Then,

$$\begin{aligned} \underset{(2 \times 2)}{\mathbf{A}} \cdot \underset{(2 \times 3)}{\mathbf{B}} &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 1 \cdot 1 + 2 \cdot 3 & 1 \cdot 2 + 2 \cdot 4 & 1 \cdot 1 + 2 \cdot 2 \\ 3 \cdot 1 + 4 \cdot 3 & 3 \cdot 2 + 4 \cdot 4 & 3 \cdot 1 + 4 \cdot 2 \end{bmatrix} \\ &= \begin{bmatrix} 7 & 10 & 5 \\ 15 & 22 & 11 \end{bmatrix} = \underset{(2 \times 3)}{\mathbf{C}} \end{aligned}$$

The resulting matrix \mathbf{C} has 2 rows and 3 columns. The $(1, 1)$ element of \mathbf{C} is the *dot product* of the first row of \mathbf{A} with the first column of \mathbf{B} :

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} = 1 \cdot 1 + 2 \cdot 3 = 7.$$

The $(1, 2)$ element of \mathbf{C} is the *dot product* of the first row of \mathbf{A} with the second column of \mathbf{B} :

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 4 \end{pmatrix} = 1 \cdot 2 + 2 \cdot 4 = 10.$$

In general, the (i, j) element of \mathbf{C} is the dot product of the i th row of \mathbf{A} with the j th column of \mathbf{B} . If \mathbf{A} is $n \times m$ and \mathbf{B} is $m \times p$ then $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ is $n \times p$.

As another example, let,

$$\underset{(2 \times 2)}{\mathbf{A}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } \underset{(2 \times 1)}{\mathbf{B}} = \begin{bmatrix} 2 \\ 6 \end{bmatrix}.$$

Then,

$$\begin{aligned}
 \underset{(2 \times 2)}{\mathbf{A}} \cdot \underset{(2 \times 1)}{\mathbf{B}} &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 6 \end{bmatrix} \\
 &= \begin{bmatrix} 1 \cdot 2 + 2 \cdot 6 \\ 3 \cdot 2 + 4 \cdot 6 \end{bmatrix} \\
 &= \begin{bmatrix} 14 \\ 30 \end{bmatrix}.
 \end{aligned}$$

As a final example, let,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}.$$

Then

$$\mathbf{x}'\mathbf{y} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = 1 \cdot 4 + 2 \cdot 5 + 3 \cdot 6 = 32$$

Example 3.6. Matrix multiplication in R.

In R, matrix multiplication is performed with the `%*%` operator. For example:

```

matA = matrix(1:4, 2, 2, byrow = TRUE)
matB = matrix(c(1, 2, 1, 3, 4, 2), 2, 3, byrow = TRUE)
matA

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4

matB

##      [,1] [,2] [,3]
## [1,]    1    2    1
## [2,]    3    4    2

dim(matA)

## [1] 2 2

dim(matB)

## [1] 2 3

matC = matA %*% matB
matC

```

```

##      [,1] [,2] [,3]
## [1,]    7   10    5
## [2,]   15   22   11

# note: A%*%B is generally not equal to B%*%A here B%*%A doesn't work b/c A and
# B are not conformable
matB %*% matA

## Error in matB %*% matA: non-conformable arguments

```

The next example shows that matrix multiplication in R also works on numeric vectors:

```

matA = matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE)
vecB = c(2, 6)
matA %*% vecB

##      [,1]
## [1,]   14
## [2,]   30

vecX = c(1, 2, 3)
vecY = c(4, 5, 6)
t(vecX) %*% vecY

##      [,1]
## [1,]   32

crossprod(vecX, vecY)

##      [,1]
## [1,]   32

```

Miscellaneous properties

Matrix multiplication satisfies the associative property: Let A , B and C be conformable matrices. Then

$$A(B + C) = AB + AC.$$

Also, the transpose of the product of two matrices is the product of the transposes in opposite order:

$$(AB)' = B'A'.$$

3.2.4 The identity matrix

The identity matrix plays a similar role as the number 1. Multiplying any number by 1 gives back that number. In matrix algebra, pre-multiplying or post-multiplying a matrix \mathbf{A} by a conformable identity matrix gives back the matrix \mathbf{A} . To illustrate, let

$$\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

denote the 2 dimensional identity matrix and let

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

denote an arbitrary 2×2 matrix. Then,

$$\begin{aligned} \mathbf{I}_2 \cdot \mathbf{A} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}, \end{aligned}$$

and,

$$\begin{aligned} \mathbf{A} \cdot \mathbf{I}_2 &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}. \end{aligned}$$

Example 3.7. The identity matrix in R.

Use the `diag()` function to create an identity matrix:

```
matI = diag(2)
matI

##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1

matA = matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE)
matI %*% matA

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4

matA %*% matI

##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

■

3.2.5 Diagonal, lower triangular and upper triangular matrices

Consider an $n \times n$ matrix \mathbf{A}

$$\mathbf{A} = \begin{bmatrix} d_1 & u_{12} & u_{13} & \cdots & u_{1n} \\ l_{21} & d_2 & u_{23} & \cdots & u_{2n} \\ l_{31} & l_{32} & d_3 & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & d_n \end{bmatrix}.$$

The main diagonal consists of the n elements $\{d_1, d_2, \dots, d_n\}$. The lower triangle consists of the $n(n-1)/2$ elements $\{l_{21}, l_{31}, l_{32}, \dots, l_{n(n-1)}\}$ below the main diagonal, and the upper triangle consists of the $n(n-1)/2$ elements $\{u_{12}, u_{13}, \dots, u_{(n-1)n}\}$ above the main diagonal.

An n -dimensional *diagonal matrix* \mathbf{D} is an $n \times n$ square matrix with an $n \times 1$ vector $\mathbf{d} = (d_1, \dots, d_n)'$ along the main diagonal and zeros elsewhere:

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ 0 & 0 & d_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_n \end{bmatrix}.$$

An $n \times n$ *lower triangular matrix* \mathbf{L} has all values above the main diagonal equal to zero:

$$\mathbf{L} = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ l_{21} & d_2 & 0 & \cdots & 0 \\ l_{31} & l_{32} & d_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & d_n \end{bmatrix}.$$

An $n \times n$ *upper triangular matrix* \mathbf{U} has all values below the main diagonal equal to zero:

$$\mathbf{A} = \begin{bmatrix} d_1 & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & d_2 & u_{23} & \cdots & u_{2n} \\ 0 & 0 & d_3 & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & d_n \end{bmatrix}.$$

Example 3.8. Creating a diagonal matrix in R

Diagonal matrices can be created with the `diag()` function. To create a 3×3 diagonal matrix with $d = (1, 2, 3)$ along the main diagonal, use

```
matD = diag(1:3)
matD

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    2    0
## [3,]    0    0    3
```

Example 3.9. Extracting lower and upper triangular elements of a square matrix

The R functions `lower.tri()` and `upper.tri()` can be used to extract the lower and upper triangular parts of a square matrix:

```
matA = matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9), 3, 3)
matA

##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9

lower.tri(matA)

##      [,1] [,2] [,3]
## [1,] FALSE FALSE FALSE
## [2,] TRUE  FALSE FALSE
## [3,] TRUE  TRUE  FALSE

upper.tri(matA)

##      [,1] [,2] [,3]
## [1,] FALSE TRUE  TRUE
## [2,] FALSE FALSE TRUE
## [3,] FALSE FALSE FALSE

matA[lower.tri(matA)]

## [1] 2 3 6

matA[upper.tri(matA)]

## [1] 4 7 8
```

■

3.3 Representing Summation Using Matrix Notation

Consider the sum:

$$\sum_{k=1}^n x_k = x_1 + \cdots + x_n.$$

Let $\mathbf{x} = (x_1, \dots, x_n)'$ be an $n \times 1$ vector and $\mathbf{1} = (1, \dots, 1)'$ be an $n \times 1$ vector of ones. Then,

$$\mathbf{x}'\mathbf{1} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = x_1 + \cdots + x_n = \sum_{k=1}^n x_k,$$

and,

$$\mathbf{1}'\mathbf{x} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 + \cdots + x_n = \sum_{k=1}^n x_k.$$

Next, consider the sum of squared x values,

$$\sum_{k=1}^n x_k^2 = x_1^2 + \cdots + x_n^2.$$

This sum can be conveniently represented as,

$$\mathbf{x}'\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1^2 + \cdots + x_n^2 = \sum_{k=1}^n x_k^2.$$

Last, consider the sum of cross products,

$$\sum_{k=1}^n x_k y_k = x_1 y_1 + \cdots + x_n y_n.$$

This sum can be compactly represented by,

$$\mathbf{x}'\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = x_1 y_1 + \cdots + x_n y_n = \sum_{k=1}^n x_k y_k.$$

Note that $\mathbf{x}'\mathbf{y} = \mathbf{y}'\mathbf{x}$.

Example 3.10. Computing sums in R.

In R, summing the elements in a vector can be done using matrix algebra.

```
# create vector of 1's and a vector of x
onevec = rep(1, 3)
onevec

## [1] 1 1 1

xvec = c(1, 2, 3)
# sum elements in x
t(xvec) %*% onevec

##      [,1]
## [1,]    6
```

The functions `crossprod()` and `sum()` are generally computationally more efficient:

```
crossprod(xvec, onevec)

##      [,1]
## [1,]    6

sum(xvec)

## [1] 6
```

Sums of squares are best computed using:

```
crossprod(xvec)

##      [,1]
## [1,]    14

sum(xvec^2)

## [1] 14
```

The dot-product two vectors can be conveniently computed using the `crossprod()` function:

```
yvec = 4:6
xvec

## [1] 1 2 3

yvec

## [1] 4 5 6

crossprod(xvec, yvec)

##      [,1]
## [1,]    32
```

```
crossprod(yvec, xvec)

##      [,1]
## [1,] 32
```

■

3.4 Systems of Linear Equations

Consider the system of two linear equations:

$$x + y = 1, \quad (3.1)$$

$$2x - y = 1. \quad (3.2)$$

As shown in Figure 3.4, equations (3.1) and (3.2) represent two straight lines which intersect at the point $x = 2/3$ and $y = 1/3$. This point of intersection is determined by solving for the values of x and y such that $x + y = 2x - y$.²

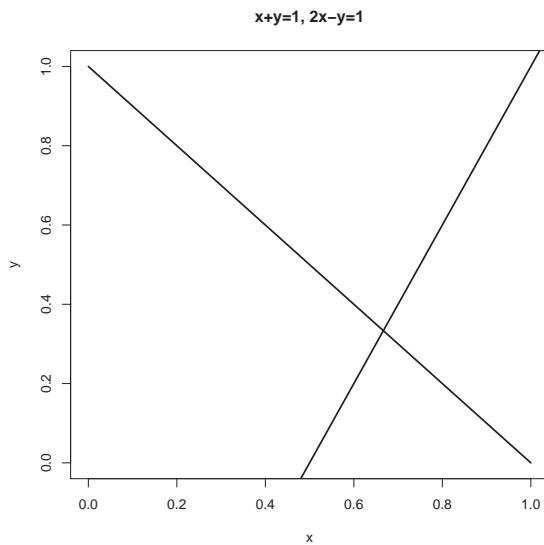


Fig. 3.1 System of two linear equations.

The two linear equations can be written in matrix form as:

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

² Solving for x gives $x = 2y$. Substituting this value into the equation $x + y = 1$ gives $2y + y = 1$ and solving for y gives $y = 1/3$. Solving for x then gives $x = 2/3$.

or,

$$\mathbf{A} \cdot \mathbf{z} = \mathbf{b},$$

where,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

If there was a (2×2) matrix \mathbf{B} , with elements b_{ij} , such that $\mathbf{B} \cdot \mathbf{A} = \mathbf{I}_2$, where \mathbf{I}_2 is the (2×2) identity matrix, then we could *solve* for the elements in \mathbf{z} as follows. In the equation $\mathbf{A} \cdot \mathbf{z} = \mathbf{b}$, pre-multiply both sides by \mathbf{B} to give:

$$\begin{aligned} \mathbf{B} \cdot \mathbf{A} \cdot \mathbf{z} &= \mathbf{B} \cdot \mathbf{b} \\ \implies \mathbf{I} \cdot \mathbf{z} &= \mathbf{B} \cdot \mathbf{b} \\ \implies \mathbf{z} &= \mathbf{B} \cdot \mathbf{b}, \end{aligned}$$

or

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} \cdot 1 + b_{12} \cdot 1 \\ b_{21} \cdot 1 + b_{22} \cdot 1 \end{bmatrix}.$$

If such a matrix \mathbf{B} exists it is called the *inverse* of \mathbf{A} and is denoted \mathbf{A}^{-1} . Intuitively, the inverse matrix \mathbf{A}^{-1} plays a similar role as the inverse of a number. Suppose a is a number; e.g., $a = 2$. Then we know that $1/a \cdot a = a^{-1}a = 1$. Similarly, in matrix algebra $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_2$ where \mathbf{I}_2 is the identity matrix. Now, consider solving the equation $a \cdot x = 1$. By simple division we have that $x = (1/a)x = a^{-1}x$. Similarly, in matrix algebra, if we want to solve the system of linear equations $\mathbf{Ax} = \mathbf{b}$ we pre-multiply by \mathbf{A}^{-1} and get the solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

Using $\mathbf{B} = \mathbf{A}^{-1}$, we may express the solution for \mathbf{z} as:

$$\mathbf{z} = \mathbf{A}^{-1}\mathbf{b}.$$

As long as we can determine the elements in \mathbf{A}^{-1} then we can solve for the values of x and y in the vector \mathbf{z} . The system of linear equations has a solution as long as the two lines intersect, so we can determine the elements in \mathbf{A}^{-1} provided the two lines are not parallel. If the two lines are parallel, then one of the equations is a multiple of the other. In this case we say that \mathbf{A} is *not invertible*.

There are general numerical algorithms for finding the elements of \mathbf{A}^{-1} (e.g., Gaussian elimination) and matrix programming languages and spreadsheets have these algorithms available. However, if \mathbf{A} is a (2×2) matrix then there is a simple formula for \mathbf{A}^{-1} . Let \mathbf{A} be a (2×2) matrix such that:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}.$$

Then,

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix},$$

where $\det(\mathbf{A}) = a_{11}a_{22} - a_{21}a_{12}$ denotes the determinant of \mathbf{A} and is assumed to be not equal to zero. By brute force matrix multiplication we can verify this formula:

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{A} &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22}a_{11} - a_{12}a_{21} & a_{22}a_{12} - a_{12}a_{22} \\ -a_{21}a_{11} + a_{11}a_{21} & -a_{21}a_{12} + a_{11}a_{22} \end{bmatrix} \\ &= \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{bmatrix} a_{22}a_{11} - a_{12}a_{21} & 0 \\ 0 & -a_{21}a_{12} + a_{11}a_{22} \end{bmatrix} \\ &= \begin{bmatrix} \frac{a_{22}a_{11} - a_{12}a_{21}}{a_{11}a_{22} - a_{21}a_{12}} & 0 \\ 0 & \frac{-a_{21}a_{12} + a_{11}a_{22}}{a_{11}a_{22} - a_{21}a_{12}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Let's apply the above rule to find the inverse of \mathbf{A} in our example linear system (3.1)-(3.2):

$$\mathbf{A}^{-1} = \frac{1}{-1 - 2} \begin{bmatrix} -1 & -1 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix}.$$

Notice that,

$$\mathbf{A}^{-1}\mathbf{A} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Our solution for \mathbf{z} is then,

$$\begin{aligned} \mathbf{z} &= \mathbf{A}^{-1}\mathbf{b} \\ &= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{-1}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}, \end{aligned}$$

so that $x = 2/3$ and $y = 1/3$.

Example 3.11. Solving systems of linear equations in R.

In R, the `solve()` function is used to compute the inverse of a matrix and solve a system of linear equations. The linear system $x + y = 1$ and $2x - y = 1$ can be represented using:

```
matA = matrix(c(1, 1, 2, -1), 2, 2, byrow = TRUE)
vecB = c(1, 1)
```

First we solve for \mathbf{A}^{-1} .³

```
matA.inv = solve(matA)
matA.inv

##      [,1]     [,2]
## [1,] 0.3333  0.3333
## [2,] 0.6667 -0.3333

matA.inv %*% matA

##      [,1]     [,2]
## [1,]    1 -5.551e-17
## [2,]    0  1.000e+00

matA %*% matA.inv

##      [,1]     [,2]
## [1,]    1 5.551e-17
## [2,]    0  1.000e+00
```

Then we solve the system $\mathbf{z} = \mathbf{A}^{-1}\mathbf{b}$:

```
z = matA.inv %*% vecB
z

##      [,1]
## [1,] 0.6667
## [2,] 0.3333
```

In general, if we have n linear equations in n unknown variables we may write the system of equations as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots = \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

which we may then express in matrix form as,

³ Notice that the calculations in R do not show $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ exactly. The (1,2) element of $\mathbf{A}^{-1}\mathbf{A}$ is -5.552e-17, which for all practical purposes is zero. However, due to the limitations of machine calculations the result is not exactly zero.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

or,

$$\underset{(n \times n)}{\mathbf{A}} \cdot \underset{(n \times 1)}{\mathbf{x}} = \underset{(n \times 1)}{\mathbf{b}}.$$

The solution to the system of equations is given by:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b},$$

where $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ and \mathbf{I} is the $(n \times n)$ identity matrix. If the number of equations is greater than two, then we generally use numerical algorithms to find the elements in \mathbf{A}^{-1} .

3.4.1 Rank of a matrix

A $n \times m$ matrix \mathbf{A} has m columns $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ where each column is an $n \times 1$ vector and n rows where each row is an $1 \times m$ row vector:

$$\underset{(n \times m)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_m \end{bmatrix}.$$

Two vectors \mathbf{a}_1 and \mathbf{a}_2 are linearly independent if $c_1\mathbf{a}_1 + c_2\mathbf{a}_2 = 0$ implies that $c_1 = c_2 = 0$. A set of vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ are linearly independent if $c_1\mathbf{a}_1 + c_2\mathbf{a}_2 + \cdots + c_m\mathbf{a}_m = 0$ implies that $c_1 = c_2 = \cdots = c_m = 0$. That is, no vector \mathbf{a}_i can be expressed as a non-trivial linear combination of the other vectors.

The *column rank* of the $n \times m$ matrix \mathbf{A} , denoted $\text{rank}(\mathbf{A})$, is equal to the maximum number of linearly independent columns. The row rank of a matrix is equal to the maximum number of linearly independent rows, and is given by $\text{rank}(\mathbf{A}')$. It turns out that the column rank and row rank of a matrix are the same. Hence, $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}') \leq \min(m, n)$. If $\text{rank}(\mathbf{A}) = m$ then \mathbf{A} is called *full column rank*, and if $\text{rank}(\mathbf{A}) = n$ then \mathbf{A} is called *full row rank*. If \mathbf{A} is not full rank then it is called reduced rank.

Example 3.12. Determining the rank of a matrix in R.

Consider the 2×3 matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Here, $\text{rank}(\mathbf{A}) \leq \min(3, 2) = 2$, the number of rows. Now, $\text{rank}(\mathbf{A}) = 2$ since the rows of \mathbf{A} are linearly independent. In R, the rank of a matrix can be found using the **Matrix** function **rankMatrix()**:

```
library(Matrix)
Amat = matrix(c(1, 3, 5, 2, 4, 6), 2, 3, byrow = TRUE)
as.numeric(rankMatrix(Amat))

## [1] 2
```

■

The rank of an $n \times n$ square matrix \mathbf{A} is directly related to its invertibility. If $\text{rank}(\mathbf{A}) = n$ then \mathbf{A}^{-1} exists. This result makes sense. Since \mathbf{A}^{-1} is used to solve a system of n linear equations in n unknowns, it will have a unique solution as long as no equation in the system can be written as a linear combination of the other equations.

3.4.2 Partitioned matrices and partitioned inverses

Consider a general $n \times m$ matrix \mathbf{A} :

$$\underset{(n \times m)}{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}.$$

In some situations we might want to partition \mathbf{A} into sub-matrices containing sub-blocks of the elements of \mathbf{A} :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

where the sub-matrices \mathbf{A}_{11} , \mathbf{A}_{12} , \mathbf{A}_{21} , and \mathbf{A}_{22} contain the appropriate sub-elements of \mathbf{A} . For example, consider

$$\mathbf{A} = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array} \right] = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

where

$$\mathbf{A}_{11} = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix}, \mathbf{A}_{12} = \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix},$$

$$\mathbf{A}_{21} = \begin{bmatrix} 9 & 10 \\ 13 & 14 \end{bmatrix}, \mathbf{A}_{22} = \begin{bmatrix} 11 & 12 \\ 15 & 16 \end{bmatrix}.$$

In R sub-matrices can be combined column-wise using `cbind()`, and row-wise using `rbind()`. For example, to create the matrix \mathbf{A} from the sub-matrices use

```
A11mat = matrix(c(1, 2, 5, 6), 2, 2, byrow = TRUE)
A12mat = matrix(c(3, 4, 7, 8), 2, 2, byrow = TRUE)
A21mat = matrix(c(9, 10, 13, 14), 2, 2, byrow = TRUE)
A22mat = matrix(c(11, 12, 15, 16), 2, 2, byrow = TRUE)
Amat = rbind(cbind(A11mat, A12mat), cbind(A21mat, A22mat))
Amat

##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
## [4,]   13   14   15   16
```

The basic matrix operations work on sub-matrices in the obvious way. To illustrate, consider another matrix \mathbf{B} that is conformable with \mathbf{A} and partitioned in the same way:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}.$$

Then

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} + \mathbf{B}_{11} & \mathbf{A}_{12} + \mathbf{B}_{12} \\ \mathbf{A}_{21} + \mathbf{B}_{21} & \mathbf{A}_{22} + \mathbf{B}_{22} \end{bmatrix}.$$

If all of the sub-matrices of \mathbf{A} and \mathbf{B} are conformable then

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} \\ \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} \end{bmatrix}.$$

The transpose of a partitioned matrix satisfies

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}' = \begin{bmatrix} \mathbf{A}'_{11} & \mathbf{A}'_{21} \\ \mathbf{A}'_{12} & \mathbf{A}'_{22} \end{bmatrix}.$$

Notice the interchange of the two off-diagonal blocks.

A partitioned matrix \mathbf{A} with appropriate invertible sub-matrices \mathbf{A}_{11} and \mathbf{A}_{22} has a partitioned inverse of the form

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{C}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} - \mathbf{A}_{11}\mathbf{A}_{12}\mathbf{C}^{-1} \\ -\mathbf{C}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{C}^{-1} \end{bmatrix},$$

where $\mathbf{C} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$. This formula can be verified by direct calculation.

3.5 Positive Definite Matrices

Let \mathbf{A} be an $n \times n$ symmetric matrix. The matrix \mathbf{A} is *positive definite* if for any $n \times 1$ vector $\mathbf{x} \neq \mathbf{0}$

$$\mathbf{x}'\mathbf{A}\mathbf{x} > 0.$$

This condition implies that there is no non-zero vector \mathbf{x} such that $\mathbf{A}\mathbf{x} = \mathbf{0}$, which implies that \mathbf{A} has full rank n .

The matrix \mathbf{A} is *positive semi-definite* if for any $n \times 1$ vector $\mathbf{x} \neq \mathbf{0}$

$$\mathbf{x}'\mathbf{A}\mathbf{x} \geq 0.$$

Hence, if \mathbf{A} is positive semi-definite then there exists some $n \times 1$ vector \mathbf{x} such that $\mathbf{A}\mathbf{x} = \mathbf{0}$, which implies that \mathbf{A} does not have full rank n .

3.5.1 Matrix square root

To motivate the idea of the square root of a matrix, consider the square root factorization of a positive number a :

$$a = \sqrt{a} \times \sqrt{a} = a^{1/2} \times a^{1/2}.$$

That is, we can factorize any positive number into the product of its square root. It turns out we can do a similar factorization for a positive definite and symmetric matrix.

let \mathbf{A} be an $n \times n$ positive definite and symmetric matrix. Then it is possible to factor \mathbf{A} as

$$\mathbf{A} = \mathbf{C}'\mathbf{C},$$

where \mathbf{C} is an $n \times n$ upper triangular matrix with non-negative diagonal elements called the *Cholesky factor* of \mathbf{A} . This factorization is also called a *matrix square root factorization* by defining the square root matrix $\mathbf{A}^{1/2} = \mathbf{C}$. Then we can write $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$.⁴ If all of

⁴ The Cholesky factorization is one way of defining the matrix square root, where the square root matrix is upper triangular. Other matrix square factorizations exist where the square root matrix is not necessarily upper triangular.

the diagonal elements of \mathbf{C} are positive then \mathbf{A} is positive definite. Otherwise, \mathbf{A} is positive semi-definite.

Example 3.13. Cholesky decomposition in R

The R function `chol()` computes the Cholesky factorization of a square symmetrix matrix. For example, consider the 2×2 correlation matrix:

```
Sigma = matrix(c(1, 0.75, 0.75, 1), 2, 2)
Sigma

##      [,1] [,2]
## [1,] 1.00 0.75
## [2,] 0.75 1.00
```

The Cholesky matrix square root factorization is

```
C = chol(Sigma)
C

##      [,1] [,2]
## [1,]    1 0.7500
## [2,]    0 0.6614

t(C) %*% C

##      [,1] [,2]
## [1,] 1.00 0.75
## [2,] 0.75 1.00
```

Here, all the diagonal elements of \mathbf{C} are positive so that covariance matrix is positive definite. Next, suppose the correlation matrix is

```
Sigma = matrix(c(1, 1, 1, 1), 2, 2)
Sigma

##      [,1] [,2]
## [1,]    1    1
## [2,]    1    1
```

so that the two variables are perfectly correlated. Then

```
Sigma = matrix(c(1, 1, 1, 1), 2, 2)
Sigma

##      [,1] [,2]
## [1,]    1    1
## [2,]    1    1

chol(Sigma, pivot = TRUE)

##      [,1] [,2]
## [1,]    1    1
## [2,]    0    0
```

```
## attr(,"pivot")
## [1] 1 2
## attr(,"rank")
## [1] 1
```

Here, we see that one of the diagonal elements of Cholesky factor is zero and that the correlation matrix is rank deficient.

■

3.6 Multivariate Probability Distributions Using Matrix Algebra

In this section, we show how matrix algebra can be used to simplify many of the messy expressions concerning expectations and covariances between multiple random variables, and we show how certain multivariate probability distributions (e.g., the multivariate normal distribution) can be expressed using matrix algebra.

3.6.1 Random vectors

Let X_1, \dots, X_n denote n random variables for $i = 1, \dots, n$ let $\mu_i = E[X_i]$ and $\sigma_i^2 = \text{var}(X_i)$, and let $\sigma_{ij} = \text{cov}(X_i, X_j)$ for $i \neq j$. Define the $n \times 1$ random vector $\mathbf{X} = (X_1, \dots, X_n)'$. Associated with \mathbf{X} is the $n \times 1$ vector of expected values:

$$\underset{n \times 1}{\boldsymbol{\mu}} = E[\mathbf{X}] = \begin{pmatrix} E[X_1] \\ \vdots \\ E[X_n] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix}.$$

3.6.2 Covariance matrix

The covariance matrix, $\boldsymbol{\Sigma}$, summarizes the variances and covariances of the elements of the random vector \mathbf{X} . In general, the covariance matrix of a random vector \mathbf{X} (sometimes called the variance of the vector \mathbf{X}) with mean vector $\boldsymbol{\mu}$ is defined as:

$$\text{cov}(\mathbf{X}) = E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})'] = \boldsymbol{\Sigma}.$$

If \mathbf{X} has n elements then $\boldsymbol{\Sigma}$ will be the symmetric $n \times n$ matrix,

$$\underset{n \times n}{\boldsymbol{\Sigma}} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{pmatrix}.$$

For the case $N = 2$, we have by direct calculation

$$\begin{aligned} E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)'] &= E \left[\begin{pmatrix} X_1 - \mu_1 \\ X_2 - \mu_2 \end{pmatrix} \cdot (X_1 - \mu_1, X_2 - \mu_2) \right] \\ &= E \left[\begin{pmatrix} (X_1 - \mu_1)^2 & (X_1 - \mu_1)(X_2 - \mu_2) \\ (X_2 - \mu_2)(X_1 - \mu_1) & (X_2 - \mu_2)^2 \end{pmatrix} \right] \\ &= \begin{pmatrix} E[(X_1 - \mu_1)^2] & E[(X_1 - \mu_1)(X_2 - \mu_2)] \\ E[(X_2 - \mu_2)(X_1 - \mu_1)] & E[(X_2 - \mu_2)^2] \end{pmatrix} \\ &= \begin{pmatrix} \text{var}(X_1) & \text{cov}(X_1, X_2) \\ \text{cov}(X_2, X_1) & \text{var}(X_2) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \boldsymbol{\Sigma}. \end{aligned}$$

3.6.3 Correlation matrix

The correlation matrix, \mathbf{C} , summarizes all pairwise correlations between the elements of the $n \times 1$ random vector \mathbf{X} and is given by

$$\mathbf{C} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{12} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n} & \rho_{2n} & \cdots & 1 \end{pmatrix}$$

Let $\boldsymbol{\Sigma}$ denote the covariance matrix of \mathbf{X} . Then the correlation matrix \mathbf{C} can be computed using

$$\mathbf{C} = \mathbf{D}^{-1} \boldsymbol{\Sigma} \mathbf{D}^{-1}, \quad (3.3)$$

where \mathbf{D} is an $n \times n$ diagonal matrix with the standard deviations of the elements of \mathbf{X} along the main diagonal

$$\mathbf{D} = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{pmatrix}.$$

Example 3.14. Solving systems of linear equations in R.

In R the `cov2cor()` function can be used to compute the correlation matrix given the covariance matrix using (3.3). ■

3.6.4 Variance of linear combination of random vectors

Consider the $n \times 1$ random vector \mathbf{X} with mean vector μ and covariance matrix Σ . Let $\mathbf{a} = (a_1, \dots, a_n)'$ be an $n \times 1$ vector of constants and consider the random variable $Y = \mathbf{a}'\mathbf{X} = a_1X_1 + \dots + a_nX_n$. Then,

$$\mu_Y = E[Y] = E[\mathbf{a}'\mathbf{X}] = \mathbf{a}'E[\mathbf{X}] = \mathbf{a}'\mu.$$

and,

$$\text{var}(Y) = \text{var}(\mathbf{a}'\mathbf{X}) = E[(\mathbf{a}'\mathbf{X} - \mathbf{a}'\mu)^2] = E[(\mathbf{a}'(\mathbf{X} - \mu))^2]$$

since $Y = \mathbf{a}'\mathbf{X}$ is a scalar. Now we use a trick from matrix algebra. If z is a scalar (think of $z = 2$) then $z'z = z \cdot z' = z^2$. Let $z = \mathbf{a}'(\mathbf{X} - \mu)$ and so $z \cdot z' = \mathbf{a}'(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\mathbf{a}$. Then,

$$\begin{aligned} \text{var}(Y) &= E[z^2] = E[z \cdot z'] \\ &= E[\mathbf{a}'(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\mathbf{a}] \\ &= \mathbf{a}'E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)']\mathbf{a} \\ &= \mathbf{a}'\text{cov}(\mathbf{X})\mathbf{a} = \mathbf{a}'\Sigma\mathbf{a}. \end{aligned}$$

3.6.5 Covariance between linear combination of two random vectors

Consider the $n \times 1$ random vector \mathbf{X} with mean vector μ and covariance matrix Σ . Let $\mathbf{a} = (a_1, \dots, a_n)'$ and $\mathbf{b} = (b_1, \dots, b_n)'$ be $n \times 1$ vectors of constants, and consider the random variable $Y = \mathbf{a}'\mathbf{X} = a_1X_1 + \dots + a_nX_n$ and $Z = \mathbf{b}'\mathbf{X} = b_1X_1 + \dots + b_nX_n$. From the definition of covariance we have:

$$\text{cov}(Y, Z) = E[(Y - E[Y])(Z - E[Z])]$$

which may be rewritten in matrix notation as,

$$\begin{aligned}
\text{cov}(\mathbf{a}'\mathbf{X}, \mathbf{b}'\mathbf{X}) &= E[(\mathbf{a}'\mathbf{X} - \mathbf{a}'\mu)(\mathbf{b}'\mathbf{X} - \mathbf{b}'\mu)] \\
&= E[\mathbf{a}'(\mathbf{X} - \mu)\mathbf{b}'(\mathbf{X} - \mu)] \\
&= E[\mathbf{a}'(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\mathbf{b}] \\
&= \mathbf{a}'E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)']\mathbf{b} \\
&= \mathbf{a}'\Sigma\mathbf{b}.
\end{aligned}$$

Since $\mathbf{a}'(\mathbf{X} - \mu)$ and $\mathbf{b}'(\mathbf{X} - \mu)$ are scalars, we can use the trick:

$$\mathbf{a}'(\mathbf{X} - \mu)\mathbf{b}'(\mathbf{X} - \mu) = \mathbf{a}'(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\mathbf{b}.$$

3.6.6 Variance of a vector of linear functions of a random vector

Consider the $n \times 1$ random vector \mathbf{X} with mean vector μ and covariance matrix Σ . Let \mathbf{A} be an $m \times n$ matrix. Then the $m \times 1$ random vector $\mathbf{Y} = \mathbf{AX}$ represents m linear functions of \mathbf{X} . By the linearity of expectation, we have $E[\mathbf{Y}] = \mathbf{AE}[\mathbf{X}] = \mathbf{A}\mu$. The $m \times m$ covariance matrix of \mathbf{Y} is given by

$$\begin{aligned}
\text{var}(\mathbf{Y}) &= E[(\mathbf{Y} - E[\mathbf{Y}])(\mathbf{Y} - E[\mathbf{Y}])'] \\
&= E[(\mathbf{AX} - \mathbf{A}\mu)(\mathbf{AX} - \mathbf{A}\mu)'] \\
&= E[\mathbf{A}(\mathbf{X} - \mu)(\mathbf{A}(\mathbf{X} - \mu))'] \\
&= E[\mathbf{A}(\mathbf{X} - \mu)(\mathbf{X} - \mu)'\mathbf{A}'] \\
&= \mathbf{AE}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)']\mathbf{A}' \\
&= \mathbf{A}\Sigma\mathbf{A}'.
\end{aligned}$$

3.6.7 Bivariate normal distribution

Let X and Y be distributed bivariate normal. The joint pdf is given by

$$\begin{aligned}
f(x, y) &= \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho_{XY}^2}} \times \\
&\exp\left\{-\frac{1}{2(1-\rho_{XY}^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - \frac{2\rho_{XY}(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right\}
\end{aligned} \tag{3.4}$$

where $E[X] = \mu_X$, $E[Y] = \mu_Y$, $\text{sd}(X) = \sigma_X$, $\text{sd}(Y) = \sigma_Y$, and $\rho_{XY} = \text{cor}(X, Y)$. The correlation coefficient ρ_{XY} describes the dependence between X and Y . If $\rho_{XY} = 0$ then X and Y are independent and the pdf collapses to the pdf of the standard bivariate normal distribution.

The formula for the bivariate normal distribution (3.4) is a bit messy. We can greatly simplify the formula by using matrix algebra. Define the 2×1 vectors $\mathbf{x} = (x, y)'$ and $\mu = (\mu_X, \mu_Y)'$, and the 2×2 matrix:

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{pmatrix}.$$

Then the bivariate normal distribution (3.4) may be compactly expressed as,

$$f(\mathbf{x}) = \frac{1}{2\pi \det(\Sigma)^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)' \Sigma^{-1} (\mathbf{x}-\mu)},$$

where,

$$\det(\Sigma) = \sigma_X^2 \sigma_Y^2 - \sigma_{XY}^2 = \sigma_X^2 \sigma_Y^2 - \sigma_X^2 \sigma_Y^2 \rho_{XY}^2 = \sigma_X^2 \sigma_Y^2 (1 - \rho_{XY}^2).$$

3.6.8 Multivariate normal distribution

Consider n random variables X_1, \dots, X_n and assume they are jointly normally distributed. Define the $n \times 1$ vectors $\mathbf{X} = (X_1, \dots, X_n)', \mathbf{x} = (x_1, \dots, x_n)'$ and $\mu = (\mu_1, \dots, \mu_n)'$, and the $n \times n$ covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{pmatrix}.$$

Then $\mathbf{X} \sim N(\mu, \Sigma)$ means that the random vector \mathbf{X} has a multivariate normal distribution with mean vector μ and covariance matrix Σ . The pdf of the multivariate normal distribution can be compactly expressed as:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2\pi^{n/2} \det(\Sigma)^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)' \Sigma^{-1} (\mathbf{x}-\mu)} \\ &= (2\pi)^{-n/2} \det(\Sigma)^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\mu)' \Sigma^{-1} (\mathbf{x}-\mu)}. \end{aligned}$$

Example 3.15. Creating a multivariate normal random vector from a vector of independent standard normal random variables.

Let $\mathbf{Z} = (Z_1, \dots, Z_n)'$ denote a vector of n iid standard normal random variables. Then $\mathbf{Z} \sim N(\mathbf{0}, \mathbf{I}_n)$ where \mathbf{I}_n denotes the n -dimensional identity matrix. Given \mathbf{Z} , we would like to create a random vector \mathbf{X} such that $\mathbf{X} \sim N(\mu, \Sigma)$. Let $\Sigma^{1/2}$ denote the Cholesky factor of Σ such that $\Sigma = \Sigma^{1/2} \Sigma^{1/2'}$. Define \mathbf{X} using

$$\mathbf{X} = \mu + \Sigma^{1/2} \mathbf{Z}.$$

Then

$$\begin{aligned} E[\mathbf{X}] &= \mu + \Sigma^{1/2} E[\mathbf{Z}] = \mu, \\ \text{var}(\mathbf{X}) &= \text{var}(\Sigma^{1/2} \mathbf{Z}) = \Sigma^{1/2} \text{var}(\mathbf{Z}) \Sigma^{1/2'} = \Sigma^{1/2} \mathbf{I}_n \Sigma^{1/2'} = \Sigma. \end{aligned}$$

Hence, $\mathbf{X} \sim N(\mu, \Sigma)$. ■

Example 3.16. Simulating multivariate normal random vectors in R

To be completed ■

3.7 Portfolio Math Using Matrix Algebra

Let R_i denote the return on asset $i = A, B, C$ and assume that R_A, R_B and R_C are jointly normally distributed with means, variances and covariances:

$$\mu_i = E[R_i], \sigma_i^2 = \text{var}(R_i), \text{cov}(R_i, R_j) = \sigma_{ij}.$$

Let x_i denote the share of wealth invested in asset i ($i = A, B, C$), and assume that all wealth is invested in the three assets so that $x_A + x_B + x_C = 1$. The portfolio return, $R_{p,x}$, is the random variable:

$$R_{p,x} = x_A R_A + x_B R_B + x_C R_C. \quad (3.5)$$

The subscript "x" indicates that the portfolio is constructed using the x-weights x_A, x_B and x_C . The expected return on the portfolio is:

$$\mu_{p,x} = E[R_{p,x}] = x_A \mu_A + x_B \mu_B + x_C \mu_C, \quad (3.6)$$

and the variance of the portfolio return is:

$$\sigma_{p,x}^2 = \text{var}(R_{p,x}) = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + x_C^2 \sigma_C^2 + 2x_A x_B \sigma_{AB} + 2x_A x_C \sigma_{AC} + 2x_B x_C \sigma_{BC}. \quad (3.7)$$

Notice that variance of the portfolio return depends on three variance terms and six covariance terms. Hence, with three assets there are twice as many covariance terms than variance terms contributing to portfolio variance. Even with three assets, the algebra representing the portfolio characteristics (3.5) - (3.7) is cumbersome. We can greatly simplify the portfolio algebra using matrix notation.

Define the following 3×1 column vectors containing the asset returns and portfolio weights:

$$\mathbf{R} = \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix}.$$

The probability distribution of the random return vector \mathbf{R} is simply the joint distribution of the elements of \mathbf{R} . Here all returns are jointly normally distributed and this joint distribution is completely characterized by the means, variances and covariances of the returns. We can

easily express these values using matrix notation as follows. The 3×1 vector of portfolio expected values is:

$$E[\mathbf{R}] = E \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} = \begin{pmatrix} E[R_A] \\ E[R_B] \\ E[R_C] \end{pmatrix} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} = \mu,$$

and the 3×3 covariance matrix of returns is,

$$\begin{aligned} \text{var}(\mathbf{R}) &= \begin{pmatrix} \text{var}(R_A) & \text{cov}(R_A, R_B) & \text{cov}(R_A, R_C) \\ \text{cov}(R_B, R_A) & \text{var}(R_B) & \text{cov}(R_B, R_C) \\ \text{cov}(R_C, R_A) & \text{cov}(R_C, R_B) & \text{var}(R_C) \end{pmatrix} \\ &= \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} = \Sigma. \end{aligned}$$

Notice that the covariance matrix is symmetric (elements off the diagonal are equal so that $\Sigma = \Sigma'$, where Σ' denotes the transpose of Σ) since $\text{cov}(R_A, R_B) = \text{cov}(R_B, R_A)$, $\text{cov}(R_A, R_C) = \text{cov}(R_C, R_A)$ and $\text{cov}(R_B, R_C) = \text{cov}(R_C, R_B)$.

The return on the portfolio using vector notation is:

$$R_{p,x} = \mathbf{x}'\mathbf{R} = (x_A, x_B, x_C) \cdot \begin{pmatrix} R_A \\ R_B \\ R_C \end{pmatrix} = x_A R_A + x_B R_B + x_C R_C.$$

Similarly, the expected return on the portfolio is:

$$\begin{aligned} \mu_{p,x} &= E[\mathbf{x}'\mathbf{R}] = \mathbf{x}'E[\mathbf{R}] = \mathbf{x}'\mu \\ &= (x_A, x_B, x_C) \cdot \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \end{pmatrix} = x_A \mu_A + x_B \mu_B + x_C \mu_C. \end{aligned}$$

The variance of the portfolio is:

$$\begin{aligned}\sigma_{p,x}^2 &= \text{var}(\mathbf{x}'\mathbf{R}) = \mathbf{x}'\Sigma\mathbf{x} = (x_A, x_B, x_C) \cdot \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} \begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix} \\ &= x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + x_C^2\sigma_C^2 + 2x_Ax_B\sigma_{AB} + 2x_Ax_C\sigma_{AC} + 2x_Bx_C\sigma_{BC}.\end{aligned}$$

Finally, the condition that the portfolio weights sum to one can be expressed as:

$$\mathbf{x}'\mathbf{1} = (x_A, x_B, x_C) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = x_A + x_B + x_C = 1,$$

where $\mathbf{1}$ is a 3×1 vector with each element equal to 1.

Consider another portfolio with weights $\mathbf{y} = (y_A, y_B, y_C)'$. The return on this portfolio is:

$$R_{p,y} = \mathbf{y}'\mathbf{R} = y_A R_A + y_B R_B + y_C R_C.$$

We often need to compute the covariance between the return on portfolio \mathbf{x} and the return on portfolio \mathbf{y} , $\text{cov}(R_{p,x}, R_{p,y})$. This can be easily expressed using matrix algebra:

$$\begin{aligned}\sigma_{xy} &= \text{cov}(R_{p,x}, R_{p,y}) = \text{cov}(\mathbf{x}'\mathbf{R}, \mathbf{y}'\mathbf{R}) \\ &= E[(\mathbf{x}'\mathbf{R} - \mathbf{x}'\mu)(\mathbf{y}'\mathbf{R} - \mathbf{y}'\mu)] = E[\mathbf{x}'(\mathbf{R} - \mu)(\mathbf{R} - \mu)'\mathbf{y}] \\ &= \mathbf{x}'E[(\mathbf{R} - \mu)(\mathbf{R} - \mu)']\mathbf{y} = \mathbf{x}'\Sigma\mathbf{y}.\end{aligned}$$

Notice that,

$$\begin{aligned}\sigma_{xy} &= \mathbf{x}'\Sigma\mathbf{y} = (x_A, x_B, x_C) \cdot \begin{pmatrix} \sigma_A^2 & \sigma_{AB} & \sigma_{AC} \\ \sigma_{AB} & \sigma_B^2 & \sigma_{BC} \\ \sigma_{AC} & \sigma_{BC} & \sigma_C^2 \end{pmatrix} \begin{pmatrix} y_A \\ y_B \\ y_C \end{pmatrix} \\ &= x_A y_A \sigma_A^2 + x_B y_B \sigma_B^2 + x_C y_C \sigma_C^2 \\ &\quad + (x_A y_B + x_B y_A) \sigma_{AB} + (x_A y_C + x_C y_A) \sigma_{AC} + (x_B y_C + x_C y_B) \sigma_{BC},\end{aligned}$$

which is quite a messy expression!

3.8 Derivatives of Simple Matrix Functions

Result: Let \mathbf{A} be an $n \times n$ symmetric matrix, and let \mathbf{x} and \mathbf{y} be $n \times 1$ vectors. Then,

$$\frac{\partial}{\partial \mathbf{x}}_{n \times 1} \mathbf{x}' \mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}' \mathbf{y} \\ \vdots \\ \frac{\partial}{\partial x_n} \mathbf{x}' \mathbf{y} \end{pmatrix} = \mathbf{y}, \quad (3.8)$$

$$\frac{\partial}{\partial \mathbf{x}}_{n \times 1} \mathbf{A} \mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1} (\mathbf{A} \mathbf{x})' \\ \vdots \\ \frac{\partial}{\partial x_n} (\mathbf{A} \mathbf{x})' \end{pmatrix} = \mathbf{A}, \quad (3.9)$$

$$\frac{\partial}{\partial \mathbf{x}}_{n \times 1} \mathbf{x}' \mathbf{A} \mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}' \mathbf{A} \mathbf{x} \\ \vdots \\ \frac{\partial}{\partial x_n} \mathbf{x}' \mathbf{A} \mathbf{x} \end{pmatrix} = 2 \mathbf{A} \mathbf{x}. \quad (3.10)$$

We will demonstrate these results with simple examples. Let,

$$\mathbf{A} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

First, consider (3.8). Now,

$$\mathbf{x}' \mathbf{y} = x_1 y_1 + x_2 y_2.$$

Then,

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}' \mathbf{y} = \begin{pmatrix} \frac{\partial}{\partial x_1} \mathbf{x}' \mathbf{y} \\ \frac{\partial}{\partial x_2} \mathbf{x}' \mathbf{y} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} (x_1 y_1 + x_2 y_2) \\ \frac{\partial}{\partial x_2} (x_1 y_1 + x_2 y_2) \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{y}.$$

Next, consider (3.9). Note that,

$$\mathbf{A} \mathbf{x} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} ax_1 + bx_2 \\ bx_1 + cx_2 \end{pmatrix},$$

and,

$$(\mathbf{A} \mathbf{x})' = (ax_1 + bx_2, bx_1 + cx_2).$$

Then,

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{A} \mathbf{x} = \begin{pmatrix} \frac{\partial}{\partial x_1} (ax_1 + bx_2, bx_1 + cx_2) \\ \frac{\partial}{\partial x_2} (ax_1 + bx_2, bx_1 + cx_2) \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \mathbf{A}.$$

Finally, consider (3.10). We have,

$$\mathbf{x}' \mathbf{A} \mathbf{x} = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = ax_1^2 + 2bx_1 x_2 + cx_2^2.$$

Then,

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}} \mathbf{x}' \mathbf{A} \mathbf{x} &= \begin{pmatrix} \frac{\partial}{\partial x_1} (ax_1^2 + 2bx_1x_2 + cx_2^2) \\ \frac{\partial}{\partial x_2} (ax_1^2 + 2bx_1x_2 + cx_2^2) \end{pmatrix} = \begin{pmatrix} 2ax_1 + 2bx_2 \\ 2bx_1 + 2cx_2 \end{pmatrix} \\ &= 2 \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 2\mathbf{A}\mathbf{x}.\end{aligned}$$

Example 3.17. Calculating an asset's marginal contribution to portfolio volatility.

In portfolio risk budgeting (see chapter 14), asset i 's marginal contribution to portfolio volatility $\sigma_p = (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}$ is given by

$$\text{MCR}_i^\sigma = \frac{\partial \sigma_p}{\partial x_i} = \frac{\partial (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}}{\partial x_i},$$

and approximates how much portfolio volatility changes when the allocation to asset i increases by a small amount. Using the chain rule and matrix derivatives we can compute the entire vector of asset marginal contributions at once

$$\begin{aligned}\frac{\partial (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}}{\partial \mathbf{x}} &= \frac{1}{2} (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{-1/2} \frac{\partial \mathbf{x}' \boldsymbol{\Sigma} \mathbf{x}}{\partial x_i} = \frac{1}{2} (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{-1/2} 2\boldsymbol{\Sigma} \mathbf{x} \\ &= (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{-1/2} \boldsymbol{\Sigma} \mathbf{x} = \frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p}.\end{aligned}$$

Then asset i 's marginal contribution is given by the i th row of $\frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p}$. ■

Example 3.18. Finding the global minimum variance portfolio.

Let \mathbf{R} denote an $n \times 1$ random vector of asset returns with $E[\mathbf{R}] = \mu$ and $\text{var}(\mathbf{R}) = \boldsymbol{\Sigma}$. The global minimum variance portfolio (see Chapter xxx, section xxx) \mathbf{m} solves the constrained minimization problem:

$$\min_{\mathbf{m}} \sigma_{p,m}^2 = \mathbf{m}' \boldsymbol{\Sigma} \mathbf{m} \text{ s.t. } \mathbf{m}' \mathbf{1} = 1. \quad (3.11)$$

The Lagrangian function is:

$$L(\mathbf{m}, \lambda) = \mathbf{m}' \boldsymbol{\Sigma} \mathbf{m} + \lambda(\mathbf{m}' \mathbf{1} - 1).$$

The first order conditions can be expressed in matrix notation as,

$$\underset{(n \times 1)}{\mathbf{0}} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \mathbf{m}} = \frac{\partial}{\partial \mathbf{m}} \mathbf{m}' \boldsymbol{\Sigma} \mathbf{m} + \frac{\partial}{\partial \mathbf{m}} \lambda(\mathbf{m}' \mathbf{1} - 1) = 2 \cdot \boldsymbol{\Sigma} \mathbf{m} + \lambda \cdot \mathbf{1} \quad (3.12)$$

$$\underset{(1 \times 1)}{0} = \frac{\partial L(\mathbf{m}, \lambda)}{\partial \lambda} = \frac{\partial}{\partial \lambda} \mathbf{m}' \boldsymbol{\Sigma} \mathbf{m} + \frac{\partial}{\partial \lambda} \lambda(\mathbf{m}' \mathbf{1} - 1) = \mathbf{m}' \mathbf{1} - 1 \quad (3.13)$$

These first order conditions represent a system of $n + 1$ linear equations in $n + 1$ unknowns (\mathbf{m} and λ). These equations can be represented in matrix form as the system

$$\begin{bmatrix} 2\boldsymbol{\Sigma} \mathbf{1} \\ \mathbf{1}' \ 0 \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix},$$

which is of the form $\mathbf{Az} = \mathbf{b}$ for

$$\mathbf{A} = \begin{bmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{bmatrix}, \mathbf{z} = \begin{bmatrix} \mathbf{m} \\ \lambda \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}.$$

The portfolio weight vector \mathbf{m} can be found as the first n elements of $\mathbf{z} = \mathbf{A}^{-1}\mathbf{b}$. ■

3.9 Further Reading

A classic textbook on linear algebra is Strang (1980). A review of basic matrix algebra with applications in economics and finance is given in Chang (1984). Matrix algebra results that are useful for statistical analysis are presented in Searle (1982), Gentile (2010) and Banerjee and Roy (2014). Lutkepohl (1997) and Abadir and Magnus (2005) present advanced matrix algebra useful for econometrics and statistics. Fieller (2015) overviews the basics of matrix algebra for statistics using R.

3.10 Exercises

Exercise 3.1. Use R to answer the following questions.

1. Create the following matrices and vectors

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 4 & 8 \\ 6 & 1 & 3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 4 & 4 & 0 \\ 5 & 9 & 1 \\ 2 & 2 & 5 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \mathbf{y} = \begin{bmatrix} 5 \\ 2 \\ 7 \end{bmatrix}$$

2. Compute the transposes of the above matrices and vectors.
3. Compute $\mathbf{A} + \mathbf{B}$, $\mathbf{A} - \mathbf{B}$, $2\mathbf{A}$, \mathbf{Ax} , \mathbf{AB} , $\mathbf{y}'\mathbf{Ax}$

Exercise 3.2. Consider the system of linear equations

$$\begin{aligned} x + y &= 1 \\ 2x + 4y &= 2 \end{aligned}$$

1. Plot the two lines and note the solution to the system of equations. Hint: you can use the R functions `abline()` or `curve()` to plot the lines.
2. Write the system in matrix notation as $\mathbf{Az} = \mathbf{b}$ and solve for \mathbf{z} by computing $\mathbf{z} = \mathbf{A}^{-1}\mathbf{b}$.

Exercise 3.3. Consider creating an equally weighted portfolio of three assets denoted A, B, and C. Assume the following information

$$\mu = \begin{bmatrix} 0.01 \\ 0.04 \\ 0.02 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.10 & 0.30 & 0.10 \\ 0.30 & 0.15 & -0.20 \\ 0.10 & -0.20 & 0.08 \end{bmatrix}.$$

Use R to answer the following questions.

1. Create the vector of portfolio weights.
2. Compute the expected return on the portfolio.
3. Compute the variance and standard deviation of the portfolio.

Exercise 3.4. Let R_i be a random variable denoting the simple return on asset i ($i = 1, \dots, N$) with $E[R_i] = \mu_i$, $\text{var}(R_i) = \sigma_i^2$ and $\text{cov}(R_i, R_j) = \sigma_{ij}$. Define the $N \times 1$ vectors $\mathbf{R} = (R_1, \dots, R_N)'$, $\mu = (\mu_1, \dots, \mu_N)'$, $\mathbf{x} = (x_1, \dots, x_N)'$, $\mathbf{y} = (y_1, \dots, y_N)'$, and $\mathbf{1} = (1, \dots, 1)'$, and the $N \times N$ covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}.$$

The vectors \mathbf{x} and \mathbf{y} contain portfolio weights (investment shares) that sum to one. Using simple matrix algebra, answer the following questions.

1. For the portfolios defined by the vectors \mathbf{x} and \mathbf{y} give the matrix algebra expressions for the portfolio returns $R_{p,x}$ and $R_{p,y}$ and the portfolio expected returns $\mu_{p,x}$ and $\mu_{p,y}$.
2. For the portfolios defined by the vectors \mathbf{x} and \mathbf{y} give the matrix algebra expressions for the constraint that the portfolio weights sum to one.
3. For the portfolios defined by the vectors \mathbf{x} and \mathbf{y} give the matrix algebra expressions for the portfolio variances $\sigma_{p,x}^2$ and $\sigma_{p,y}^2$, and the covariance between $R_{p,x}$ and $R_{p,y}$.
4. In the expression for the portfolio variance $\sigma_{p,x}^2$, how many variance terms are there? How many unique covariance terms are there?

References

1. Abadir, K.M., and Magnus, J.R. (2005). *Matrix Algebra* (Econometric Exercises, Vol. 1). Cambridge University Press.
2. Campbell, J.Y., Lo, A.W., and MacKinlay, A.C. (1997). *The Econometrics of Financial Markets*. Princeton, New Jersey: Princeton University Press.
3. Chiang, A. (1984). *Fundamental Methods of Mathematical Economics, Third Edition*. McGraw-Hill, New York.
4. Lutkepohl, H. (1997). *Handbook of Matrices*. Wiley.
5. Strang, G. (1980). *Linear Algebra and Its Applications*. Academic Press.

Chapter 4

Time Series Concepts

Updated: July 14, 2016
Copyright © Eric Zivot 2015

Financial variables such as asset prices and returns are naturally ordered by time. That is, these variables are *time series* variables. When we construct returns, the time index or data frequency becomes the investment horizon associated with the return. Typical data frequencies are daily, weekly, monthly and annual. In building probability models for time series variables, the time ordering of the data matters because we may think there are important temporal dependencies among the variables. For example, we might have reason to believe that the return on an asset this month is correlated with the return on the same asset in the previous month. This autocorrelation can only be defined if the time ordering of the data is preserved. A major complication in analyzing time series data is that the usual assumption of random sampling from a common population is not appropriate because it does not allow for any kind of time dependence in the time series variables. We would like to retain the notion that the observed data come from some population model, perhaps with time-invariant parameters, but we would like to allow the variables to have time dependencies. Fortunately, we can do this if the time series data come from a stationary time series process.

This chapter reviews some basic times series concepts that are essential for describing and modeling financial time series. Section 4.1 defines univariate time series processes and introduces the important concepts of stationarity and ergodicity. Covariance stationary time series processes are defined, which gives meaning to measuring linear time dependence using autocorrelation. The benchmark Gaussian White Noise process and related processes are introduced and illustrated using R. Some common non-stationary time series processes are also discussed including the famous random walk model. Section 4.2 introduces covariance stationary multivariate time series process. Such processes allow for dynamic interactions among groups of time series variables. Section 4.3 discusses time series model building and introduces the class of univariate autoregressive-moving average time series models and multivariate vector autoregression models. The properties of some simple models are derived and it is shown how to simulate observations from these models using R

The R packages used in this chapter are **mvtnorm** and **vars**. Make sure these packages are installed and loaded before running the R examples in this chapter.

4.1 Stochastic Processes

A discrete-time *stochastic process* or *time series process*

$$\{\dots, Y_1, Y_2, \dots, Y_t, Y_{t+1}, \dots\} = \{Y_t\}_{t=-\infty}^{\infty},$$

is a sequence of random variables indexed by time t^1 . In most applications, the time index is a regularly spaced index representing calendar time (e.g., days, months, years, etc.) but it can also be irregularly spaced representing event time (e.g., intra-day transaction times). In modeling time series data, the ordering imposed by the time index is important because we often would like to capture the temporal relationships, if any, between the random variables in the stochastic process. In random sampling from a population, the ordering of the random variables representing the sample does not matter because they are independent.

A realization of a stochastic process with T observations is the sequence of observed data

$$\{Y_1 = y_1, Y_2 = y_2, \dots, Y_T = y_T\} = \{y_t\}_{t=1}^T.$$

The goal of time series modeling is to describe the probabilistic behavior of the underlying stochastic process that is believed to have generated the observed data in a concise way. In addition, we want to be able to use the observed sample to estimate important characteristics of a time series model such as measures of time dependence. In order to do this, we need to make a number of assumptions regarding the joint behavior of the random variables in the stochastic process such that we may treat the stochastic process in much the same way as we treat a random sample from a given population.

4.1.1 Stationary stochastic processes

We often describe random sampling from a population as a sequence of independent, and identically distributed (iid) random variables $X_1, X_2 \dots$ such that each X_i is described by the same probability distribution F_X , and write $X_i \sim F_X$. With a time series process, we would like to preserve the identical distribution assumption but we do not want to impose the restriction that each random variable in the sequence is independent of all of the other variables. In many contexts, we would expect some dependence between random variables close together in time (e.g., X_1 , and X_2) but little or no dependence between random variables far apart in time (e.g., X_1 and X_{100}). We can allow for this type of behavior using the concepts of *stationarity* and *ergodicity*.

We start with the definition of strict stationarity.

Definition 4.1. Strict stationarity.

A stochastic process $\{Y_t\}$ is *strictly stationary* if, for any given finite integer r and for any set of subscripts t_1, t_2, \dots, t_r the joint distribution of $(Y_{t_1}, Y_{t_2}, \dots, Y_{t_r})$ depends only on

¹ To conserve on notation, we will represent the stochastic process $\{Y_t\}_{t=-\infty}^{\infty}$ simply as $\{Y_t\}$.

$t_1 - t, t_2 - t, \dots, t_r - t$ but not on t . In other words, the joint distribution of $(Y_{t_1}, Y_{t_2}, \dots, Y_{t_r})$ is the same as the joint distribution of $(Y_{t_1-t}, Y_{t_2-t}, \dots, Y_{t_r-t})$ for any value of t .

In simple terms, the joint distribution of random variables in a strictly stationary stochastic process is time invariant. For example, the joint distribution of (Y_1, Y_5, Y_7) is the same as the distribution of (Y_{12}, Y_{16}, Y_{18}) . Just like in an iid sample, in a strictly stationary process all of the individual random variables Y_t ($t = -\infty, \dots, \infty$) have the same marginal distribution F_Y . This means they all have the same mean, variance etc., assuming these quantities exist. However, assuming strict stationarity does not make any assumption about the correlations between $Y_t, Y_{t_1}, \dots, Y_{t_r}$ other than that the correlation between Y_t and Y_{t_r} only depends on $t - t_r$ (the time between Y_t and Y_{t_r}) and not on t . That is, strict stationarity allows for general temporal dependence between the random variables in the stochastic process.

A useful property of strict stationarity is that it is preserved under general transformations, as summarized in the following proposition.

Proposition 4.1. *Let $\{Y_t\}$ be strictly stationary and let $g(\cdot)$ be any function of the elements in $\{Y_t\}$. Then $\{g(Y_t)\}$, is also strictly stationary.*

For example, if $\{Y_t\}$ is strictly stationary then $\{Y_t^2\}$ and $\{Y_t Y_{t-1}\}$ are also strictly stationary. The following are some simple examples of strictly stationary processes.

Example 4.1. iid sequence.

If $\{Y_t\}$ is an iid sequence, then it is strictly stationary. ■

Example 4.2. Non iid sequence.

Let $\{Y_t\}$ be an iid sequence and let $X \sim N(0, 1)$ independent of $\{Y_t\}$. Define $Z_t = Y_t + X$. The sequence $\{Z_t\}$ is not an independent sequence (because of the common X) but is an identically distributed sequence and is strictly stationary. ■

Strict stationarity places very strong restrictions on the behavior of a time series. A related concept that imposes weaker restrictions and is convenient for time series model building is *covariance stationarity* or *weak stationarity*.

Definition 4.2. Covariance stationarity.

A stochastic process $\{Y_t\}$ is *covariance stationary* if

1. $E[Y_t] = \mu < \infty$ does not depend on t
2. $\text{var}(Y_t) = \sigma^2 < \infty$ does not depend on t
3. $\text{cov}(Y_t, Y_{t-j}) = \gamma_j$ exists, is finite, and depends only on j but not on t for $j = 0, 1, 2, \dots$

The term γ_j is called the j^{th} order *autocovariance*. The j^{th} order *autocorrelation* is defined as:

$$\rho_j = \frac{\text{cov}(Y_t, Y_{t-j})}{\sqrt{\text{var}(Y_t)\text{var}(Y_{t-j})}} = \frac{\gamma_j}{\sigma^2}. \quad (4.1)$$

The autocovariances, γ_j , measure the direction of linear dependence between Y_t and Y_{t-j} . The autocorrelations, ρ_j , measure both the direction and strength of linear dependence between Y_t and Y_{t-j} . With covariance stationarity, instead of assuming the entire joint distribution of a collection of random variables is time invariant we make a weaker assumption that only the mean, variance and autocovariances of the random variables are time invariant. A strictly stationary stochastic process $\{Y_t\}$ such that $E[Y_t]$, $\text{var}(Y_t)$, and all pairwise covariances exist is a *covariance stationary* stochastic process. However, a covariance stationary process need not be strictly stationary.

The autocovariances and autocorrelations are measures of the linear temporal dependence in a covariance stationary stochastic process. A graphical summary of this temporal dependence is given by the plot of ρ_j against j , and is called the *autocorrelation function* (ACF). Figure 4.1 illustrates an ACF for a hypothetical covariance stationary time series with $\rho_j = (0.9)^j$ for $j = 1, 2, \dots, 10$ created with

```
rho = 0.9
rhoVec = (rho)^(1:10)
ts.plot(rhoVec, type = "h", lwd = 2, col = "blue", xlab = "Lag j", ylab = expression(rho[j]))
```

For this process the strength of linear time dependence decays toward zero geometrically fast as j increases.

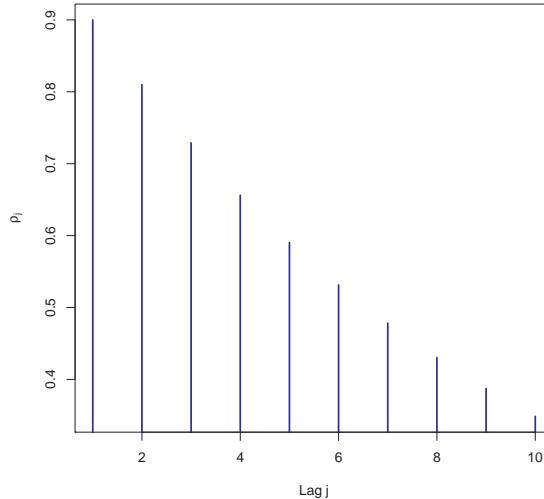


Fig. 4.1 ACF for time series with $\rho_j = (0.9)^j$

The definition of covariance stationarity requires that $E[Y_t] < \infty$ and $\text{var}(Y_t) < \infty$. That is, $E[Y_t]$ and $\text{var}(Y_t)$ must exist and be finite numbers. This is true if Y_t is normally distributed. However, it is not true if Y_t has a Student's t distribution with one degree of freedom². Hence, a strictly stationary stochastic process $\{Y_t\}$ where the (marginal) pdf of Y_t (for all t) has very fat tails may not be covariance stationary.

Example 4.3. Gaussian White Noise.

Let $Y_t \sim \text{iid } N(0, \sigma^2)$. Then $\{Y_t\}$ is called a *Gaussian white noise* (GWN) process and is denoted $Y_t \sim \text{GWN}(0, \sigma^2)$. Notice that:

$$\begin{aligned} E[Y_t] &= 0 \text{ independent of } t, \\ \text{var}(Y_t) &= \sigma^2 \text{ independent of } t, \\ \text{cov}(Y_t, Y_{t-j}) &= 0 \text{ (for } j > 0\text{) independent of } t \text{ for all } j, \end{aligned}$$

so that $\{Y_t\}$ satisfies the properties of a covariance stationary process. The defining characteristic of a GWN process is the lack of any predictable pattern over time in the realized values of the process. The term white noise comes from the electrical engineering literature and represents the absence of any signal.³

Simulating observations from a GWN process in R is easy: just simulate iid observations from a normal distribution. For example, to simulate $T = 250$ observations from the $\text{GWN}(0,1)$ process use:

```
set.seed(123)
y = rnorm(250)
```

The simulated iid $N(0,1)$ values are generated using the `rnorm()` function. The command `set.seed(123)` initializes R's internal random number generator using the seed value 123. Every time the random number generator seed is set to a particular value, the random number generator produces the same set of random numbers. This allows different people to create the same set of random numbers so that results are reproducible. The simulated data is illustrated in Figure 4.1.1 created using:

```
ts.plot(y, main = "Gaussian White Noise Process", xlab = "time", ylab = "y(t)",
        col = "blue", lwd = 2)
abline(h = 0)
```

The function `ts.plot()` creates a time series line plot with a dummy time index. An equivalent plot can be created using the generic `plot()` function with optional argument `type="l"`. The data in Figure 4.1.1 fluctuate randomly about the mean value zero and exhibit a constant volatility of one (typical magnitude of a fluctuation about zero). There is no visual evidence of any predictable pattern in the data.

² This is also called a Cauchy distribution. For this distribution $E[Y_t] = \text{var}(Y_t) = \text{cov}(Y_t, Y_{t-j}) = \infty$.

³ As an example of white noise, think of tuning an AM radio. In between stations there is no signal and all you hear is static. This is white noise.

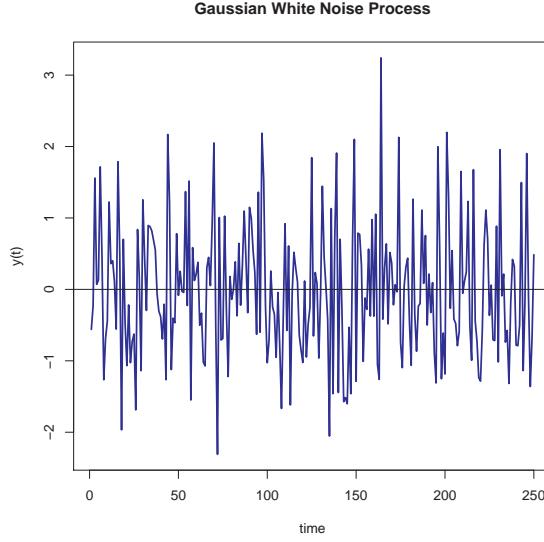


Fig. 4.2 Realization of a $\text{GWN}(0,1)$ process.

Example 4.4. GWN model for continuously compounded returns.

Let r_t denote the continuously compounded monthly return on Microsoft stock and assume that $r_t \sim \text{iid } N(0.01, (0.05)^2)$. We can represent this distribution in terms of a GWN process as follows

$$r_t = 0.01 + \varepsilon_t, \quad \varepsilon_t \sim \text{GWN}(0, (0.05)^2).$$

Hence, $\{r_t\}$ is a GWN process with a non-zero mean: $r_t \sim \text{GWN}(0.01, (0.05)^2)$. $T = 60$ simulated values of $\{r_t\}$ are computed using:

```
set.seed(123)
y = rnorm(60, mean = 0.01, sd = 0.05)
ts.plot(y, main = "GWN Process for Monthly Continuously Compounded Returns", xlab = "time",
        ylab = "r(t)", col = "blue", lwd = 2)
abline(h = c(0, 0.01, -0.04, 0.06), lwd = 2, lty = c("solid", "solid", "dotted",
        "dotted"), col = c("black", "red", "red", "red"))
```

and are illustrated in Figure 4.1.1. Notice that the returns fluctuate around the mean value of 0.01 and the size of a typical deviation from the mean is about 0.05. An implication of the GWN assumption for monthly returns is that non-overlapping multiperiod returns are also GWN. For example, consider the two-month return $r_t(2) = r_t + r_{t-1}$. The non-overlapping process $\{r_t(2)\} = \{\dots, r_{t-2}(2), r_t(2), r_{t+2}(2), \dots\}$ is GWN with mean $E[r_t(2)] = 2 \cdot \mu = 0.02$, variance $\text{var}(r_t(2)) = 2 \cdot \sigma^2 = 0.005$, and standard deviation $\text{sd}(r_t(2)) = \sqrt{2}\sigma = 0.071$.

■

Example 4.5. Independent white noise.

Let $Y_t \sim \text{iid } (0, \sigma^2)$. Then $\{Y_t\}$ is called an *independent white noise* (IWN) process and is denoted $Y_t \sim \text{IWN}(0, \sigma^2)$. The difference between GWN and IWN is that with IWN we don't

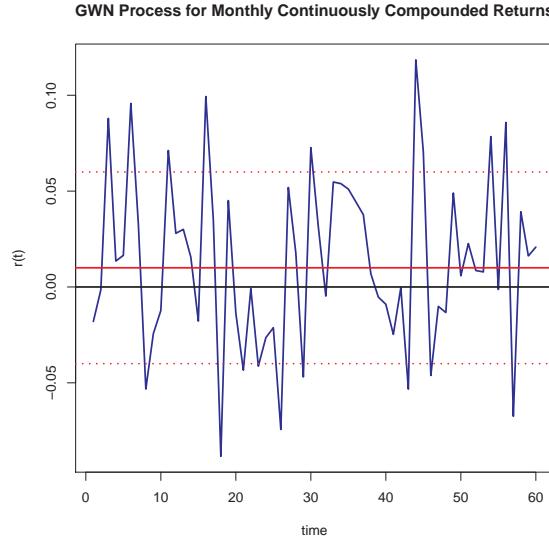


Fig. 4.3 Simulated returns from $GWN(0.01, (0.05)^2)$.

specify that all random variables are normally distributed. The random variables can have any distribution with mean zero and variance σ^2 . To illustrate, suppose $Y_t = \frac{1}{\sqrt{3}} \times t_3$ where t_3 denotes a Student's t distribution with 3 degrees of freedom. This process has $E[Y_t] = 0$ and $\text{var}(Y_t) = 1$.⁴ Figure 4.4 shows simulated observations from this process created using the R commands

```
set.seed(123)
y = (1/sqrt(3)) * rt(250, df = 3)
ts.plot(y, main = "Independent White Noise Process", xlab = "time", ylab = "y(t)",
        col = "blue", lwd = 2)
abline(h = 0)
```

The simulated IWN process resembles the GWN in Figure 4.4 but has more extreme observations.

Example 4.6. Weak white noise.

Let $\{Y_t\}$ be a sequence of uncorrelated random variables each with mean zero and variance σ^2 . Then $\{Y_t\}$ is called a *weak white noise* (WWN) process and is denoted $Y_t \sim \text{WWN}(0, \sigma^2)$. With a WWN process, the random variables are not independent, only uncorrelated. This allows for potential non-linear temporal dependence between the random variables in the process.

⁴ Recall, if $Y_t \sim t_v$ then $\text{var}(Y_t) = \frac{v}{v-2}$ for $v > 2$.

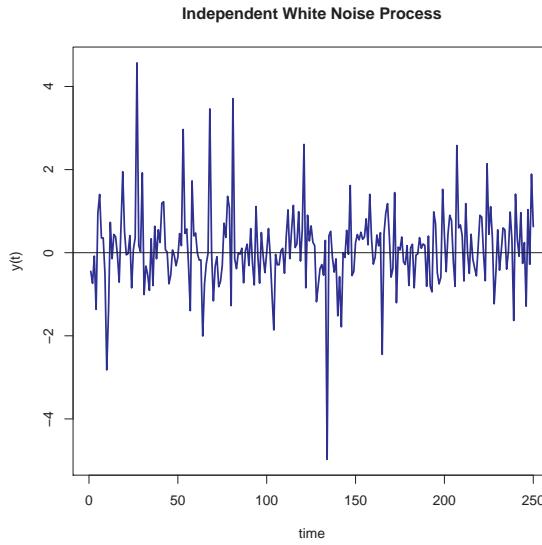


Fig. 4.4 Simulation of IWN(0,1) process: $Y_t \sim \frac{1}{\sqrt{3}} \times t_3$

4.1.2 Non-Stationary processes

In a covariance stationary stochastic process it is assumed that the means, variances and autocovariances are independent of time. In a non-stationary process, one or more of these assumptions is not true. The following examples illustrate some typical non-stationary time series processes.

Example 4.7. Deterministically trending process.

Suppose $\{Y_t\}$ is generated according to the *deterministically trending process*:

$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t, \quad \varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2), \\ t = 0, 1, 2, \dots$$

Then $\{Y_t\}$ is nonstationary because the mean of Y_t depends on t :

$$E[Y_t] = \beta_0 + \beta_1 t \text{ depends on } t.$$

Figure 4.1.2 shows a realization of this process with $\beta_0 = 0$, $\beta_1 = 0.1$ and $\sigma_\varepsilon^2 = 1$ created using the R commands:

```
set.seed(123)
e = rnorm(250)
y.dt = 0.1 * seq(1, 250) + e
ts.plot(y.dt, lwd = 2, col = "blue", main = "Deterministic Trend + Noise")
abline(a = 0, b = 0.1)
```

Here the non-stationarity is created by the deterministic trend $\beta_0 + \beta_1 t$ in the data. The non-stationary process $\{Y_t\}$ can be transformed into a stationary process by simply subtracting off the trend:

$$X_t = Y_t - \beta_0 - \beta_1 t = \varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2).$$

The detrended process $X_t \sim \text{GWN}(0, \sigma_\varepsilon^2)$ is obviously covariance stationary. ■

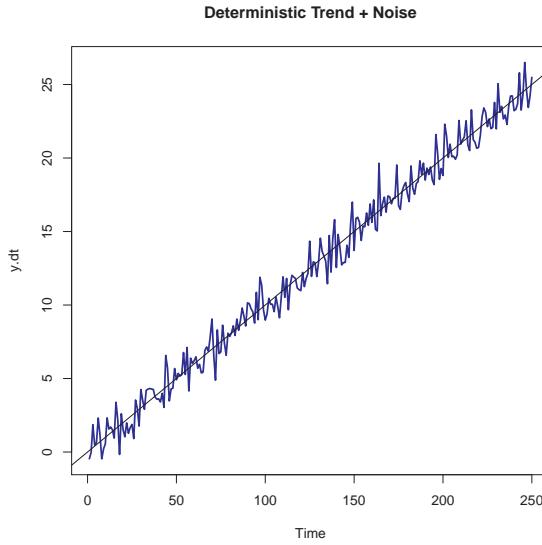


Fig. 4.5 Deterministically trending nonstationary process $Y_t = 0.1 \times t + \varepsilon_t, \varepsilon_t \sim \text{GWN}(0, 1)$.

Example 4.8. Random walk.

A random walk (RW) process $\{Y_t\}$ is defined as:

$$\begin{aligned} Y_t &= Y_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2), \\ Y_0 &\text{ is fixed (non-random).} \end{aligned}$$

By recursive substitution starting at $t = 1$, we have:

$$\begin{aligned} Y_1 &= Y_0 + \varepsilon_1, \\ Y_2 &= Y_1 + \varepsilon_2 = Y_0 + \varepsilon_1 + \varepsilon_2, \\ &\vdots \\ Y_t &= Y_{t-1} + \varepsilon_t = Y_0 + \varepsilon_1 + \cdots + \varepsilon_t \\ &= Y_0 + \sum_{j=1}^t \varepsilon_j. \end{aligned}$$

Now, $E[Y_t] = Y_0$ which is independent of t . However,

$$\text{var}(Y_t) = \text{var} \left(\sum_{j=1}^t \varepsilon_j \right) = \sum_{j=1}^t \sigma_\varepsilon^2 = \sigma_\varepsilon^2 \times t,$$

which depends on t , and so $\{Y_t\}$ is not stationary.

Figure 4.1.2 shows a realization of the RW process with $Y_0 = 0$ and $\sigma_\varepsilon^2 = 1$ created using the R commands:

```
set.seed(321)
e = rnorm(250)
y.rw = cumsum(e)
ts.plot(y.rw, lwd = 2, col = "blue", main = "Random Walk")
abline(h = 0)
```

The RW process looks much different from the GWN process in Figure 4.1.1. As the variance of the process increases linearly with time, the uncertainty about where the process will be at a given point in time increases with time.

Although $\{Y_t\}$ is non-stationary, a simple first-differencing transformation, however, yields a covariance stationary process:

$$X_t = Y_t - Y_{t-1} = \varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2).$$

■

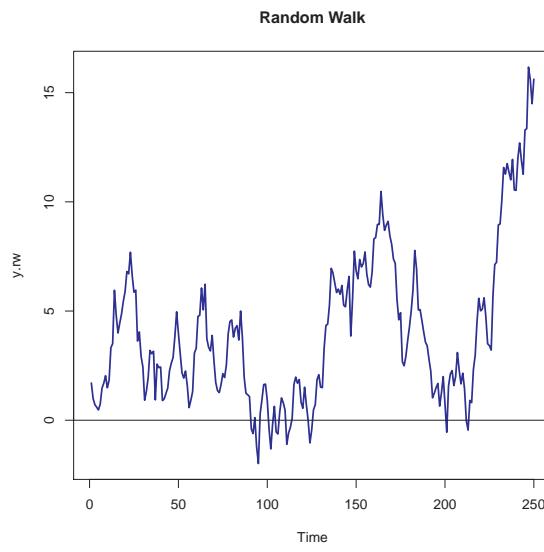


Fig. 4.6 Random walk process: $Y_t = Y_{t-1} + \varepsilon_t, \varepsilon_t \sim \text{GWN}(0, 1)$.

Example 4.9. Random walk with drift model for log stock prices.

Let r_t denote the continuously compounded monthly return on Microsoft stock and assume that $r_t \sim \text{GWN}(\mu, \sigma^2)$. Since $r_t = \ln(P_t/P_{t-1})$ it follows that $\ln P_t = \ln P_{t-1} + r_t$. Now, re-express r_t as $r_t = \mu + \varepsilon_t$ where $\varepsilon_t \sim \text{GWN}(0, \sigma^2)$. Then $\ln P_t = \ln P_{t-1} + \mu + \varepsilon_t$. By recursive substitution we have $\ln P_t = \ln P_0 + \mu t + \sum_{t=1}^t \varepsilon_t$ and so $\ln P_t$ follows a random walk process with drift value μ . Here, $E[\ln P_t] = \mu t$ and $\text{var}(\ln P_t) = \sigma^2 t$ so $\ln P_t$ is non-stationary because both the mean and variance depend on t . In this model, prices, however, do not follow a random walk since $P_t = e^{\ln P_t} = P_{t-1}e^{r_t}$.

■

4.1.3 Ergodicity

In a strictly stationary or covariance stationary stochastic process no assumption is made about the strength of dependence between random variables in the sequence. For example, in a covariance stationary stochastic process it is possible that $\rho_1 = \text{cor}(Y_t, Y_{t-1}) = \rho_{100} = \text{cor}(Y_t, Y_{t-100}) = 0.5$, say. However, in many contexts it is reasonable to assume that the strength of dependence between random variables in a stochastic process diminishes the farther apart they become. That is, $\rho_1 > \rho_2 \dots$ and that eventually $\rho_j = 0$ for j large enough. This diminishing dependence assumption is captured by the concept of *ergodicity*.

Definition 4.3. Ergodicity (intuitive definition).

Intuitively, a stochastic process $\{Y_t\}$ is *ergodic* if any two collections of random variables partitioned far apart in the sequence are essentially independent.

The formal definition of ergodicity is highly technical and requires advanced concepts in probability theory. However, the intuitive definition captures the essence of the concept. The stochastic process $\{Y_t\}$ is ergodic if Y_t and Y_{t-j} are essentially independent if j is large enough.

If a stochastic process $\{Y_t\}$ is covariance stationary and ergodic then strong restrictions are placed on the joint behavior of the elements in the sequence and on the type of temporal dependence allowed.

Example 4.10. White noise processes.

If $\{Y_t\}$ is GWN or IWN then it is both covariance stationary and ergodic.

■

Example 4.11. Covariance stationary but not ergodic process (White 1984, page 41).

Let $Y_t \sim \text{GWN}(0, 1)$ and let $X \sim N(0, 1)$ independent of $\{Y_t\}$. Define $Z_t = Y_t + X$. Then $\{Z_t\}$ is covariance stationary but not ergodic. To see why $\{Z_t\}$ is not ergodic, note that for all $j > 0$:

$$\begin{aligned}
\text{var}(Z_t) &= \text{var}(Y_t + X) = 1 + 1 = 2, \\
\gamma_j &= \text{cov}(Y_t + X, Y_{t-j} + X) = \text{cov}(Y_t, Y_{t-j}) + \text{cov}(Y_t, X) + \text{cov}(Y_{t-j}, X) + \text{cov}(X, X) \\
&= \text{cov}(X, X) = \text{var}(X) = 1, \\
\rho_j &= \frac{1}{2} \text{ for all } j.
\end{aligned}$$

Hence, the correlation between random variables separated far apart does not eventually go to zero and so $\{Z_t\}$ cannot be ergodic. ■

4.2 Multivariate Time Series

Consider n time series variables $\{Y_{1t}\}, \dots, \{Y_{nt}\}$. A *multivariate time series* is the $(n \times 1)$ vector time series $\{\mathbf{Y}_t\}$ where the i^{th} row of $\{\mathbf{Y}_t\}$ is $\{Y_{it}\}$. That is, for any time t , $\mathbf{Y}_t = (Y_{1t}, \dots, Y_{nt})'$. Multivariate time series analysis is used when one wants to model and explain the interactions and co-movements among a group of time series variables. In finance, multivariate time series analysis is used to model systems of asset returns, asset prices and exchange rates, the term structure of interest rates, asset returns/prices, and economic variables, etc.. Many of the time series concepts described previously for univariate time series carry over to multivariate time series in a natural way. Additionally, there are some important time series concepts that are particular to multivariate time series. The following sections give the details of these extensions.

4.2.1 Stationary and ergodic multivariate time series

A multivariate time series $\{\mathbf{Y}_t\}$ is covariance stationary and ergodic if all of its component time series are stationary and ergodic. The mean of \mathbf{Y}_t is defined as the $(n \times 1)$ vector

$$E[\mathbf{Y}_t] = (\mu_1, \dots, \mu_n)' = \boldsymbol{\mu},$$

where $\mu_i = E[Y_{it}]$ for $i = 1, \dots, n$. The variance/covariance matrix of \mathbf{Y}_t is the $(n \times n)$ matrix

$$\begin{aligned}
\text{var}(\mathbf{Y}_t) &= \boldsymbol{\Sigma} = E[(\mathbf{Y}_t - \boldsymbol{\mu})(\mathbf{Y}_t - \boldsymbol{\mu})'] \\
&= \begin{pmatrix} \text{var}(Y_{1t}) & \text{cov}(Y_{1t}, Y_{2t}) & \cdots & \text{cov}(Y_{1t}, Y_{nt}) \\ \text{cov}(Y_{2t}, Y_{1t}) & \text{var}(Y_{2t}) & \cdots & \text{cov}(Y_{2t}, Y_{nt}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(Y_{nt}, Y_{1t}) & \text{cov}(Y_{nt}, Y_{2t}) & \cdots & \text{var}(Y_{nt}) \end{pmatrix}.
\end{aligned}$$

The matrix $\boldsymbol{\Sigma}$ has elements $\sigma_{ij} = \text{cov}(Y_{it}, Y_{jt})$. The correlation matrix of \mathbf{Y}_t is the $(n \times n)$ matrix

$$\text{cor}(\mathbf{Y}_t) = \mathbf{C}_0 = \mathbf{D}^{-1}\boldsymbol{\Gamma}_0\mathbf{D}^{-1},$$

where \mathbf{D} is an $(n \times n)$ diagonal matrix with j^{th} diagonal element $\sigma_j = \text{sd}(Y_{jt})$.

Cross covariance and correlation matrices

For a univariate time series $\{Y_t\}$, the autocovariances, γ_k , and autocorrelations, ρ_k , summarize the linear time dependence in the data. With a multivariate time series $\{\mathbf{Y}_t\}$ each component has autocovariances and autocorrelations but there are also cross lead-lag covariances and correlations between all possible pairs of components. The autocovariances and autocorrelations of Y_{jt} , for $j = 1, \dots, n$, are defined as

$$\begin{aligned}\gamma_{jj}^k &= \text{cov}(Y_{jt}, Y_{jt-k}), \\ \rho_{jj}^k &= \text{corr}(Y_{jt}, Y_{jt-k}) = \frac{\gamma_{jj}^k}{\sigma_j^2},\end{aligned}$$

and these are symmetric in k : $\gamma_{jj}^k = \gamma_{jj}^{-k}$, $\rho_{jj}^k = \rho_{jj}^{-k}$. The *cross lag covariances* and *cross lag correlations* between Y_{it} and Y_{jt} are defined as

$$\begin{aligned}\gamma_{ij}^k &= \text{cov}(Y_{it}, Y_{jt-k}), \\ \rho_{ij}^k &= \text{corr}(Y_{it}, Y_{jt-k}) = \frac{\gamma_{ij}^k}{\sqrt{\sigma_i^2 \sigma_j^2}},\end{aligned}$$

and they are not necessarily symmetric in k . In general,

$$\gamma_{ij}^k = \text{cov}(Y_{it}, Y_{jt-k}) \neq \text{cov}(Y_{it}, Y_{jt+k}) = \text{cov}(Y_{jt}, Y_{it-k}) = \gamma_{ij}^{-k}.$$

If $\gamma_{ij}^k \neq 0$ for some $k > 0$ then Y_{jt} is said to *lead* Y_{it} . This implies that past values of Y_{jt} are useful for predicting future values of Y_{it} . Similarly, if $\gamma_{ij}^{-k} \neq 0$ for some $k > 0$ then Y_{it} is said to *lead* Y_{jt} . It is possible that Y_{it} leads Y_{jt} and vice-versa. In this case, there is said to be *dynamic feedback* between the two series.

All of the lag k cross covariances and correlations are summarized in the $(n \times n)$ lag k cross covariance and lag k cross correlation matrices

$$\begin{aligned}\boldsymbol{\Gamma}_k &= E[(\mathbf{Y}_t - \mu)(\mathbf{Y}_{t-k} - \mu)'] \\ &= \begin{pmatrix} \text{cov}(Y_{1t}, Y_{1t-k}) & \text{cov}(Y_{1t}, Y_{2t-k}) & \cdots & \text{cov}(Y_{1t}, Y_{nt-k}) \\ \text{cov}(Y_{2t}, Y_{1t-k}) & \text{cov}(Y_{2t}, Y_{2t-k}) & \cdots & \text{cov}(Y_{2t}, Y_{nt-k}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(Y_{nt}, Y_{1t-k}) & \text{cov}(Y_{nt}, Y_{2t-k}) & \cdots & \text{cov}(Y_{nt}, Y_{nt-k}) \end{pmatrix}, \\ \mathbf{C}_k &= \mathbf{D}^{-1}\boldsymbol{\Gamma}_k\mathbf{D}^{-1}.\end{aligned}$$

The matrices $\mathbf{\Gamma}_k$ and \mathbf{C}_k are not symmetric in k but it is easy to show that $\mathbf{\Gamma}_{-k} = \mathbf{\Gamma}'_k$ and $\mathbf{C}_{-k} = \mathbf{C}'_k$.

Example 4.12. Multivariate Gaussian white noise processes.

Let $\{\mathbf{Y}_t\}$ be an $n \times 1$ vector time series process. If $\mathbf{Y}_t \sim iid N(\mathbf{0}, \Sigma)$ then $\{\mathbf{Y}_t\}$ is called multivariate Gaussian white noise and is denoted $\mathbf{Y}_t \sim GWN(\mathbf{0}, \Sigma)$. Notice that

$$\begin{aligned} E[\mathbf{Y}_t] &= \mathbf{0}, \\ \text{var}(\mathbf{Y}_t) &= \Sigma, \\ \text{cov}(Y_{jt}, Y_{jt-k}) &= \gamma_{jj}^k = 0 \text{ (for } k > 0) \\ \text{cov}(Y_{it}, Y_{jt-k}) &= \gamma_{ij}^k = 0 \text{ (for } k > 0) \end{aligned}$$

Hence, $\{\mathbf{Y}_t\}$ exhibits no time dependence. That is, each element of \mathbf{Y}_t exhibits no time dependence and there is no dynamic feedback between any two elements. Simulating observations from $GWN(\mathbf{0}, \Sigma)$ requires simulating from a multivariate normal distribution, which can be done using the **mvtnorm** function **rmvnorm()**. For example, to simulate and plot $T = 250$ observation from a bivariate $GWN(\mathbf{0}, \Sigma)$ process with

$$\Sigma = \begin{pmatrix} 4 & 1 \\ 1 & 1 \end{pmatrix} \Rightarrow C = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

use:

```
library(mvtnorm)
Sigma = matrix(c(4, 1, 1, 1), 2, 2)
set.seed(123)
Y = rmvnorm(250, sigma = Sigma)
colnames(Y) = c("Y1", "Y2")
ts.plot(Y, lwd = 2, col = c("black", "blue"))
abline(h = 0)
legend("topleft", legend = c("Y1", "Y2"), lwd = 2, col = c("black", "blue"), lty = 1)
```

The simulated values are shown on the same plot in Figure 4.7. Both series fluctuate randomly about zero, and the first series (black line) has larger fluctuations (volatility) than the second series (blue line). The two series are contemporaneously correlated ($\rho_{12} = 0.5$) but are both uncorrelated over time ($\rho_{11}^k = \rho_{22}^k = 0$, $k > 0$) and are not cross-lag correlated ($\rho_{12}^k = \rho_{21}^k = 0$, $k > 0$). ■

4.3 Time Series Models

Time series models are probability models that are used to describe the behavior of a stochastic process. In many cases of interest, it is assumed that the stochastic process to be modeled is covariance stationary and ergodic. Then, the main feature of the process to be modeled is the time dependence between the random variables. In this section, we illustrate

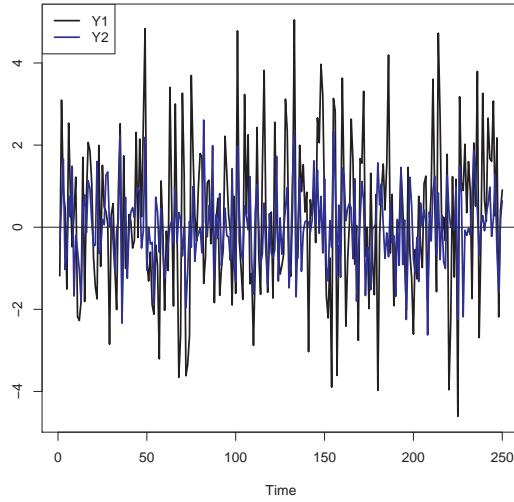


Fig. 4.7 Simulated bivariate GWN process.

some simple models for covariance stationary and ergodic time series that exhibit particular types of linear time dependence captured by autocorrelations. The univariate models, made popular originally by Box and Jenkins (1976), are called *autoregressive moving average* (ARMA) models. The multivariate model, made popular by Sims (1980), is called the *vector autoregressive* (VAR) model. These models are used extensively in economics and finance for modeling univariate and multivariate time series.

4.3.1 Moving average models

Moving average models are simple covariance stationary and ergodic time series models built from linear functions of GWN that can capture time dependence between random variables that lasts only for a finite number of lags.

MA(1) Model

Suppose you want to create a covariance stationary and ergodic stochastic process $\{Y_t\}$ in which Y_t and Y_{t-1} are correlated but Y_t and Y_{t-j} are not correlated for $j > 1$. That is, the time dependence in the process only lasts for one period. Such a process can be created using the *first order moving average* (MA(1)) model:

$$\begin{aligned} Y_t &= \mu + \varepsilon_t + \theta \varepsilon_{t-1}, \quad -1 < \theta < 1, \\ \varepsilon_t &\sim \text{GWN}(0, \sigma_\varepsilon^2). \end{aligned} \tag{4.2}$$

The MA(1) model is a simple linear function of the GWN random variables ε_t and ε_{t-1} . This linear structure allows for easy analysis of the model. The moving average parameter θ determines the sign and magnitude of the correlation between Y_t and Y_{t-1} . Clearly, if $\theta = 0$ then $Y_t = \mu + \varepsilon_t$ so that $\{Y_t\}$ is GWN with non-zero mean μ and exhibits no time dependence. As will be shown below, the MA(1) model produces a covariance stationary and ergodic process for any (finite) value of θ . The restriction $-1 < \theta < 1$ is called the *invertibility* restriction and will be explained below.

To verify that (4.2) process is a covariance stationary process we must show that the mean, variance and autocovariances are time invariant. For the mean, we have:

$$E[Y_t] = \mu + E[\varepsilon_t] + \theta E[\varepsilon_{t-1}] = \mu,$$

because $E[\varepsilon_t] = E[\varepsilon_{t-1}] = 0$.

For the variance, we have

$$\begin{aligned} \text{var}(Y_t) &= \sigma^2 = E[(Y_t - \mu)^2] = E[(\varepsilon_t + \theta\varepsilon_{t-1})^2] \\ &= E[\varepsilon_t^2] + 2\theta E[\varepsilon_t\varepsilon_{t-1}] + \theta^2 E[\varepsilon_{t-1}^2] \\ &= \sigma_\varepsilon^2 + 0 + \theta^2 \sigma_\varepsilon^2 = \sigma_\varepsilon^2(1 + \theta^2). \end{aligned}$$

The term $E[\varepsilon_t\varepsilon_{t-1}] = \text{cov}(\varepsilon_t, \varepsilon_{t-1}) = 0$ because $\{\varepsilon_t\}$ is independent process.

For $\gamma_1 = \text{cov}(Y_t, Y_{t-1})$, we have:

$$\begin{aligned} \text{cov}(Y_t, Y_{t-1}) &= E[(Y_t - \mu)(Y_{t-1} - \mu)] \\ &= E[(\varepsilon_t + \theta\varepsilon_{t-1})(\varepsilon_{t-1} + \theta\varepsilon_{t-2})] \\ &= E[\varepsilon_t\varepsilon_{t-1}] + \theta E[\varepsilon_t\varepsilon_{t-2}] \\ &\quad + \theta E[\varepsilon_{t-1}^2] + \theta^2 E[\varepsilon_{t-1}\varepsilon_{t-2}] \\ &= 0 + 0 + \theta\sigma_\varepsilon^2 + 0 = \theta\sigma_\varepsilon^2. \end{aligned}$$

Here, the sign of γ_1 is the same as the sign of θ .

For $\rho_1 = \text{cor}(Y_t, Y_{t-1})$ we have:

$$\rho_1 = \frac{\gamma_1}{\sigma^2} = \frac{\theta\sigma_\varepsilon^2}{\sigma_\varepsilon^2(1 + \theta^2)} = \frac{\theta}{(1 + \theta^2)}.$$

Clearly, $\rho_1 = 0$ if $\theta = 0$; $\rho_1 > 0$ if $\theta > 0$; $\rho_1 < 0$ if $\theta < 0$. Also, the largest value for $|\rho_1|$ is 0.5 which occurs when $|\theta| = 1$. Hence, a MA(1) model cannot describe a stochastic process that has $|\rho_1| > 0.5$. Also, note that there is more than one value of θ that produces the same value of ρ_1 . For example, θ and $1/\theta$ give the same value for ρ_1 . The invertibility restriction $-1 < \theta < 1$ provides a unique mapping between θ and ρ_1 .

For $\gamma_2 = \text{cov}(Y_t, Y_{t-2})$, we have:

$$\begin{aligned}
\text{cov}(Y_t, Y_{t-2}) &= E[(Y_t - \mu)(Y_{t-2} - \mu)] \\
&= E[(\varepsilon_t + \theta\varepsilon_{t-1})(\varepsilon_{t-2} + \theta\varepsilon_{t-3})] \\
&= E[\varepsilon_t\varepsilon_{t-2}] + \theta E[\varepsilon_t\varepsilon_{t-3}] \\
&\quad + \theta E[\varepsilon_{t-1}\varepsilon_{t-2}] + \theta^2 E[\varepsilon_{t-1}\varepsilon_{t-3}] \\
&= 0 + 0 + 0 + 0 = 0.
\end{aligned}$$

Similar calculations can be used to show that $\text{cov}(Y_t, Y_{t-j}) = \gamma_j = 0$ for $j > 1$. Hence, for $j > 1$ we have $\rho_j = 0$ and there is only time dependence between Y_t and Y_{t-1} but no time dependence between Y_t and Y_{t-j} for $j > 1$. Because $\rho_j = 0$ for $j > 1$ the MA(1) process is trivially ergodic.

Example 4.13. Simulated values from MA(1) process.

Consider simulating $T = 250$ observations from (4.2) with $\mu = 1$, $\theta = 0.9$ and $\sigma_\varepsilon = 1$. When simulating an MA(1) process, you need to decide how to start the simulation. The value of Y_t at $t = 0$, y_0 , is called the *initial value* and is the starting value for the simulation. Now, the first two observations from (4.2) starting at $t = 0$ are

$$\begin{aligned}
y_0 &= \mu + \varepsilon_0 + \theta\varepsilon_{-1}, \\
y_1 &= \mu + \varepsilon_1 + \theta\varepsilon_0.
\end{aligned}$$

It is common practice is to set $\varepsilon_{-1} = \varepsilon_0 = 0$ so that $y_0 = \mu$, $y_1 = \mu + \varepsilon_1 = y_0 + \varepsilon_1$ and ε_1 is the first simulated error term. The remaining observations for $t = 2, \dots, T$ are created from (4.2). We can implement the simulation in a number of ways in R. The most straightforward way is to use a simple loop:

```

n.obs = 250
mu = 1
theta = 0.9
sigma.e = 1
set.seed(123)
e = rnorm(n.obs, sd = sigma.e)
y = rep(0, n.obs)
y[1] = mu + e[1]
for (i in 2:n.obs) {
    y[i] = mu + e[i] + theta * e[i - 1]
}
head(y, n = 3)

## [1] 0.440 0.265 2.352

```

The “for loop” in R can be slow, however, especially for a very large number of simulations. The simulation can be more efficiently implemented using vectorized calculations as illustrated below:

```

set.seed(123)
e = rnorm(n.obs, sd = sigma.e)
em1 = c(0, e[1:(n.obs - 1)])
y = mu + e + theta * em1
head(y, n = 3)

```

```
## [1] 0.440 0.265 2.352
```

The vectorized calculation avoids looping all together and computes all of the simulated values at the same time. This can be considerably faster than the “for loop” calculation.

The MA(1) model is a special case of the more general autoregressive integrated moving average (ARIMA) model. R has many built-in functions and several packages for working with ARIMA models. In particular, the R function `arima.sim()` can be used to simulate observations from a MA(1) process. It essentially implements the simulation loop described above. The arguments of `arima.sim()` are

```
args(arima.sim)

## function (model, n, rand.gen = rnorm, innov = rand.gen(n, ...),
##           n.start = NA, start.innov = rand.gen(n.start, ...), ...)
## NULL
```

where `model` is a list with named components describing the ARIMA model parameters (excluding the mean value), `n` is the number of simulated observations, `rand.gen` specifies the pdf for ε_t , `innov` is a vector ε_t values of length `n`, `n.start` is the number of pre-simulation (burn-in) values for ε_t , `start.innov` is a vector of `n.start` pre-simulation values for ε_t , and ... specify any additional arguments for `rand.gen`. For example, to perform the same simulations as above use:

```
ma1.model = list(ma = 0.9)
set.seed(123)
y = mu + arima.sim(model = ma1.model, n = 250, n.start = 1, start.innov = 0)
head(y, n = 3)

## [1] 0.440 0.265 2.352
```

The `ma` component of the “list” object `ma1.model` specifies the value of θ for the MA(1) model, and is used as an input to the function `arima.sim()`. The options `n.start = 1` and `start.innov = 0` sets the start-up initial value $\varepsilon_0 = 0$. By default, `arima.sim()` sets $\mu = 0$, specifies $\varepsilon_t \sim \text{GWN}(0, 1)$, and returns $\varepsilon_t + \theta\varepsilon_{t-1}$ for $t = 1, \dots, T$. The simulated value for Y_t is constructed by adding the value of `mu` ($\mu = 1$) to the output of `arima.sim()`.

The function `ARMAacf()` can be used to compute the theoretical autocorrelations, ρ_j , from the MA(1) model (recall, $\rho_1 = \theta/(1 + \theta^2)$ and $\rho_j = 0$ for $j > 1$). For example, to compute ρ_j for $j = 1, \dots, 10$ use:

```
ma1.acf = ARMAacf(ar = 0, ma = 0.9, lag.max = 10)
ma1.acf

##    0     1     2     3     4     5     6     7     8     9    10
## 1.000 0.497 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

Figure 4.3.1 shows the simulated data and the theoretical ACF created using:

```

par(mfrow = c(2, 1))
ts.plot(y, main = "MA(1) Process: mu=1, theta=0.9", xlab = "time", ylab = "y(t)",
         col = "blue", lwd = 2)
abline(h = c(0, 1))
plot(0:10, mai.acf, type = "h", col = "blue", lwd = 2, main = "ACF for MA(1): theta=0.9",
      xlab = "lag", ylab = "rho(j)")
abline(h = 0)
par(mfrow = c(1, 1))

```

Compared to the GWN process in 4.1.1, the MA(1) process is a bit smoother in its appearance. This is due to the positive one-period time dependence captured by $\rho_1 = 0.4972$.

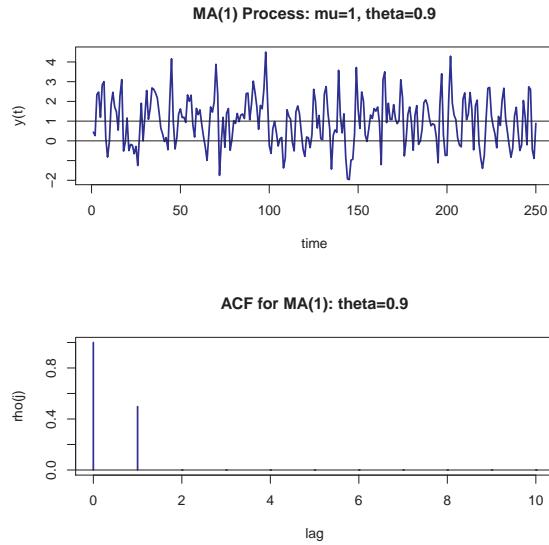


Fig. 4.8 Simulated values and theoretical ACF from MA(1) process with $\mu = 1$, $\theta = 0.9$ and $\sigma_\varepsilon^2 = 1$.

Example 4.14. MA(1) model for overlapping continuously compounded returns.

Let R_t denote the one-month continuously compounded return and assume that:

$$R_t \sim \text{iid } N(\mu, \sigma^2).$$

Consider creating a monthly time series of two-month continuously compounded returns using:

$$R_t(2) = R_t + R_{t-1}.$$

The time series of these two-month returns, observed monthly, overlap by one month:

$$\begin{aligned}
R_t(2) &= R_t + R_{t-1}, \\
R_{t-1}(2) &= R_{t-1} + R_{t-2}, \\
R_{t-2}(2) &= R_{t-2} + R_{t-3}, \\
&\vdots
\end{aligned}$$

The one-month overlap in the two-month returns implies that $\{R_t(2)\}$ follows an MA(1) process. To show this, we need to show that the autocovariances of $\{R_t(2)\}$ behave like the autocovariances of an MA(1) process.

To verify that $\{R_t(2)\}$ follows an MA(1) process, first we have:

$$\begin{aligned}
E[R_t(2)] &= E[R_t] + E[R_{t-1}] = 2\mu, \\
\text{var}(R_t(2)) &= \text{var}(R_t + R_{t-1}) = 2\sigma^2.
\end{aligned}$$

Next, we have:

$$\text{cov}(R_t(2), R_{t-1}(2)) = \text{cov}(R_t + R_{t-1}, R_{t-1} + R_{t-2}) = \text{cov}(R_{t-1}, R_{t-1}) = \text{var}(R_{t-1}) = \sigma^2,$$

and,

$$\begin{aligned}
\text{cov}(R_t(2), R_{t-2}(2)) &= \text{cov}(R_t + R_{t-1}, R_{t-2} + R_{t-3}) = 0, \\
\text{cov}(R_t(2), R_{t-j}(2)) &= 0 \text{ for } j > 1.
\end{aligned}$$

Hence, the autocovariances of $\{R_t(2)\}$ are those of an MA(1) process.

Notice that

$$\rho_1 = \frac{\sigma^2}{2\sigma^2} = \frac{1}{2}.$$

What MA(1) process describes $\{R_t(2)\}$? Because $\rho_1 = \frac{\theta}{1+\theta^2} = 0.5$ it follows that $\theta = 1$. Hence, the MA(1) process has mean 2μ and $\theta = 1$ and can be expressed as the MA(1) model:

$$\begin{aligned}
R_t(2) &= 2\mu + \varepsilon_t + \varepsilon_{t-1}, \\
\varepsilon_t &\sim \text{GWN}(0, \sigma^2).
\end{aligned}$$

■

MA(q) Model

The MA(q) model has the form

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}, \text{ where } \varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2). \quad (4.3)$$

The MA(q) model is stationary and ergodic provided $\theta_1, \dots, \theta_q$ are finite. The moments of the MA(q) (see end-of-chapter exercises) are

$$\begin{aligned} E[Y_t] &= \mu, \\ \gamma_0 &= \sigma^2(1 + \theta_1^2 + \cdots + \theta_q^2), \\ \gamma_j &= \begin{cases} (\theta_j + \theta_{j+1}\theta_1 + \theta_{j+2}\theta_2 + \cdots + \theta_q\theta_{q-j})\sigma^2 & \text{for } j = 1, 2, \dots, q \\ 0 & \text{for } j > q \end{cases}. \end{aligned}$$

Hence, the ACF of an MA(q) is non-zero up to lag q and is zero afterward.

Example 4.15. Overlapping returns and MA(q) models

MA(q) models often arise in finance through data aggregation transformations. For example, let $R_t = \ln(P_t/P_{t-1})$ denote the monthly continuously compounded return on an asset with price P_t . Define the annual return at time t using monthly returns as $R_t(12) = \ln(P_t/P_{t-12}) = \sum_{j=0}^{11} R_{t-j}$. Suppose $R_t \sim \text{GWN}(\mu, \sigma^2)$ and consider a sample of monthly returns of size T , $\{R_1, R_2, \dots, R_T\}$. A sample of annual returns may be created using *overlapping* or *non-overlapping* returns. Let $\{R_{12}(12), R_{13}(12), \dots, R_T(12)\}$ denote a sample of $T^* = T - 11$ monthly overlapping annual returns and $\{R_{12}(12), R_{24}(12), \dots, R_T(12)\}$ denote a sample of $T/12$ non-overlapping annual returns. Researchers often use overlapping returns in analysis due to the apparent larger sample size. One must be careful using overlapping returns because the monthly annual return sequence $\{R_t(12)\}$ is not a Gaussian white noise process even if the monthly return sequence $\{R_t\}$ is. To see this, straightforward calculations give:

$$\begin{aligned} E[R_t(12)] &= 12\mu, \\ \gamma_0 &= \text{var}(R_t(12)) = 12\sigma^2, \\ \gamma_j &= \text{cov}(R_t(12), R_{t-j}(12)) = (12 - j)\sigma^2 \text{ for } j < 12, \\ \gamma_j &= 0 \text{ for } j \geq 12. \end{aligned}$$

Since $\gamma_j = 0$ for $j \geq 12$ notice that $\{R_t(12)\}$ behaves like an MA(11) process:

$$\begin{aligned} R_t(12) &= 12\mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_{11}\varepsilon_{t-11}, \\ \varepsilon_t &\sim \text{GWN}(0, \sigma^2). \end{aligned}$$

■

4.3.2 Autoregressive Models

Moving average models can capture almost any kind of autocorrelation structure. However, this may require many moving average terms in (4.3). Another type of simple time series model is the *autoregressive model*. This model can capture complex autocorrelation patterns with a small number of parameters and is used more often in practice than the moving average model.

AR(1) Model

Suppose you want to create a covariance stationary and ergodic stochastic process $\{Y_t\}$ in which Y_t and Y_{t-1} are correlated, Y_t and Y_{t-2} are slightly less correlated, Y_t and Y_{t-3} are even less correlated and eventually Y_t and Y_{t-j} are uncorrelated for j large enough. That is, the time dependence in the process decays to zero as the random variables in the process get farther and farther apart. Such a process can be created using the *first order autoregressive* (AR(1)) model:

$$\begin{aligned} Y_t - \mu &= \phi(Y_{t-1} - \mu) + \varepsilon_t, \quad -1 < \phi < 1 \\ \varepsilon_t &\sim \text{iid } N(0, \sigma_\varepsilon^2) \end{aligned} \tag{4.4}$$

It can be shown that the AR(1) model is covariance stationary and ergodic provided $-1 < \phi < 1$. We will show that the AR(1) process has the following properties:

$$E[Y_t] = \mu, \tag{4.5}$$

$$\text{var}(Y_t) = \sigma^2 = \sigma_\varepsilon^2 / (1 - \phi^2), \tag{4.6}$$

$$\text{cov}(Y_t, Y_{t-1}) = \gamma_1 = \sigma^2 \phi, \tag{4.7}$$

$$\text{cor}(Y_t, Y_{t-1}) = \rho_1 = \gamma_1 / \sigma^2 = \phi, \tag{4.8}$$

$$\text{cov}(Y_t, Y_{t-j}) = \gamma_j = \sigma^2 \phi^j, \tag{4.9}$$

$$\text{cor}(Y_t, Y_{t-j}) = \rho_j = \gamma_j / \sigma^2 = \phi^j. \tag{4.10}$$

Notice that the restriction $|\phi| < 1$ implies that:

$$\lim_{j \rightarrow \infty} \rho_j = \phi^j = 0,$$

so that Y_t is essentially independent of Y_{t-j} for large j and so $\{Y_t\}$ is ergodic. For example, if $\phi = 0.5$ then $\rho_{10} = (0.5)^{10} = 0.001$; if $\phi = 0.9$ then $\rho_{10} = (0.9)^{10} = 0.349$. Hence, the closer ϕ is to unity the stronger is the time dependence in the process. If $\phi = 1$, then (4.4) becomes the random walk model $Y_t = Y_{t-1} + \varepsilon_t$ and is a non-stationary process.

Verifying covariance stationarity for the AR(1) model is more involved than for the MA(1) model, and establishing the properties (4.5) - (4.10) involves some tricks. In what follows, we will assume that $\{Y_t\}$ is a covariance stationary process and that $|\phi| < 1$. First, consider the derivation for (4.5). We have:

$$\begin{aligned} E[Y_t] &= \mu + \phi(E[Y_{t-1}] - \mu) + E[\varepsilon_t] \\ &= \mu + \phi E[Y_{t-1}] - \phi \mu. \end{aligned}$$

Given that $\{Y_t\}$ is covariance stationary it follows that $E[Y_t] = E[Y_{t-1}]$. Substituting $E[Y_t] = E[Y_{t-1}]$ into the above and solving for $E[Y_t]$ gives (4.5).

A similar trick can be used to derive (4.6):

$$\text{var}(Y_t) = \phi^2(\text{var}(Y_{t-1})) + \text{var}(\varepsilon_t) = \phi^2(\text{var}(Y_t)) + \sigma_\varepsilon^2,$$

which uses the fact that Y_{t-1} is independent of ε_t (because Y_{t-1} only depends on $t-1$ values) and $\text{var}(Y_t) = \text{var}(Y_{t-1})$ given that $\{Y_t\}$ is covariance stationary. Solving for $\sigma^2 = \text{var}(Y_t)$ gives (4.6). To determine (4.7), multiply both sides of (4.4) by $Y_{t-1} - \mu$ and take expectations to give:

$$\gamma_1 = E[(Y_t - \mu)(Y_{t-1} - \mu)] = \phi E[(Y_{t-1} - \mu)^2] + E[\varepsilon_t(Y_{t-1} - \mu)] = \phi\sigma^2,$$

which uses the fact that Y_{t-1} is independent of ε_t , and $\text{var}(Y_t) = \text{var}(Y_{t-1}) = \sigma^2$. Finally, to determine (4.9), multiply both sides of (4.4) by $Y_{t-j} - \mu$ and take expectations to give:

$$\begin{aligned}\gamma_j &= E[(Y_t - \mu)(Y_{t-j} - \mu)] = \phi E[(Y_{t-1} - \mu)(Y_{t-j} - \mu)] + E[\varepsilon_t(Y_{t-j} - \mu)] \\ &= \phi\gamma_{j-1},\end{aligned}$$

which uses the fact that Y_{t-j} is independent of ε_t , and $E[(Y_{t-1} - \mu)(Y_{t-j} - \mu)] = \gamma_{j-1}$ provided $\{Y_t\}$ is covariance stationary. Using recursive substitution and $\gamma_0 = \sigma^2$ gives (4.9).

The AR(1) Model can be re-expressed in the form of a linear regression model as follows:

$$\begin{aligned}Y_t - \mu &= \phi(Y_{t-1} - \mu) + \varepsilon_t \Rightarrow \\ Y_t &= \mu - \phi\mu + \phi Y_{t-1} + \varepsilon_t \\ &= c + \phi Y_{t-1} + \varepsilon_t,\end{aligned}$$

where $c = (1 - \phi)\mu \Rightarrow \mu = c/(1 - \phi)$. This regression model form is convenient for estimation by ordinary least squares.

Example 4.16. Simulated values from AR(1) process.

Consider simulating 250 observations from (4.4) with $\mu = 1$, $\phi = 0.9$ and $\sigma_\varepsilon = 1$. To start the simulation, an initial value or start-up value for Y_0 is required. A commonly used initial value is the mean value μ so that $Y_1 = \mu + \varepsilon_1$. As with the MA(1) model, this can be performed using a simple “for loop” in R:

```
phi = 0.9
mu = 1
sigma.e = 1
n.obs = 250
y = rep(0, n.obs)
set.seed(123)
e = rnorm(n.obs, sd = sigma.e)
y[1] = mu + e[1]
for (i in 2:n.obs) {
  y[i] = mu + phi * (y[i - 1] - mu) + e[i]
}
head(y, 3)

## [1] 0.440 0.265 1.898
```

Unfortunately, there is no easy way to vectorize the loop calculation. However, the R function `filter()`, with optional argument `method = "recursive"`, implements the AR(1) recursion efficiently in C code and so is more efficient than the for loop code in R above:

```
y = mu + filter(e, phi, method = "recursive")
head(y, 3)

## [1] 0.440 0.265 1.898
```

The R function `arima.sim()`, which internally uses the `filter()` function, can also be used to simulate observations from an AR(1) process. For the AR(1) model, the function `arima.sim()` simulates the components form of the AR(1) model

$$Y_t = \mu + u_t,$$

$$u_t = \phi u_{t-1} + \epsilon_t.$$

Hence, to replicate the “for loop” simulation with `arima.sim()` use

```
ar1.model = list(ar = 0.9)
mu = 1
set.seed(123)
y = mu + arima.sim(model = ar1.model, n = 250, n.start = 1, start.innov = 0)
head(y, 3)

## [1] 0.440 0.265 1.898
```

The R function `ARMAacf()` can be used to compute the theoretical ACF for an AR(1) model as follows

```
ar1.acf = ARMAacf(ar = 0.9, ma = 0, lag.max = 10)
```

The simulated AR(1) values and the ACF are shown in Figure 4.3.2. Compared to the MA(1) process in 4.3.1, the realizations from the AR(1) process are much smoother. That is, when Y_t wanders high above its mean it tends to stay above the mean for a while and when it wanders low below the mean it tends to stay below for a while.



AR(p) Model

The covariance stationary AR(p) model in mean-adjusted form is

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) + \varepsilon_t, \quad (4.11)$$

$$\varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2),$$

where $\mu = E[Y_t]$. Like the AR(1), restrictions on the autoregressive parameters ϕ_1, \dots, ϕ_p are required for $\{Y_t\}$ to be covariance stationary and ergodic. A detailed treatment of these restrictions is beyond the scope of this book. However, one simple necessary condition for $\{Y_t\}$ to be covariance stationary is $|\phi| < 1$ where $\phi = \phi_1 + \cdots + \phi_p$. Hence, in the AR(p) model the sum of the autoregressive components ϕ has a similar interpretation as the single autoregressive coefficient in the AR(1) model.

The regression form of the AR(p) is

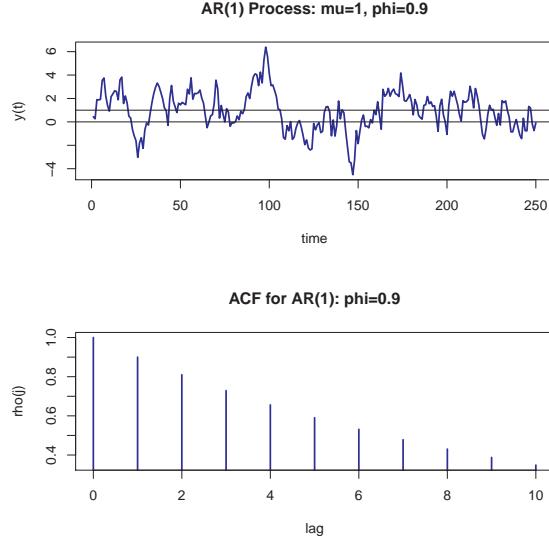


Fig. 4.9 Simulated values and ACF from AR(1) model with $\mu = 1, \phi = 0.9$ and $\sigma_\varepsilon^2 = 1$.

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t,$$

where $c = \mu / (1 - \phi_1 - \cdots - \phi_p) = \mu / (1 - \phi)$. This form is convenient for estimation purposes because it is in the form of a linear regression.

The AR(p) model is used very often in practice because of its simple linear structure and because it can capture a wide variety of autocorrelation patterns such as exponential decay, damped cyclical patterns, and oscillating damped cyclical patterns. Unfortunately, the mathematical derivation of the autocorrelations in the AR(p) model is complicated and tedious. The exercises at the end of the chapter illustrate some of the calculations for the AR(2) model.

Example 4.17. Simulated values from AR(2) process.

To be completed

4.3.3 Autoregressive Moving Average Models

Autoregressive and moving averages model can be combined into a general model called the autoregressive moving average (ARMA) model. The ARMA model with p autoregressive components and q moving average components, denoted ARMA(p,q) is given by

$$\begin{aligned} Y_t - \mu &= \phi_1(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) \\ &\quad + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q} \\ \varepsilon_t &\sim \text{WN}(0, \sigma^2) \end{aligned}$$

The regression formulation is

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta \varepsilon_{t-1} + \cdots + \theta \varepsilon_{t-q}$$

where $c = \mu / (1 - \phi_1 - \cdots - \phi_p) = \mu / (1 - \phi)$ and $\phi = \phi_1 + \cdots + \phi_p$. This model combines aspects of the pure moving average models and the pure autoregressive models and can capture many types of autocorrelation patterns. For modeling typical non-seasonal economic and financial data, it is seldom necessary to consider models in which $p > 2$ and $q > 2$.

4.3.4 Vector Autoregressive Models

The most popular multivariate time series model is the *vector autoregressive* (VAR) model. The VAR model is a multivariate extension of the univariate autoregressive model (4.11). For example, a bivariate VAR(1) model for $\mathbf{Y}_t = (Y_{1t}, Y_{2t})'$ has the form

$$\begin{pmatrix} Y_{1t} \\ Y_{2t} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_{11}^1 & a_{12}^1 \\ a_{21}^1 & a_{22}^1 \end{pmatrix} \begin{pmatrix} Y_{1t-1} \\ Y_{2t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix},$$

or

$$\begin{aligned} Y_{1t} &= c_1 + a_{11}^1 Y_{1t-1} + a_{12}^1 Y_{2t-1} + \varepsilon_{1t}, \\ Y_{2t} &= c_2 + a_{21}^1 Y_{1t-1} + a_{22}^1 Y_{2t-1} + \varepsilon_{2t}, \end{aligned}$$

where

$$\begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \sim \text{iid } N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} \right).$$

In the equations for Y_1 and Y_2 , the lagged values of both Y_1 and Y_2 are present. Hence, the VAR(1) model allows for dynamic feedback between Y_1 and Y_2 and can capture cross-lag correlations between the variables. In matrix notation, the model is

$$\begin{aligned} \mathbf{Y}_t &= \mathbf{A} \mathbf{Y}_{t-1} + \varepsilon_t, \\ \varepsilon_t &\sim N(\mathbf{0}, \boldsymbol{\Sigma}), \end{aligned}$$

where

$$\mathbf{A} = \begin{pmatrix} a_{11}^1 & a_{12}^1 \\ a_{21}^1 & a_{22}^1 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}.$$

The general VAR(p) model for $\mathbf{Y}_t = (Y_{1t}, Y_{2t}, \dots, Y_{nt})'$ has the form

$$\mathbf{Y}_t = \mathbf{c} + \mathbf{A}_1 \mathbf{Y}_{t-1} + \mathbf{A}_2 \mathbf{Y}_{t-2} + \cdots + \mathbf{A}_p \mathbf{Y}_{t-p} + \varepsilon_t,$$

where \mathbf{A}_i are $(n \times n)$ coefficient matrices and ε_t is an $(n \times 1)$ unobservable zero mean white noise vector process with covariance matrix Σ . VAR models are capable of capturing much of the complicated dynamics observed in stationary multivariate time series.

Example 4.18. Simulated values from a bivariate VAR(1) process.

To be completed

4.4 Further Reading

This chapter gives a very brief introduction to time series modeling. More thorough treatments of time series analysis with an orientation towards economics and finance with examples in R are given in Ruppert and Matteson (2016), Tsay (2010), and Zivot (2016). The CRAN task view for Time Series is an excellent summary of R packages used for time series analysis.

4.5 Exercises

Exercise 4.1. Suppose the time series $\{X_t\}_{t=-\infty}^{\infty}$ is independent white noise. That is $X_t \sim i.i.d. (0, \sigma^2)$. Define two new time series $\{Y_t\}_{t=-\infty}^{\infty}$ and $\{Z_t\}_{t=-\infty}^{\infty}$, where $Y_t = X_t^2$ and $Z_t = |X_t|$. Are $\{Y_t\}_{t=-\infty}^{\infty}$ and $\{Z_t\}_{t=-\infty}^{\infty}$ also independent white noise processes? Why or why not?

Exercise 4.2. Realizations from four stochastic processes are given in Figure . Which processes appear to be covariance stationary and which processes appear to be non-stationary? Briefly justify your answers.

Exercise 4.3. Consider the MA(1) model

$$\begin{aligned} Y_t &= 0.05 + \varepsilon_t + \theta \varepsilon_{t-1}, \quad -1 < \theta < 1 \\ \varepsilon_t &\sim iid N(0, (0.10)^2). \end{aligned}$$

This process has mean $E[Y_t] = 0.05$.

1. Calculate $\text{var}(Y_t)$ and $\rho_1 = \text{cor}(Y_t, Y_{t-1})$ for $\theta = 0.5$ and $\theta = 0.9$.
2. Using the R function `arima.sim()`, simulate and plot 250 observations of the MA(1) process with $\theta = 0.5$ and $\theta = 0.9$. Briefly comment on the behavior of the simulated data series. Does it look covariance stationary? Does it show evidence of time dependence?

Exercise 4.4. Consider the AR(1) model

$$\begin{aligned} Y_t - 0.05 &= \phi(Y_{t-1} - 0.05) + \varepsilon_t, \quad -1 < \phi < 1 \\ \varepsilon_t &\sim iid N(0, (0.10)^2). \end{aligned}$$

This process has mean $E[Y_t] = 0.05$.

1. Calculate $\text{var}(Y_t)$ for $\phi = 0.5$ and $\phi = 0.9$.
2. Calculate $\rho_j = \text{cor}(Y_t, Y_{t-j})$ for $\phi = 0.5$ and $\phi = 0.9$ and for $j = 1, \dots, 5$.
3. Using the R function `arima.sim()`, simulate and plot 250 observations of the AR(1) process with $\theta = 0.5$ and $\theta = 0.9$. Briefly comment on the behavior of the simulated data series. Does it look covariance stationary? Does it show evidence of time dependence? How is it different from the MA(1) process

Exercise 4.5. Figure shows a realization of a stochastic process representing a monthly time series of overlapping 2-month continuously compounded returns $r_t(2) = r_t + r_{t-1}$, where the 1-month continuously compounded returns r_t follow a Gaussian White noise process with variance 1.

1. Based on the sample autocorrelations, which time series process is most appropriate for describing the series: MA(1) or AR(1)? Justify your answer.
2. If you think the process is an AR(1) process, what do you think is the value of the autoregressive parameter? If you think the process is a MA(1) process, what do you think is the value of the moving average parameter?

Chapter 5

Descriptive Statistics for Financial Data

Updated: July 22, 2016

Copyright © Eric Zivot 2015, 2016

In this chapter we use graphical and numerical descriptive statistics to study the distribution and dependence properties of daily and monthly asset returns on a number of representative assets. The purpose of this chapter is to introduce the techniques of exploratory data analysis for financial time series and to document a set of stylized facts for monthly and daily asset returns that will be used in later chapters to motivate probability models for asset returns.

This chapter is organized as follows. Section 5.1 introduces the example data and reviews graphical and numerical descriptive statistics for univariate data. Graphical descriptive statistics include time plots, histograms, QQ-plots and boxplots. Numerical statistics are the sample statistics associated with the analogous characteristics of univariate distributions. For the example data, it is shown that monthly returns are approximately normally distributed but that daily returns have much fatter tails than the normal distribution. Section 5.2 covers univariate time series descriptive statistics including sample autocovariances, autocorrelations and the sample autocorrelation function. For the example data, it is shown that daily and monthly asset returns are uncorrelated but that daily returns exhibit a nonlinear time dependence related to time varying volatility. Section 5.3 presents bivariate descriptive statistics including scatterplots, sample covariance and correlation, and sample cross-lag covariances and correlations. Section 5.4 concludes with a summary of some stylized facts for monthly and daily returns.

The R packages used in this chapter are **corrplot**, **IntroCompFinR**, **PerformanceAnalytics**, **tseries**, **zoo** and **xts**. Make sure these packages are installed and loaded before running the R examples in the chapter.

5.1 Univariate Descriptive Statistics

Let $\{R_t\}$ denote a univariate time series of asset returns (simple or continuously compounded). Throughout this chapter we will assume that $\{R_t\}$ is a covariance stationary and ergodic stochastic process such that:

$$\begin{aligned} E[R_t] &= \mu \text{ independent of } t, \\ \text{var}(R_t) &= \sigma^2 \text{ independent of } t, \\ \text{cov}(R_t, R_{t-j}) &= \gamma_j \text{ independent of } t, \\ \text{corr}(R_t, R_{t-j}) &= \gamma_j / \sigma^2 = \rho_j \text{ independent of } t. \end{aligned}$$

In addition, we will assume that each R_t is identically distributed with unknown pdf $f_R(r)$.

An observed sample of size T of historical asset returns $\{r_t\}_{t=1}^T$ is assumed to be a realization from the stochastic process $\{R_t\}$ for $t = 1, \dots, T$. That is,

$$\{r_t\}_{t=1}^T = \{R_1 = r_1, \dots, R_T = r_T\}$$

The goal of *exploratory data analysis* (eda) is to use the observed sample $\{r_t\}_{t=1}^T$ to learn about the unknown pdf $f_R(r)$ as well as the time dependence properties of $\{R_t\}$.

5.1.1 Example data

We illustrate the descriptive statistical analysis of financial data using daily and monthly adjusted closing prices on Microsoft stock (ticker symbol `msft`) and the S&P 500 index (ticker symbol `^gspc`) over the period January 2, 1998 and May 31, 2012.¹ These data are obtained from `finance.yahoo.com` and are available in the R package **IntroCompFinR**. We first use the daily and monthly data to illustrate descriptive statistical analysis and to establish a number of stylized facts about the distribution and time dependence in daily and monthly returns.

Example 5.1. Constructing example data.

Daily adjusted closing price data on Microsoft and the S&P 500 index over the period January 4, 1993 through December 31, 2014 are available in the **IntroCompFinR** package as the “`xts`” objects `msftDailyPrices` and `sp500DailyPrices`, respectively.

```
data(msftDailyPrices, sp500DailyPrices)
str(msftDailyPrices)
```

¹ An adjusted closing price is adjusted for dividend payments and stock splits. Any dividend payment received between closing dates are added to the close price. If a stock split occurs between the closing dates then the all past prices are divided by the split ratio. The ticker symbol `^gspc` refers to the actual S&P 500 index, which is not a tradeable security. There are several mutual funds (e.g., Vanguard's S&P 500 fund with ticker `VFINF`) and exchange traded funds (e.g., State Street's SPDR S&P 500 ETF with ticker `SPY`) which track the S&P 500 index that are investable.

```
## An 'xts' object on 1993-01-04/2014-12-31 containing:
##   Data: num [1:5541, 1] 1.89 1.92 1.98 1.94 1.94 ...
##   - attr(*, "dimnames")=List of 2
##     ..$ : NULL
##     ..$ : chr "MSFT"
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##   NULL
```

We restrict the sample period to January 2, 1998 and May 31, 2012 using:

```
smp1 = "1998-01::2012-05"
msftDailyPrices = msftDailyPrices[smp1]
sp500DailyPrices = sp500DailyPrices[smp1]
```

End-of-month prices can be extracted from the daily prices using the **xts** function **to.monthly()**:

```
msftMonthlyPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
sp500MonthlyPrices = to.monthly(sp500DailyPrices, OHLC = FALSE)
```

It will also be convenient to create a merged “**xts**” object containing both the Microsoft and S&P 500 index prices:

```
msftSp500DailyPrices = merge(msftDailyPrices, sp500DailyPrices)
msftSp500MonthlyPrices = merge(msftMonthlyPrices, sp500MonthlyPrices)
```

We create “**xts**” objects containing simple returns using the **PerformanceAnalytics** function **Return.calculate()**:

```
msftMonthlyRetS = Return.calculate(msftMonthlyPrices, method = "simple")
msftDailyRetS = Return.calculate(msftDailyPrices, method = "simple")
sp500MonthlyRetS = Return.calculate(sp500MonthlyPrices, method = "simple")
sp500DailyRetS = Return.calculate(sp500DailyPrices, method = "simple")
msftSp500MonthlyRetS = Return.calculate(msftSp500MonthlyPrices, method = "simple")
msftSp500DailyRetS = Return.calculate(msftSp500DailyPrices, method = "simple")
```

We remove the first NA value of each object to avoid problems that some R functions have when missing values are encountered:

```
msftMonthlyRetS = msftMonthlyRetS[-1]
msftDailyRetS = msftDailyRetS[-1]
sp500MonthlyRetS = sp500MonthlyRetS[-1]
sp500DailyRetS = sp500DailyRetS[-1]
msftSp500MonthlyRetS = msftSp500MonthlyRetS[-1]
msftSp500DailyRetS = msftSp500DailyRetS[-1]
```

We also create “**xts**” objects containing monthly continuously compounded (cc) returns:

```
msftMonthlyRetC = log(1 + msftMonthlyRetS)
sp500MonthlyRetC = log(1 + sp500MonthlyRetS)
msftSp500MonthlyRetC = merge(msftMonthlyRetC, sp500MonthlyRetC)
```

5.1.2 Time plots

A natural graphical descriptive statistic for time series data is a *time plot*. This is simply a line plot with the time series data on the y-axis and the time index on the x-axis. Time plots are useful for quickly visualizing many features of the time series data.

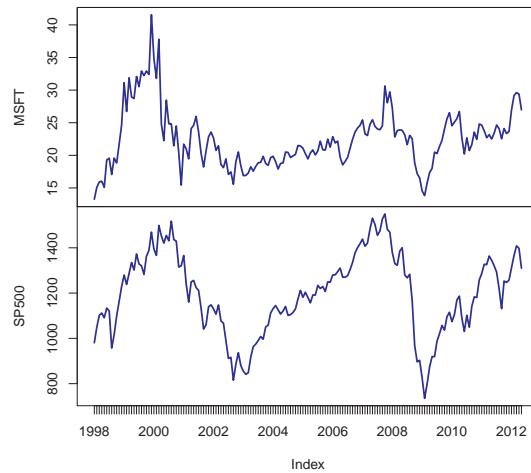


Fig. 5.1 End-of-month closing prices on Microsoft stock and the S&P 500 index.

Example 5.2. Time plots of monthly prices and returns.

A two-panel plot showing the monthly prices is given in Figure 5.1.2, and is created using the `plot` method for “zoo” objects:

```
plot.zoo(msftSp500MonthlyPrices, main = "", lwd = 2, col = "blue")
```

The prices exhibit random walk like behavior (no tendency to revert to a time independent mean) and appear to be non-stationary. Both prices show two large boom-bust periods associated with the dot-com period of the late 1990s and the run-up to the financial crisis of 2008. Notice the strong common trend behavior of the two price series.

A time plot for the monthly returns is created using:

```
my.panel <- function(...) {
  lines(...)
  abline(h = 0)
}
plot.zoo(msftSp500MonthlyRetS, main = "", panel = my.panel, lwd = 2, col = "blue")
```

and is given in Figure 5.1.2. The horizontal line at zero in each panel is created using the custom panel function `my.panel()` passed to `plot()`. In contrast to prices, returns show

clear mean-reverting behavior and the common monthly mean values look to be very close to zero. Hence, the common mean value assumption of covariance stationarity looks to be satisfied. However, the volatility (i.e., fluctuation of returns about the mean) of both series appears to change over time. Both series show higher volatility over the periods 1998 - 2003 and 2008 - 2012 than over the period 2003 - 2008. This is an indication of possible non-stationarity in volatility.² Also, the coincidence of high and low volatility periods across assets suggests a common driver to the time varying behavior of volatility. There does not appear to be any visual evidence of systematic time dependence in the returns. Later on we will see that the estimated autocorrelations are very close to zero. The returns for Microsoft and the S&P 500 tend to go up and down together suggesting a positive correlation.

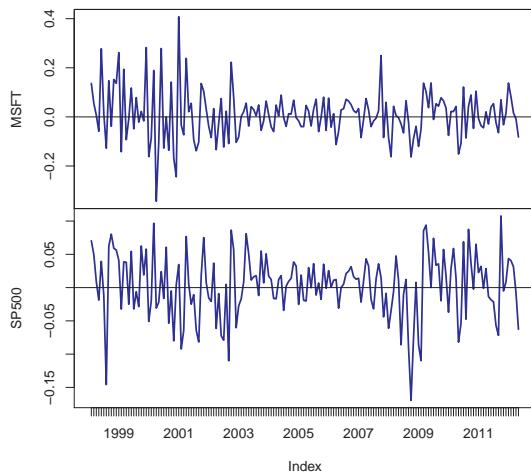


Fig. 5.2 Monthly continuously compounded returns on Microsoft stock and the S&P 500 index.

Example 5.3. Plotting returns on the same graph.

In Figure 5.1.2, the volatility of the returns on Microsoft and the S&P 500 looks to be similar but this is illusory. The y-axis scale for Microsoft is much larger than the scale for the S&P 500 index and so the volatility of Microsoft returns is actually much larger than the volatility of the S&P 500 returns. Figure 5.1.2 shows both returns series on the same time plot created using:

```
plot.zoo(msftSp500MonthlyRetS, plot.type = "single", main = "", col = c("red",
  "blue"), lty = c("dashed", "solid"), lwd = 2, ylab = "Returns")
abline(h = 0)
legend(x = "bottomright", legend = colnames(msftSp500MonthlyRetS), lty = c("dashed",
  "solid"), lwd = 2, col = c("red", "blue"))
```

² The returns can still be covariance stationary and exhibit time varying *conditional* volatility. This is explored in chapter 10.

Now the higher volatility of Microsoft returns, especially before 2003, is clearly visible. However, after 2008 the volatilities of the two series look quite similar. In general, the lower volatility of the S&P 500 index represents risk reduction due to holding a large diversified portfolio.

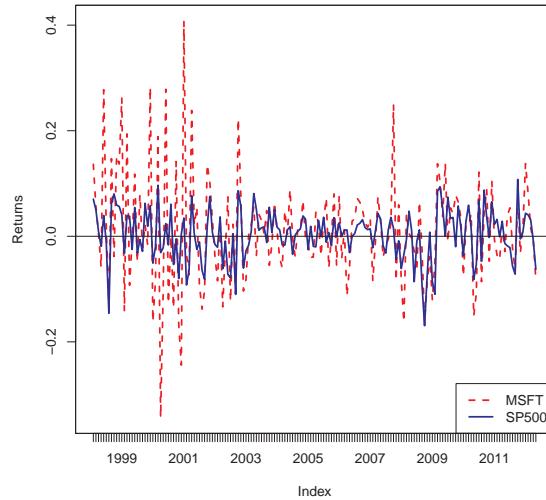


Fig. 5.3 Monthly continuously compounded returns for Microsoft and S&P 500 index on the same graph.

Example 5.4. Comparing simple and continuously compounded returns.

Figure 5.1.2 compares the simple and cc monthly returns for Microsoft created using:

```
retDiff = msftMonthlyRetS - msftMonthlyRetC
dataToPlot = merge(msftMonthlyRetS, msftMonthlyRetC, retDiff)
plot.zoo(dataToPlot, plot.type = "multiple", main = "", panel = my.panel, lwd = 2,
         col = c("black", "blue", "red"))
```

The top panel shows the simple returns, the middle panel shows the cc returns, and the bottom panel shows the difference between the simple and cc returns. Qualitatively, the simple and cc returns look almost identical. The main differences occur when the simple returns are large in absolute value (e.g., mid 2000, early 2001 and early 2008). In this case the difference between the simple and cc returns can be fairly large (e.g., as large as 0.08 in mid 2000). When the simple return is large and positive, the cc return is not quite as large; when the simple return is large and negative, the cc return is a larger negative number.

Example 5.5. Plotting daily returns.

Figure 5.1.2 shows the daily simple returns on Microsoft and the S&P 500 index created with:

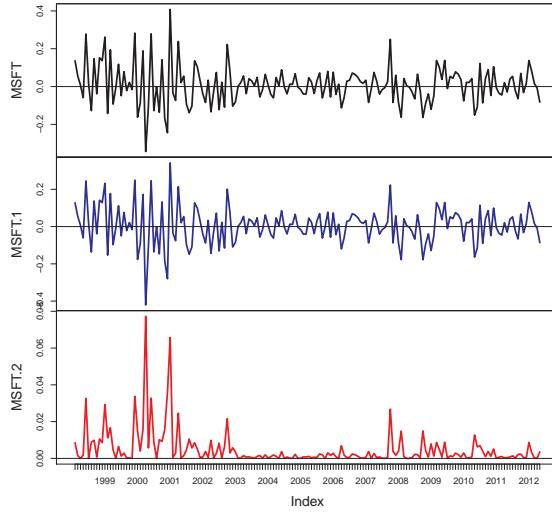


Fig. 5.4 Monthly simple and cc returns on Microsoft. Top panel: simple returns; middle panel: cc returns; bottom panel: difference between simple and cc returns.

```
plot.zoo(msftSp500DailyRetS, main = "", panel = my.panel, col = c("black", "blue"))
```

Compared with the monthly returns, the daily returns are closer to zero and the magnitude of the fluctuations (volatility) is smaller. However, the clustering of periods of high and low volatility is more pronounced in the daily returns. As with the monthly returns, the volatility of the daily returns on Microsoft is larger than the volatility of the S&P 500 returns. Also, the daily returns show some large and small “spikes” that represent unusually large (in absolute value) daily movements (e.g., Microsoft up 20% in one day and down 15% on another day). The monthly returns do not exhibit such extreme movements relative to typical volatility.

■

Equity curves

To directly compare the investment performance of two or more assets, plot the simple multi-period cumulative returns of each asset on the same graph. This type of graph, sometimes called an *equity curve*, shows how a one dollar investment amount in each asset grows over time. Better performing assets have higher equity curves. For simple returns, the k -period returns are $R_t(k) = \prod_{j=0}^{k-1} (1 + R_{t-j})$ and represent the growth of one dollar invested for k

periods. For continuously compounded returns, the k -period returns are $r_t(k) = \sum_{j=0}^{k-1} r_{t-j}$.

However, this cc k -period return must be converted to a simple k -period return, using $R_t(k) = \exp(r_t(k)) - 1$, to properly represent the growth of one dollar invested for k periods.

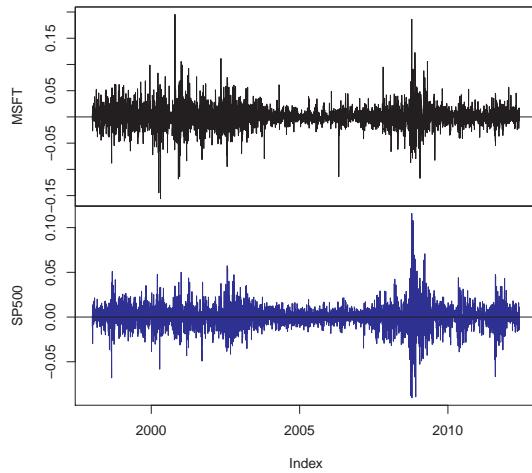


Fig. 5.5 Daily returns on Microsoft and the S&P 500 index.

Example 5.6. Equity curves for Microsoft and S&P 500 monthly returns.

To create the equity curves for Microsoft and the S&P 500 index based on simple returns use:³

```
equityCurveMsft = cumprod(1 + msftMonthlyRetS)
equityCurveSp500 = cumprod(1 + sp500MonthlyRetS)
dataToPlot = merge(equityCurveMsft, equityCurveSp500)
plot.zoo(dataToPlot, plot.type = "single", ylab = "Cumulative Returns", col = c("black",
  "blue"), lwd = 2)
legend(x = "topright", legend = c("MSFT", "SP500"), col = c("black", "blue"), lwd = 2)
```

The R function `cumprod()` creates the cumulative products needed for the equity curves. Figure 5.1.2 shows that a one dollar investment in Microsoft dominated a one dollar investment in the S&P 500 index over the given period. In particular, \$1 invested in Microsoft grew to about \$2.10 (over about 12 years) whereas \$1 invested in the S&P 500 index only grew to about \$1.40. Notice the huge increases and decreases in value of Microsoft during the dot-com bubble and bust over the period 1998 - 2001.

5.1.3 Descriptive statistics for the distribution of returns

In this section, we consider graphical and numerical descriptive statistics for the unknown marginal pdf, $f_R(r)$, of returns. Recall, we assume that the observed sample $\{r_t\}_{t=1}^T$ is a realization from a covariance stationary and ergodic time series $\{R_t\}$ where each R_t is a

³ You can also use the **PerformanceAnalytics** function `chart.CumReturns()`.

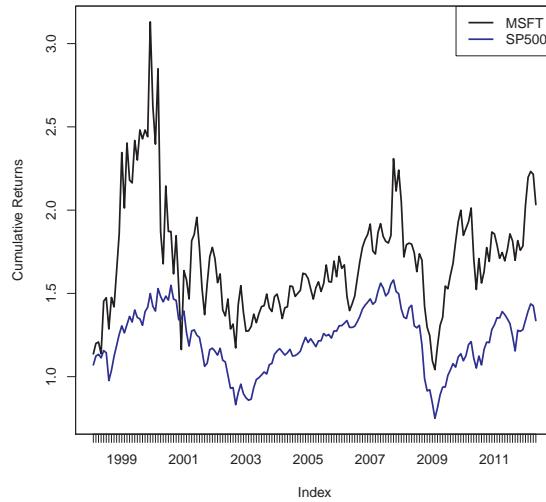


Fig. 5.6 Monthly cumulative continuously compounded returns on Microsoft and the S&P 500 index.

continuous random variable with common pdf $f_R(r)$. The goal is to use $\{r_t\}_{t=1}^T$ to describe properties of $f_R(r)$.

We study returns and not prices because prices are non-stationary. Sample descriptive statistics are only meaningful for covariance stationary and ergodic time series.

Histograms

A *histogram* of returns is a graphical summary used to describe the general shape of the unknown pdf $f_R(r)$. It is constructed as follows. Order returns from smallest to largest. Divide the range of observed values into N equally sized bins. Show the number or fraction of observations in each bin using a bar chart.

Example 5.7. Histograms for the daily and monthly returns on Microsoft and the S&P 500 index.

Figure 5.1.3 shows the histograms of the daily and monthly returns on Microsoft stock and the S&P 500 index created using the R function `hist()`:

```
par(mfrow = c(2, 2))
hist(msftMonthlyRetS, main = "", col = "cornflowerblue")
hist(msftDailyRetS, main = "", col = "cornflowerblue")
hist(sp500MonthlyRetS, main = "", col = "cornflowerblue")
hist(sp500DailyRetS, main = "", col = "cornflowerblue")
par(mfrow = c(1, 1))
```

All histograms have a bell-shape like the normal distribution. The histograms of daily returns are centered around zero and those for the monthly return are centered around values

slightly larger than zero. The bulk of the daily (monthly) returns for Microsoft and the S&P 500 are between -5% and 5% (-20% and 20%) and -3% and 3% (-10% and 10%), respectively. The histogram for the S&P 500 monthly returns is slightly skewed left (long left tail) due to more large negative returns than large positive returns whereas the histograms for the other returns are roughly symmetric.

When comparing two or more return distributions, it is useful to use the same bins for each histogram. Figure 5.1.3 shows the histograms for Microsoft and S&P 500 returns using the same 15 bins, created with the R code:

```
msftHist = hist(msftMonthlyRetS, plot = FALSE, breaks = 15)
par(mfrow = c(2, 2))
hist(msftMonthlyRetS, main = "", col = "cornflowerblue")
hist(msftDailyRetS, main = "", col = "cornflowerblue", breaks = msftHist$breaks)
hist(sp500MonthlyRetS, main = "", col = "cornflowerblue", breaks = msftHist$breaks)
hist(sp500DailyRetS, main = "", col = "cornflowerblue", breaks = msftHist$breaks)
par(mfrow = c(1, 1))
```

Using the same bins for all histograms allows us to see more clearly that the distribution of monthly returns is more spread out than the distributions for daily returns, and that the distribution of S&P 500 returns is more tightly concentrated around zero than the distribution of Microsoft returns.

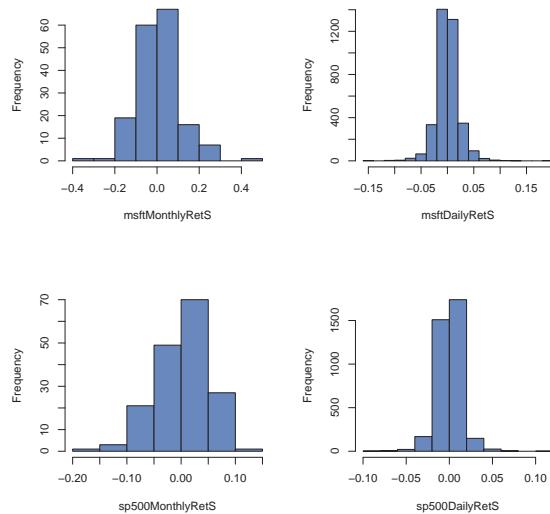


Fig. 5.7 Histograms of monthly continuously compounded returns on Microsoft stock and S&P 500 index.

Example 5.8. Are Microsoft returns normally distributed? A first look.

The shape of the histogram for Microsoft returns suggests that a normal distribution might be a good candidate for the unknown distribution of Microsoft returns. To investigate this

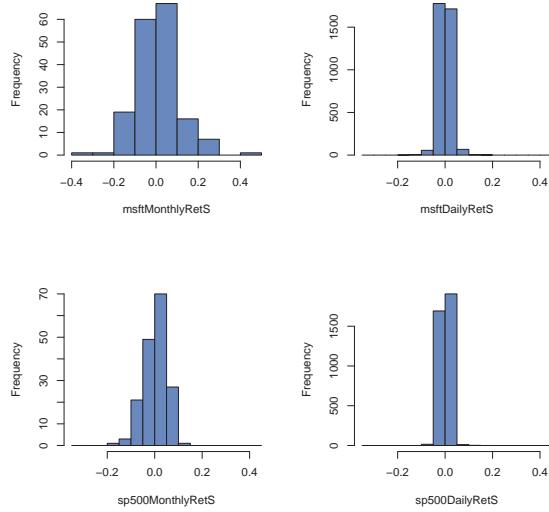


Fig. 5.8 Histograms for Microsoft and S&P 500 returns using the same bins.

conjecture, we simulate random returns from a normal distribution with mean and standard deviation calibrated to the Microsoft daily and monthly returns using:

```
set.seed(123)
gwnDaily = rnorm(length(msftDailyRetS), mean = mean(msftDailyRetS), sd = sd(msftDailyRetS))
gwnDaily = xts(gwnDaily, index(msftDailyRetS))
gwnMonthly = rnorm(length(msftMonthlyRetS), mean = mean(msftMonthlyRetS), sd = sd(msftMonthlyRetS))
gwnMonthly = xts(gwnMonthly, index(msftMonthlyRetS))
```

Figure 5.1.3 shows the Microsoft monthly returns together with the simulated normal returns created using:

```
par(mfrow = c(2, 2))
plot.zoo(msftMonthlyRetS, main = "Monthly Returns on MSFT", lwd = 2, col = "blue",
         ylim = c(-0.4, 0.4))
abline(h = 0)
plot.zoo(gwnMonthly, main = "Simulated Normal Returns", lwd = 2, col = "blue",
         ylim = c(-0.4, 0.4))
abline(h = 0)
hist(msftMonthlyRetS, main = "", col = "cornflowerblue", xlab = "returns")
hist(gwnMonthly, main = "", col = "cornflowerblue", xlab = "returns", breaks = msftHist$breaks)
par(mfrow = c(1, 1))
```

The simulated normal returns shares many of the same features as the Microsoft returns: both fluctuate randomly about zero. However, there are some important differences. In particular, the volatility of Microsoft returns appears to change over time (large before 2003, small between 2003 and 2008, and large again after 2008) whereas the simulated returns has constant volatility. Additionally, the distribution of Microsoft returns has fatter tails (more extreme large and small returns) than the simulated normal returns. Apart from these fea-

tures, the simulated normal returns look remarkably like the Microsoft monthly returns.

Figure 5.1.3 shows the Microsoft daily returns together with the simulated normal returns. The daily returns look much less like GWN than the monthly returns. Here the constant volatility of simulated GWN does not match the volatility patterns of the Microsoft daily returns, and the tails of the histogram for the Microsoft returns are “fatter” than the tails of the GWN histogram.

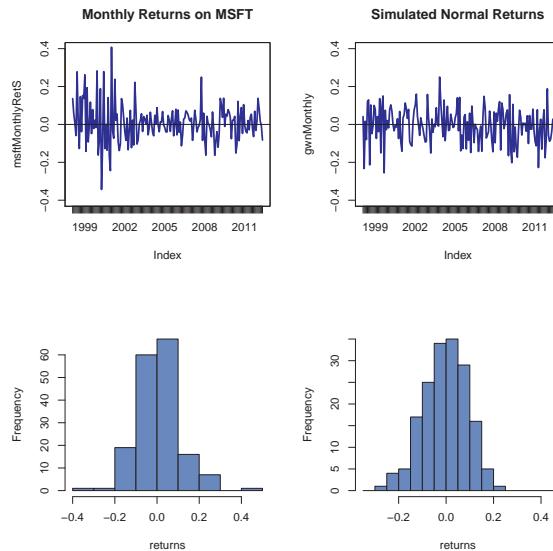


Fig. 5.9 Comparison of Microsoft monthly returns with simulated normal returns with the same mean and standard deviation as the Microsoft returns.

Smoothed histogram

Histograms give a good visual representation of the data distribution. The shape of the histogram, however, depends on the number of bins used. With a small number of bins, the histogram often appears blocky and fine details of the distribution are not revealed. With a large number of bins, the histogram might have many bins with very few observations. The `hist()` function in R smartly chooses the number of bins so that the resulting histogram typically looks good.

The main drawback of the histogram as descriptive statistic for the underlying pdf of the data is that it is discontinuous. If it is believed that the underlying pdf is continuous, it is desirable to have a continuous graphical summary of the pdf. The *smoothed histogram* achieves this goal. Given a sample of data $\{x_t\}_{t=1}^T$ the R function `density()` computes a smoothed estimate of the underlying pdf at each point x in the bins of the histogram using the formula:

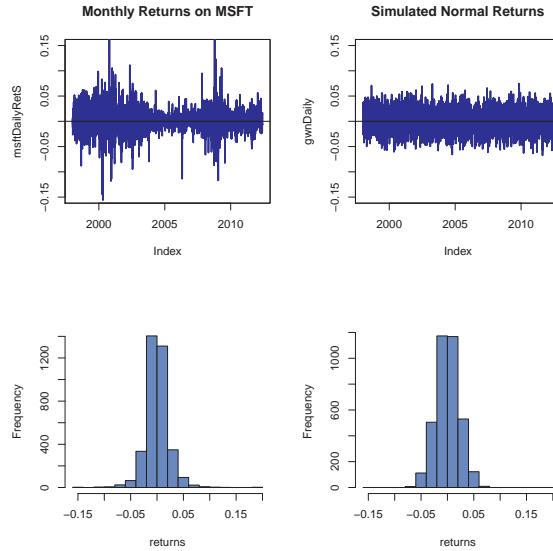


Fig. 5.10 Comparison of Microsoft daily returns with simulated normal returns with the same mean and standard deviation as the Microsoft returns.

$$\hat{f}_X(x) = \frac{1}{Tb} \sum_{t=1}^T k\left(\frac{x - x_t}{b}\right),$$

where $k(\cdot)$ is a continuous smoothing function (typically a standard normal distribution) and b is a bandwidth (or bin-width) parameter that determines the width of the bin around x in which the smoothing takes place. The resulting pdf estimate $\hat{f}_X(x)$ is a two-sided weighted average of the histogram values around x .

Example 5.9. Smoothed histogram for Microsoft monthly returns.

Figure 5.1.3 shows the histogram of Microsoft returns overlaid with the smoothed histogram created using:

```
MSFT.density = density(msftMonthlyRets)
hist(msftMonthlyRets, main = "", xlab = "Microsoft Monthly Returns", col = "cornflowerblue",
     probability = TRUE, ylim = c(0, 5.5))
points(MSFT.density, type = "l", col = "orange", lwd = 2)
```

In Figure 5.1.3, the histogram is normalized (using the argument `probability=TRUE`), so that its total area is equal to one. The smoothed density estimate transforms the blocky shape of the histogram into a smooth continuous graph. ■

Empirical CDF

Recall, the CDF of a random variable X is the function $F_X(x) = \Pr(X \leq x)$. The *empirical CDF* of a data sample $\{x_t\}_{t=1}^T$ is the function that counts the fraction of observations less

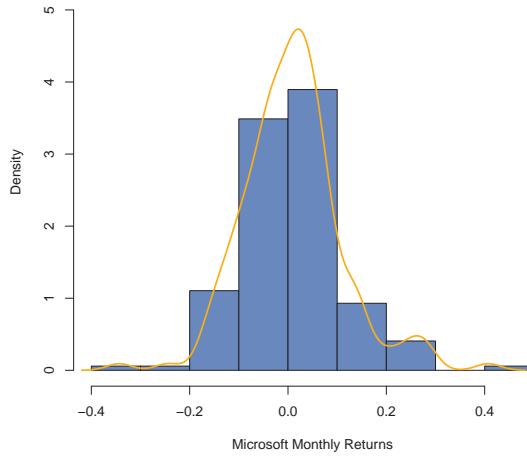


Fig. 5.11 Histogram and smoothed density estimate for the monthly returns on Microsoft.

than or equal to x :

$$\begin{aligned}\hat{F}_X(x) &= \frac{1}{T}(\#x_i \leq x) \\ &= \frac{\text{number of } x_i \text{ values } \leq x}{\text{sample size}}\end{aligned}\tag{5.1}$$

Example 5.10. Empirical CDF for monthly returns on Microsoft.

Computing the empirical CDF of Microsoft monthly returns at a given point, say $R = 0$, is a straightforward calculation in R:

```
Fhat.0 = sum(msftMonthlyRetS <= 0)/nrow(msftMonthlyRetS)
Fhat.0
## [1] 0.4709
```

Here, the expression `msftMonthlyRetS <= 0` creates a logical vector the same length as `msftMonthlyRetS` that is equal to `TRUE` when returns are less than or equal to zero and `FALSE` when returns are greater than zero. Then `Fhat.0` is equal to the fraction of `TRUE` values (returns less than or equal to zero) in the data, which gives (5.1) evaluated at zero. The R function `ecdf()` can be used to compute and plot (5.1) for all of the observed returns. For example, Figure 5.12 shows the empirical CDF for Microsoft monthly returns computed using:

```
plot(ecdf(coredata(msftMonthlyRetS)), main = "", col = "blue")
```

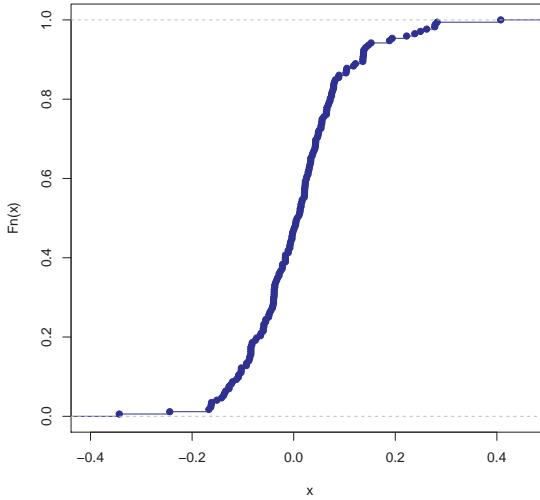


Fig. 5.12 Empirical CDF of monthly returns on Microsoft

Empirical quantiles/percentiles

Recall, for $\alpha \in (0, 1)$ the $\alpha \times 100\%$ quantile of the distribution of a continuous random variable X with CDF F_X is the point q_α^X such that $F_X(q_\alpha^X) = \Pr(X \leq q_\alpha^X) = \alpha$. Accordingly, the $\alpha \times 100\%$ *empirical quantile* (or $100 \times \alpha^{\text{th}}$ percentile) of a data sample $\{x_t\}_{t=1}^T$ is the data value \hat{q}_α such that $\alpha \cdot 100\%$ of the data are less than or equal to \hat{q}_α . Empirical quantiles can be easily determined by ordering the data from smallest to largest giving the ordered sample (also known as *order statistics*):

$$x_{(1)} < x_{(2)} < \cdots < x_{(T)}.$$

The empirical quantile \hat{q}_α is the order statistic closest to $\alpha \times T$.⁴

The *empirical quartiles* are the empirical quantiles for $\alpha = 0.25, 0.5$ and 0.75 , respectively. The second empirical quartile $\hat{q}_{.50}$ is called the *sample median* and is the data point such that half of the data is less than or equal to its value. The *interquartile range* (IQR) is the difference between the 3rd and 1st quartile:

$$\text{IQR} = \hat{q}_{.75} - \hat{q}_{.25},$$

and shows the size of the middle of the data distribution.

Example 5.11. Empirical quantiles of the Microsoft and S&P 500 monthly returns.

The R function `quantile()` computes empirical quantiles for a single data series. By default, `quantile()` returns the empirical quartiles as well as the minimum and maximum values:

⁴ There is no unique way to determine the empirical quantile from a sample of size N for all values of α . The R function `quantile()` can compute empirical quantile using one of seven different definitions.

```
quantile(msftMonthlyRetS)

##      0%      25%      50%      75%     100%
## -0.343386 -0.048831  0.008544  0.056980  0.407647

quantile(sp500MonthlyRetS)

##      0%      25%      50%      75%     100%
## -0.16942 -0.02094  0.00809  0.03532  0.10772
```

The left (right) quantiles of the Microsoft monthly returns are smaller (larger) than the respective quantiles for the S&P 500 index.

To compute quantiles for a specified α use the `probs` argument. For example, to compute the 1% and 5% quantiles of the monthly returns use:

```
quantile(msftMonthlyRetS, probs = c(0.01, 0.05))

##      1%      5%
## -0.1893 -0.1369

quantile(sp500MonthlyRetS, probs = c(0.01, 0.05))

##      1%      5%
## -0.12040 -0.08184
```

Here we see that 1% of the Microsoft cc returns are less than -0.1893 and 5% of the returns are less than -0.1369 , respectively. For the S&P 500 returns, these values are -0.1204 and -0.0818 , respectively.

To compute the median and IQR values for monthly returns use the R functions `median()` and `IQR()`, respectively:

```
apply(msftSp500MonthlyRetS, 2, median)

##      MSFT      SP500
## 0.008544 0.008090

apply(msftSp500MonthlyRetS, 2, IQR)

##      MSFT      SP500
## 0.10581 0.05626
```

The median returns are similar (about 0.8% per month) but the IQR for Microsoft is about twice as large as the IQR for the S&P 500 index.

Historical/Empirical VaR

Recall, the $\alpha \times 100\%$ value-at-risk (VaR) of an investment of $\$W$ is $\text{VaR}_\alpha = \$W \times q_\alpha^R$, where q_α^R is the $\alpha \times 100\%$ quantile of the probability distribution of the investment simple rate

of return R . The $\alpha \times 100\%$ *historical VaR* (sometimes called *empirical VaR* or *Historical Simulation VaR*) of an investment of $\$W$ is defined as:

$$\text{VaR}_{\alpha}^{HS} = \$W \times \hat{q}_{\alpha}^R,$$

where \hat{q}_{α}^R is the empirical α quantile of a sample of simple returns $\{R_t\}_{t=1}^T$. For a sample of continuously compounded returns $\{r_t\}_{t=1}^T$ with empirical α quantile \hat{q}_{α}^r ,

$$\text{VaR}_{\alpha}^{HS} = \$W \times (\exp(\hat{q}_{\alpha}^r) - 1).$$

Historical VaR is based on the distribution of the observed returns and not on any assumed distribution for returns (e.g., the normal distribution).

Example 5.12. Using empirical quantiles to compute historical Value-at-Risk.

Consider investing $W = \$100,000$ in Microsoft and the S&P 500 over a month. The 1% and 5% historical VaR values for these investments based on the historical samples of monthly returns are:

```
W = 1e+05
msftQuantiles = quantile(msftMonthlyRetS, probs = c(0.01, 0.05))
sp500Quantiles = quantile(sp500MonthlyRetS, probs = c(0.01, 0.05))
msftVaR = W * msftQuantiles
sp500VaR = W * sp500Quantiles
msftVaR

##      1%      5%
## -18929 -13694

sp500VaR

##      1%      5%
## -12040 -8184
```

Based on the empirical distribution of the monthly returns, a \$100,000 monthly investment in Microsoft will lose \$13694 or more with 5% probability and will lose \$18928 or more with 1% probability. The corresponding values for the S&P 500 are \$8183 and \$12039, respectively. The historical VaR values for the S&P 500 are considerably smaller than those for Microsoft. In this sense, investing in Microsoft is a riskier than investing in the S&P 500 index.

5.1.4 QQ-plots

Often it is of interest to see if a given data sample could be viewed as a random sample from a specified probability distribution. One easy and effective way to do this is to compare the empirical quantiles of a data sample to those from a reference probability distribution. If the quantiles match up, then this provides strong evidence that the reference distribution

is appropriate for describing the distribution of the observed data. If the quantiles do not match up, then the observed differences between the empirical quantiles and the reference quantiles can be used to determine a more appropriate reference distribution. It is common to use the normal distribution as the reference distribution, but any distribution can, in principle be used.

The quantile-quantile plot (QQ-plot) gives a graphical comparison of the empirical quantiles of a data sample to those from a specified reference distribution. The QQ-plot is an xy-plot with the reference distribution quantiles on the x-axis and the empirical quantiles on the y-axis. If the quantiles exactly match up then the QQ-plot is a straight line. If the quantiles do not match up, then the shape of the QQ-plot indicates which features of the data are not captured by the reference distribution.

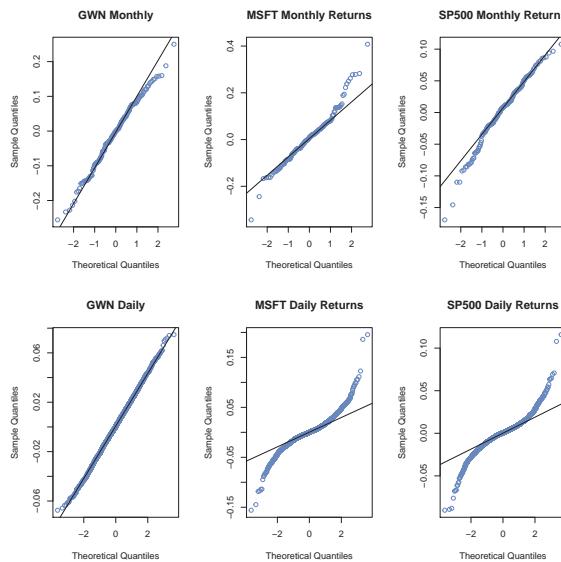


Fig. 5.13 Normal QQ-plots for GWN, Microsoft returns and S&P 500 returns.

Example 5.13. Normal QQ-plots for GWN, Microsoft and S&P 500 returns.

The R function `qqnorm()` creates a QQ-plot for a data sample using the normal distribution as the reference distribution. Figure 5.13 shows normal QQ-plots for the simulated GWN data, Microsoft returns and S&P 500 returns created using:

```
par(mfrow = c(2, 3))
qqnorm(gwnMonthly, main = "GWN Monthly", col = "cornflowerblue")
qqline(gwnMonthly)
qqnorm(msftMonthlyRetS, main = "MSFT Monthly Returns", col = "cornflowerblue")
qqline(msftMonthlyRetS)
qqnorm(sp500MonthlyRetS, main = "SP500 Monthly Returns", col = "cornflowerblue")
qqline(sp500MonthlyRetS)
qqnorm(gwnDaily, main = "GWN Daily", col = "cornflowerblue")
```

```
qqline(gwnDaily)
qqnorm(msftDailyRetS, main = "MSFT Daily Returns", col = "cornflowerblue")
qqline(msftDailyRetS)
qqnorm(sp500DailyRetS, main = "SP500 Daily Returns", col = "cornflowerblue")
qqline(sp500DailyRetS)
par(mfrow = c(1, 1))
```

The normal QQ-plot for the simulated GWN data is very close to a straight line, as it should be since the data are simulated from a normal distribution. The `qqline()` function draws a straight line through the points to help determine if the quantiles match up. The normal QQ-plots for the Microsoft and S&P 500 monthly returns are linear in the middle of the distribution but deviate from linearity in the tails of the distribution. In the normal QQ-plot for Microsoft monthly returns, the theoretical normal quantiles on the x-axis are too small in both the left and right tails because the points fall below the straight line in the left tail and fall above the straight line in the right tail. Hence, the normal distribution does not match the empirical distribution of Microsoft monthly returns in the extreme tails of the distribution. In other words, the Microsoft returns have fatter tails than the normal distribution. For the S&P 500 monthly returns, the theoretical normal quantiles are too small only for the left tail of the empirical distribution of returns (points fall below the straight line in the left tail only). This reflects the long left tail (negative skewness) of the empirical distribution of S&P 500 returns. The normal QQ-plots for the daily returns on Microsoft and the S&P 500 index both exhibit a pronounced tilted S-shape with extreme departures from linearity in the left and right tails of the distributions. The daily returns clearly have much fatter tails than the normal distribution.



Example 5.14. Student's t QQ-plot for Microsoft returns.

The function `qqPlot()` from the `car` package or the function `chart.QQPlot()` from the **PerformanceAnalytics** package can be used to create a QQ-plot against any reference distribution that has a corresponding quantile function implemented in R. For example, a QQ-plot for the Microsoft returns using a Student's t reference distribution with 5 degrees of freedom can be created using:

```
library(car)
qqPlot(coredata(msftMonthlyRetS), distribution = "t", df = 5, ylab = "MSFT quantiles",
envelope = FALSE)
```

In the function `qqPlot()`, the argument `distribution="t"` specifies that the quantiles are to be computed using the R function `qt()`.⁵ Figure 5.1.4 shows the resulting graph. Here, with a reference distribution with fatter tails than the normal distribution the QQ-plot for Microsoft returns is closer to a straight line. This indicates that the Student's t distribution with 5 degrees of freedom is a better reference distribution for Microsoft returns than the normal distribution.



⁵ The `coredata()` function is used to extract the data as a "numeric" object because `qqPlot()` does not work correctly with "zoo" or "xts" objects.

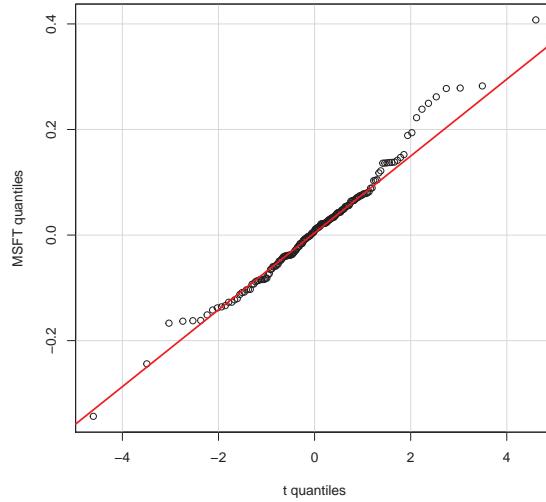


Fig. 5.14 QQ-plot of Microsoft returns using Student's t distribution with 5 degrees of freedom as the reference distribution.

5.1.5 Shape Characteristics of the Empirical Distribution

Recall, for a random variable X the measures of center, spread, asymmetry and tail thickness of the pdf are:

$$\begin{aligned} \text{center} &: \mu_X = E[X], \\ \text{spread} &: \sigma_X^2 = \text{var}(X) = E[(X - \mu_X)^2], \\ \text{spread} &: \sigma_X = \sqrt{\text{var}(X)} \\ \text{asymmetry} &: \text{skew}_X = E[(X - \mu_X)^3]/\sigma^3, \\ \text{tail thickness} &: \text{kurt}_X = E[(X - \mu_X)^4]/\sigma^4. \end{aligned}$$

The corresponding shape measures for the empirical distribution (e.g., as measured by the histogram) of a data sample $\{x_t\}_{t=1}^T$ are the sample statistics:⁶

⁶ Values with hats, “ $\hat{\cdot}$ ”, denote sample estimates of the corresponding population quantity. For example, the sample mean $\hat{\mu}_x$ is the sample estimate of the population expected value μ_X .

$$\hat{\mu}_x = \bar{x} = \frac{1}{T} \sum_{t=1}^T x_t, \quad (5.2)$$

$$\hat{\sigma}_x^2 = s_x^2 = \frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^2, \quad (5.3)$$

$$\hat{\sigma}_x = \sqrt{\hat{\sigma}_x^2}, \quad (5.4)$$

$$\widehat{\text{skew}}_x = \frac{\frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^3}{s_x^3}, \quad (5.5)$$

$$\widehat{\text{kurt}}_x = \frac{\frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})^4}{s_x^4}. \quad (5.6)$$

The sample mean, $\hat{\mu}_X$, measures the center of the histogram; the sample standard deviation, $\hat{\sigma}_x$, measures the spread of the data about the mean in the same units as the data; the sample skewness, $\widehat{\text{skew}}_x$, measures the asymmetry of the histogram; the sample kurtosis, $\widehat{\text{kurt}}_x$, measures the tail-thickness of the histogram. The sample excess kurtosis, defined as the sample kurtosis minus 3:

$$\widehat{\text{ekurt}}_x = \widehat{\text{kurt}}_x - 3, \quad (5.7)$$

measures the tail thickness of the data sample relative to that of a normal distribution.

Notice that the divisor in (5.3)-(5.6) is $T - 1$ and not T . This is called a *degrees-of-freedom* correction. In computing the sample variance, skewness and kurtosis, one degree-of-freedom in the sample is used up in the computation of the sample mean so that there are effectively only $T - 1$ observations available to compute the statistics.⁷

Example 5.15. Sample shape statistics for the returns on Microsoft and S&P 500.

The R functions for computing (5.2) - (5.6) are `mean()`, `var()` and `sd()`, respectively. There are no functions for computing (5.5) and (5.6) in base R. The functions `skewness()` and `kurtosis()` in the **PerformanceAnalytics** package compute (5.5) and the sample excess kurtosis (5.7), respectively.⁸ The sample statistics for the Microsoft and S&P 500 monthly returns are:

```
statsMonthly = rbind(apply(msftSp500MonthlyRetS, 2, mean), apply(msftSp500MonthlyRetS,
  2, var), apply(msftSp500MonthlyRetS, 2, sd), apply(msftSp500MonthlyRetS, 2,
  skewness), apply(msftSp500MonthlyRetS, 2, kurtosis))
rownames(statsMonthly) = c("Mean", "Variance", "Std Dev", "Skewness", "Excess Kurtosis")
round(statsMonthly, digits = 4)

##          MSFT      SP500
## Mean      0.0092   0.0028
## Variance  0.0103   0.0023
## Std Dev   0.1015   0.0478
## Skewness   0.4853  -0.5631
## Excess Kurtosis 1.9148  0.6545
```

⁷ If there is only one observation in the sample then it is impossible to create a measure of spread in the sample. You need at least two observations to measure deviations from the sample average. Hence the effective sample size for computing the sample variance is $T - 1$.

⁸ Similar functions are available in the **moments** package.

The mean and standard deviation for Microsoft monthly returns are 0.9% and 10%, respectively. Annualized, these values are approximately 10.8% (.009 \times 12) and 34.6% (.10 \times $\sqrt{12}$), respectively. The corresponding monthly and annualized values for S&P 500 returns are .3% and 4.7%, and 4.8% and 16.6%, respectively. Microsoft has a higher mean and volatility than S&P 500. The lower volatility for the S&P 500 reflects risk reduction due to diversification. The sample skewness for Microsoft, 0.485, is slightly positive and reflects the approximate symmetry in the histogram in Figure 5.1.3. The skewness for S&P 500, however, is moderately negative at -0.563 which reflects the somewhat long left tail of the histogram in Figure 5.1.3. The sample excess kurtosis values for Microsoft and S&P 500 are 1.915 and 0.654, respectively, and indicate that the tails of the histograms are slightly fatter than the tails of a normal distribution.

The sample statistics for the daily returns are:

```
statsDaily = rbind(apply(msftSp500DailyRetS, 2, mean), apply(msftSp500DailyRetS,
  2, var), apply(msftSp500DailyRetS, 2, sd), apply(msftSp500DailyRetS, 2, skewness),
  apply(msftSp500DailyRetS, 2, kurtosis))
rownames(statsDaily) = rownames(statsMonthly)
round(statsDaily, digits = 4)

##           MSFT    SP500
## Mean      0.0005  0.0002
## Variance  0.0005  0.0002
## Std Dev   0.0217  0.0135
## Skewness   0.2506  0.0012
## Excess Kurtosis 7.3915 6.9971
```

For a daily horizon, the sample mean for both series is zero (to three decimals). As with the monthly returns, the sample standard deviation for Microsoft, 2.2%, is about twice as big as the sample standard deviation for the S&P 500 index, 1.4%. Neither daily return series exhibits much skewness, but the excess kurtosis values are much larger than the corresponding monthly values indicating that the daily returns have much fatter tails and are more non-normally distributed than the monthly return series.

The daily simple returns are very close to the daily cc returns so the square-root-of-time rule (with 20 trading days within the month) can be applied to the sample mean and standard deviation of the simple returns:

```
20 * statsDaily["Mean", ]

##      MSFT    SP500
## 0.009331 0.003448

sqrt(20) * statsDaily["Std Dev", ]

##      MSFT    SP500
## 0.09709 0.06029
```

Comparing these values to the corresponding monthly sample statistics shows that the square-root-of-time-rule applied to the daily mean and standard deviation gives results that are fairly close to the actual monthly mean and standard deviation.

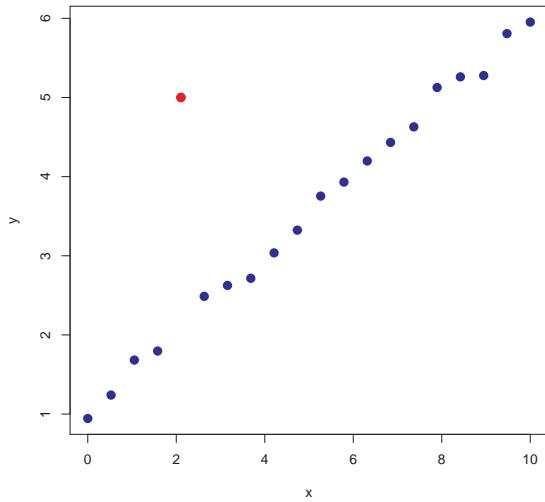


Fig. 5.15 Illustration of an “outlier” in a data sample.

5.1.6 Outliers

Figure 5.1.5 nicely illustrates the concept of an outlier in a data sample. All of the points are following a nice systematic relationship except one - the red dot “outlier”. Outliers can be thought of in two ways. First, an outlier can be the result of a data entry error. In this view, the outlier is not a valid observation and should be removed from the data sample. Second, an outlier can be a valid data point whose behavior is seemingly unlike the other data points. In this view, the outlier provides important information and should not be removed from the data sample. For financial market data, outliers are typically extremely large or small values that could be the result of a data entry error (e.g. price entered as 10 instead of 100) or a valid outcome associated with some unexpected bad or good news. Outliers are problematic for data analysis because they can greatly influence the value of certain sample statistics.

Example 5.16. Effect of outliers on sample statistics.

To illustrate the impact of outliers on sample statistics, the simulated monthly GWN data is polluted by a single large negative outlier:

```
gwnMonthlyOutlier = gwnMonthly
gwnMonthlyOutlier[20] = -sd(gwnMonthly) * 6
```

Figure 5.1.6 shows the resulting data. Visually, the outlier is much smaller than a typical negative observation and creates a pronounced asymmetry in the histogram. Table 5.1

Statistic	GWN	GWN with Outlier	% Δ
Mean	-0.0054	-0.0072	32%
Variance	0.0086	0.0101	16%
Std. Deviation	0.0929	0.1003	8%
Skewness	-0.2101	-1.0034	378%
Kurtosis	2.7635	7.1530	159%
Median	-0.0002	-0.0002	0%
IQR	0.1390	0.1390	0%

Table 5.1 Sample statistics for GWN with and without outlier.

compares the sample statistics (5.2) - (5.6) of the unpolluted and polluted data. All of the sample statistics are influenced by the single outlier with the skewness and kurtosis being influenced the most. The mean decreases by 32% and the variance and standard deviation increase by 16% and 8%, respectively. The skewness changes by 378% and the kurtosis by 159%. Table 5.1 also shows the median and the IQR, which are quantile-based statistics for the center and spread, respectively. Notice that these statistics are unaffected by the outlier.

■

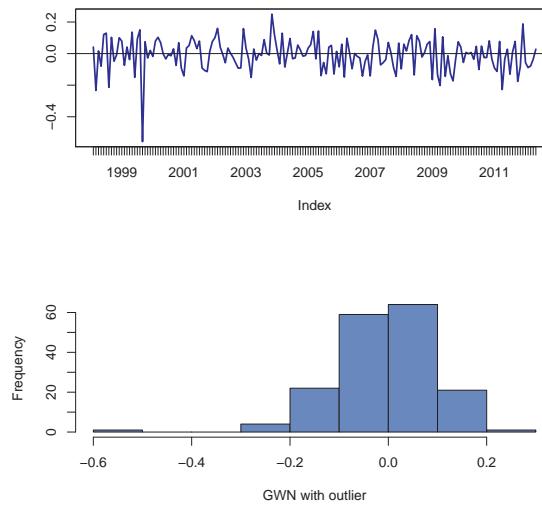


Fig. 5.16 Monthly GWN data polluted by a single outlier.

The previous example shows that the common sample statistics (5.2) - (5.6) based on the sample average and deviations from the sample average can be greatly influenced by a single outlier, whereas quantile-based sample statistics are not. Sample statistics that are not greatly influenced by a single outlier are called (outlier) *robust statistics*.

Defining outliers

A commonly used rule-of-thumb defines an outlier as an observation that is beyond the sample mean plus or minus three times the sample standard deviation. The intuition for this rule comes from the fact that if $X \sim N(\mu, \sigma^2)$ then $\Pr(\mu - 3\sigma \leq X \leq \mu + 3\sigma) \approx 0.99$. While the intuition for this definition is clear, the practical implementation of the rule using the sample mean, $\hat{\mu}$, and sample standard deviation, $\hat{\sigma}$, is problematic because these statistics can be heavily influenced by the presence of one or more outliers (as illustrated in the previous sub-section). In particular, both $\hat{\mu}$ and $\hat{\sigma}$ become larger (in absolute value) when there are outliers and this may cause the rule-of-thumb rule to miss identifying an outlier or to identify a non-outlier as an outlier. That is, the rule-of-thumb outlier detection rule is not robust to outliers!

A natural fix to the above outlier detection rule-of-thumb is to use outlier robust statistics, such as the median and the IQR, instead of the mean and standard deviation. To this end, a commonly used definition of a moderate outlier in the right tail of the distribution (value above the median) is a data point x such that:

$$\hat{q}_{.75} + 1.5 \cdot \text{IQR} < x < \hat{q}_{.75} + 3 \cdot \text{IQR}. \quad (5.8)$$

To understand this rule, recall that the IQR is where the middle 50% of the data lies. If the data were normally distributed then $\hat{q}_{.75} \approx \hat{\mu} + 0.67 \times \hat{\sigma}$ and $\text{IQR} \approx 1.34 \times \hat{\sigma}$. Then (5.8) is approximately:

$$\hat{\mu} + 2.67 \times \hat{\sigma} < x < \hat{\mu} + 4.67 \times \hat{\sigma}$$

Similarly, a moderate outlier in the left tail of the distribution (value below the median) is a data point such that,

$$\hat{q}_{.25} - 3 \cdot \text{IQR} < x < \hat{q}_{.25} - 1.5 \cdot \text{IQR}.$$

Extreme outliers are those observations x even further out in the tails of the distribution and are defined using:

$$\begin{aligned} x &> \hat{q}_{.75} + 3 \cdot \text{IQR}, \\ x &< \hat{q}_{.25} - 3 \cdot \text{IQR}. \end{aligned}$$

For example, with normally distributed data an extreme outlier would be an observation that is more than 4.67 standard deviations above the mean.

5.1.7 Box plots

A boxplot of a data series describes the distribution using outlier robust statistics. The basic boxplot is illustrated in Figure 5.17. The middle of the distribution is shown by a rectangular box whose top and bottom sides are approximately the first and third quartiles, $\hat{q}_{.25}$ and $\hat{q}_{.75}$, respectively. The line roughly in the middle of the box is the sample median,

$\hat{q}_{.5}$. The top whisker is a horizontal line that represents either the largest data value or $\hat{q}_{.75} + 1.5 \cdot IQR$, whichever is smaller. Similarly, the bottom whisker is located at either the smallest data value or $\hat{q}_{.25} - 1.5 \cdot IQR$, whichever is closer to $\hat{q}_{.25}$. Data values above the top whisker and below the bottom whisker are displayed as circles and represent extreme observations. If a data distribution is symmetric then the median is in the center of the box and the top and bottom whiskers are approximately the same distance from the box. Fat tailed distributions will have observations above and below the whiskers.

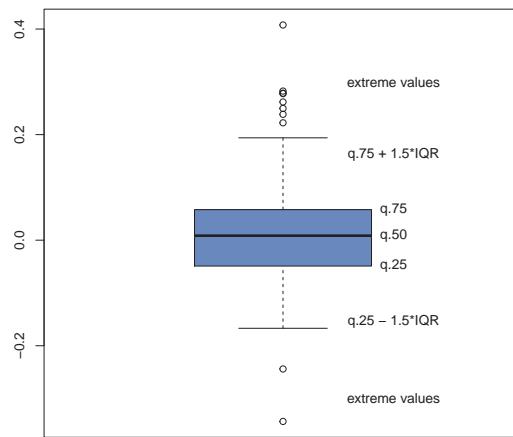


Fig. 5.17 Boxplot of return distribution.

Example 5.17. Boxplots of return distributions.

Boxplots are a great tool for visualizing the return distribution of one or more assets. Figure 5.1.7 shows the boxplots of the monthly continuously compounded and simple returns on Microsoft and the S&P500 index computed using:⁹

```
dataToPlot = merge(msftMonthlyRetS, msftMonthlyRetC, sp500MonthlyRetS, sp500MonthlyRetC)
colnames(dataToPlot) = c("msftRetS", "msftRetC", "sp500RetS", "sp500RetC")
boxplot(coredata(dataToPlot), main = "Simple vs. CC Monthly Returns", col = "cornflowerblue")
```

The Microsoft returns have both positive and negative outliers whereas the S&P 500 index returns only have negative outliers. The boxplots also clearly show that the distributions of the continuously compounded and simple returns are very similar except for the extreme tails of the distributions.

⁹ The `coredata()` function is used to extract the data from the `zoo` object because `boxplot()` doesn't like "`xts`" or "`zoo`" objects.

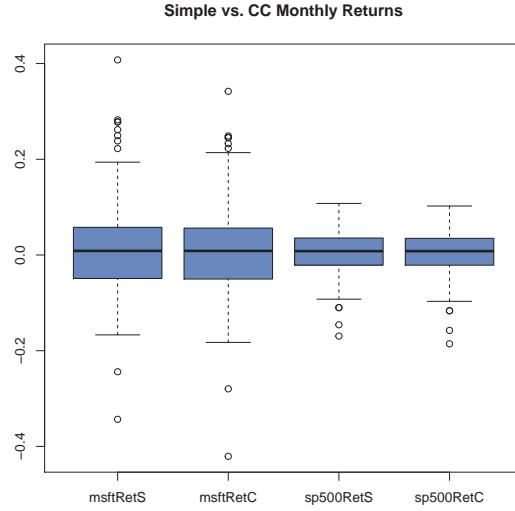


Fig. 5.18 Boxplots of the continuously compounded and simple monthly returns on Microsoft stock and the S&P 500 index.

5.2 Time Series Descriptive Statistics

In this section, we consider graphical and numerical descriptive statistics for summarizing the linear time dependence in a time series.

5.2.1 Sample Autocovariances and Autocorrelations

For a covariance stationary time series process $\{X_t\}$, the autocovariances $\gamma_j = \text{cov}(X_t, X_{t-j})$ and autocorrelations $\rho_j = \text{cov}(X_t, X_{t-j})/\sqrt{\text{var}(X_t)\text{var}(X_{t-j})} = \gamma_j/\sigma^2$ describe the linear time dependencies in the process. For a sample of data $\{x_t\}_{t=1}^T$, the linear time dependencies are captured by the sample autocovariances and autocorrelations:

$$\hat{\gamma}_j = \frac{1}{T-1} \sum_{t=j+1}^T (x_t - \bar{x})(x_{t-j} - \bar{x}), \quad j = 1, 2, \dots, T-J+1,$$

$$\hat{\rho}_j = \frac{\hat{\gamma}_j}{\hat{\sigma}^2}, \quad j = 1, 2, \dots, T-J+1,$$

where $\hat{\sigma}^2$ is the sample variance (5.3). The sample autocorrelation function (SACF) is a plot of $\hat{\rho}_j$ vs. j , and gives a graphical view of the liner time dependencies in the observed data.

Example 5.18. SACF for the Microsoft and S&P 500 returns.

The R function `acf()` computes and plots the sample autocorrelations $\hat{\rho}_j$ of a time series for a specified number of lags. For example, to compute $\hat{\rho}_j$ for $j = 1, \dots, 5$ for the monthly and daily returns on Microsoft use:

```

acf(coredatal(msftMonthlyRetS), lag.max = 5, plot = FALSE)

##
## Autocorrelations of series 'coredata(msftMonthlyRetS)', by lag
##
##    0     1     2     3     4     5
## 1.000 -0.193 -0.114  0.193 -0.139 -0.023

acf(coredatal(msftDailyRetS), lag.max = 5, plot = FALSE)

##
## Autocorrelations of series 'coredata(msftDailyRetS)', by lag
##
##    0     1     2     3     4     5
## 1.000 -0.044 -0.018 -0.010 -0.042  0.018

```

None of these sample autocorrelations are very big and indicate lack of linear time dependence in the monthly and daily returns.

Sample autocorrelations can also be visualized using the function `acf()`. Figure 5.2.1 shows the SACF for the monthly and daily returns on Microsoft and the S&P 500 created using:

```

par(mifrow = c(2, 2))
acf(coredatal(msftMonthlyRetS), main = "msftMonthlyRetS", lwd = 2)
acf(coredatal(sp500MonthlyRetS), main = "sp500MonthlyRetS", lwd = 2)
acf(coredatal(msftDailyRetS), main = "msftDailyRetS", lwd = 2)
acf(coredatal(sp500DailyRetS), main = "sp500DailyRetS", lwd = 2)
par(mifrow = c(1, 1))

```

None of the series show strong evidence for linear time dependence. The monthly returns have slightly larger sample autocorrelations at low lags than the daily returns. The dotted lines on the SACF plots are 5% critical values for testing the null hypothesis that the true autocorrelations ρ_j are equal to zero (see chapter 9). If $\hat{\rho}_j$ crosses the dotted line then the null hypothesis that $\rho_j = 0$ can be rejected at the 5% significance level.

■

5.2.2 Time dependence in volatility

The daily and monthly returns of Microsoft and the S&P 500 index do not exhibit much evidence for linear time dependence. Their returns are essentially uncorrelated over time. However, the lack of autocorrelation does not mean that returns are independent over time. There can be nonlinear time dependence. Indeed, there is strong evidence for a particular type of nonlinear time dependence in the daily returns of Microsoft and the S&P 500 index that is related to time varying volatility. This nonlinear time dependence, however, is less pronounced in the monthly returns.

Example 5.19. GWN and time independence.

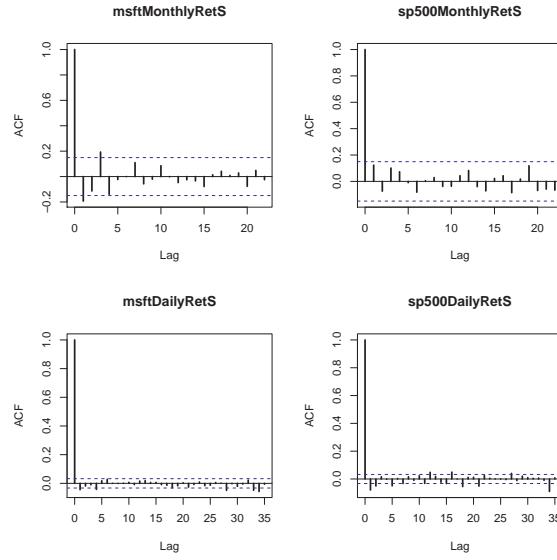


Fig. 5.19 SACFs for the monthly and daily returns on Microsoft and the S&P 500 index.

Let $y_t \sim \text{GWN}(0, \sigma^2)$. Then y_t is, by construction, independent over time. This implies that any function of y_t is also independent over time. To illustrate, Figure 5.2.2 shows y_t , y_t^2 and $|y_t|$ together with their SACFs where $y_t \sim \text{GWN}(0, 1)$ for $t = 1, \dots, 500$. Notice that all of the sample autocorrelations are essentially zero which confirms the lack of linear time dependence in y_t , y_t^2 and $|y_t|$.

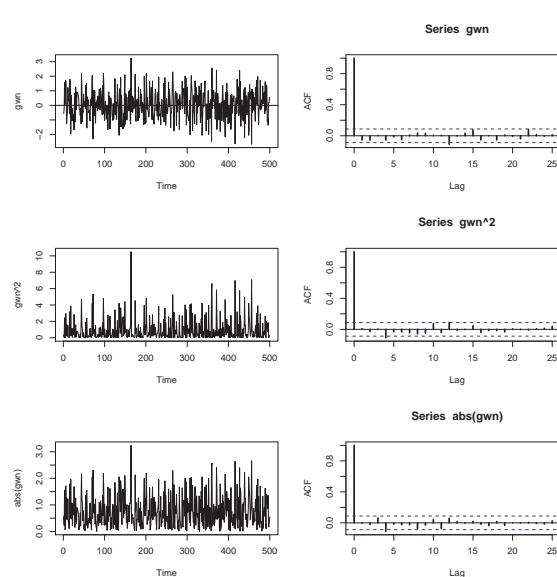


Fig. 5.20 The left panel (top to bottom) shows y_t , y_t^2 and $|y_t|$ where $y_t \sim \text{GWN}(0, 1)$ for $t = 1, \dots, 500$. The right panel shows the corresponding SACFs.

Nonlinear time dependence in asset returns

Figures 5.1.2 and 5.1.2 show that the daily and monthly returns on Microsoft and the S&P 500 index appear to exhibit periods of volatility clustering. That is, high periods of volatility tend to be followed by periods of high volatility and periods of low volatility appear to be followed by periods of low volatility. In other words, the volatility of the returns appears to exhibit some time dependence. This time dependence in volatility causes linear time dependence in the absolute value and squared value of returns.

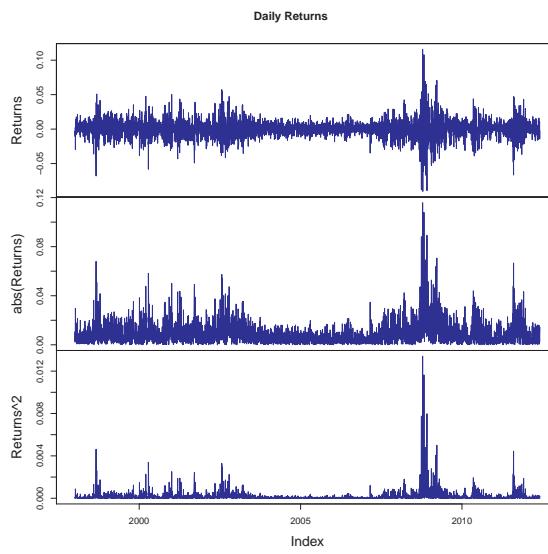


Fig. 5.21 Daily returns, absolute returns, and squared returns on the S&P 500 index.

Example 5.20. Nonlinear time dependence in the S&P 500 returns.

To illustrate nonlinear time dependence in asset returns, Figure 5.2.2 shows the daily returns, absolute returns and squared returns for the S&P 500 index. Notice how the absolute and squared returns are large (small) when the daily returns are more (less) volatile. Figure 5.2.2 shows the SACFs for the absolute and squared daily and monthly returns. There is clear evidence of time dependence in the daily absolute and squared returns while there is some evidence of time dependence in the monthly absolute and squared returns. Since the absolute and squared daily returns reflect the volatility of the daily returns, the SACFs show a strong positive linear time dependence in daily return volatility.

5.3 Bivariate Descriptive Statistics

In this section, we consider graphical and numerical descriptive statistics for summarizing two or more data series.

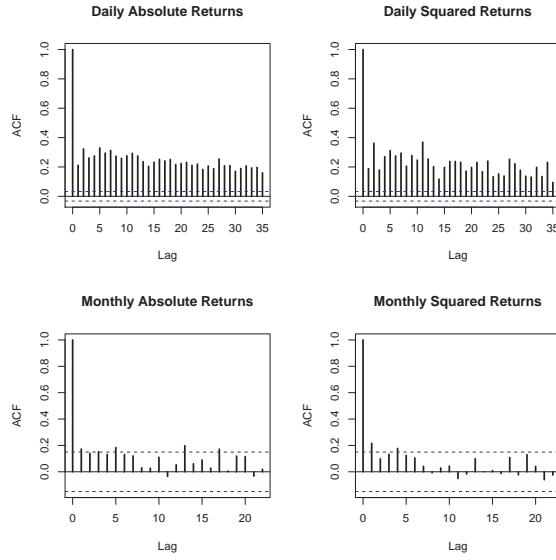


Fig. 5.22 SACFs of daily and monthly absolute and squared returns on the S&P 500 index.

5.3.1 Scatterplots

The contemporaneous dependence properties between two data series $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$ can be displayed graphically in a scatterplot, which is simply an xy-plot of the bivariate data.

Example 5.21. Scatterplot of Microsoft and S&P 500 returns.

Figure 5.3.1 shows the scatterplots between the Microsoft and S&P 500 monthly and daily returns created using:

```
par(mfrow = c(1, 2))
plot(coredata(sp500MonthlyRetS), coredata(msftMonthlyRetS), main = "Monthly returns",
      xlab = "S&P500", ylab = "MSFT", lwd = 2, pch = 16, cex = 1.25, col = "blue")
abline(v = mean(sp500MonthlyRetS))
abline(h = mean(msftMonthlyRetS))
plot(coredata(sp500DailyRetS), coredata(msftDailyRetS), main = "Daily returns",
      xlab = "S&P500", ylab = "MSFT", lwd = 2, pch = 16, cex = 1.25, col = "blue")
abline(v = mean(sp500DailyRetS))
abline(h = mean(msftDailyRetS))
```

The S&P 500 returns are put on the x-axis and the Microsoft returns on the y-axis because the “market”, as proxied by the S&P 500, is often thought as an independent variable driving individual asset returns. The upward sloping orientation of the scatterplots indicate a positive linear dependence between Microsoft and S&P 500 returns at both the monthly and daily frequencies.

Example 5.22. Pair-wise scatterplots for multiple series.

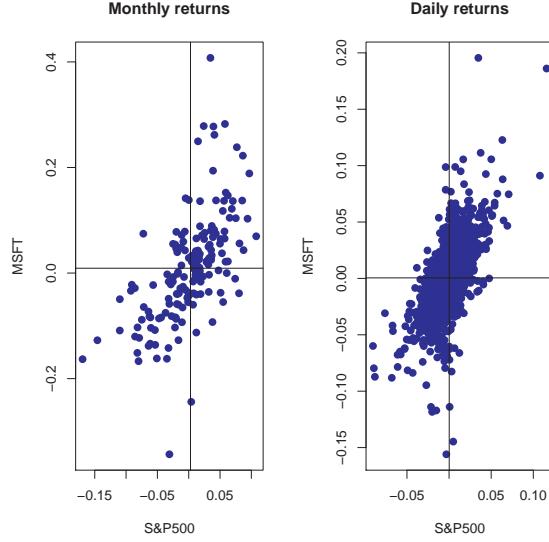


Fig. 5.23 Scatterplot of Monthly returns on Microsoft and the S&P 500 index.

For more than two data series, the R function `pairs()` plots all pair-wise scatterplots in a single plot. For example, to plot all pair-wise scatterplots for the GWN, Microsoft returns and S&P 500 returns use:

```
dataToPlot = merge(gwnMonthly, msftMonthlyRetS, sp500MonthlyRetS)
pairs(coredata(dataToPlot), col = "blue", pch = 16, cex = 1.25, cex.axis = 1.25)
```

The top row of Figure 5.3.1 shows the scatterplots between the pairs (MSFT, GWN) and (SP500, GWN), the second row shows the scatterplots between the pairs (GWN, MSFT) and (SP500, MSFT), the third row shows the scatterplots between the pairs (GWN, SP500) and (MSFT, SP500).

5.3.2 Sample covariance and correlation

For two random variables X and Y , the direction of linear dependence is captured by the covariance, $\sigma_{XY} = E[(X - \mu_X)(Y - \mu_Y)]$, and the direction and strength of linear dependence is captured by the correlation, $\rho_{XY} = \sigma_{XY}/\sigma_X\sigma_Y$. For two data series $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$, the sample covariance,

$$\hat{\sigma}_{xy} = \frac{1}{T-1} \sum_{t=1}^T (x_t - \bar{x})(y_t - \bar{y}), \quad (5.9)$$

measures the direction of linear dependence, and the sample correlation,

$$\hat{\rho}_{xy} = \frac{\hat{\sigma}_{xy}}{\hat{\sigma}_x \hat{\sigma}_y}, \quad (5.10)$$

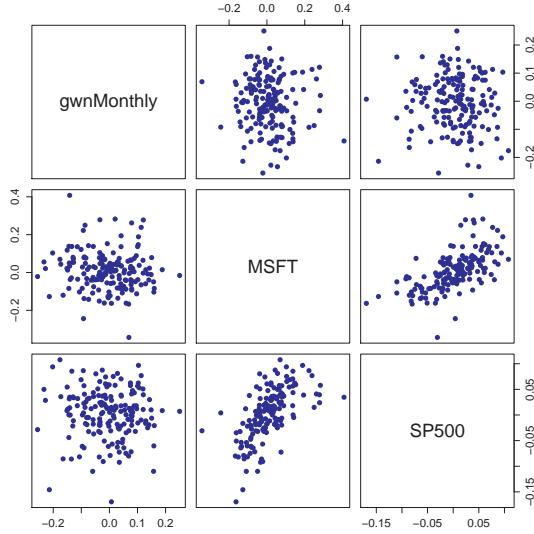


Fig. 5.24 Pair-wise scatterplots between simulated GWN, Microsoft returns and S&P 500 returns.

measures the direction and strength of linear dependence. In (5.10), $\hat{\sigma}_x$ and $\hat{\sigma}_y$ are the sample standard deviations of $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$, respectively, defined by (5.4).

When more than two data series are being analyzed, it is often convenient to compute all pair-wise covariances and correlations at once using matrix algebra. Recall, for a vector of N random variables $\mathbf{X} = (X_1, \dots, X_N)'$ with mean vector $\mu = (\mu_1, \dots, \mu_N)'$ the $N \times N$ variance-covariance matrix is defined as:

$$\Sigma = \text{var}(\mathbf{X}) = \text{cov}(\mathbf{X}) = E[(\mathbf{X} - \mu)(\mathbf{X} - \mu)'] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}.$$

For N data series $\{x_t\}_{t=1}^T$, where $x_t = (x_{1t}, \dots, x_{Nt})'$, the sample covariance matrix is computed using:

$$\hat{\Sigma} = \frac{1}{T-1} \sum_{t=1}^T (\mathbf{x}_t - \hat{\mu})(\mathbf{x}_t - \hat{\mu})' = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1N} \\ \hat{\sigma}_{12} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{1N} & \hat{\sigma}_{2N} & \cdots & \hat{\sigma}_N^2 \end{pmatrix}, \quad (5.11)$$

where $\hat{\mu}$ is the $N \times 1$ sample mean vector. Define the $N \times N$ diagonal matrix:

$$\hat{\mathbf{D}} = \begin{pmatrix} \hat{\sigma}_1 & 0 & \cdots & 0 \\ 0 & \hat{\sigma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}_N \end{pmatrix}. \quad (5.12)$$

Then the $N \times N$ sample correlation matrix $\hat{\mathbf{C}}$ is computed as:

$$\hat{\mathbf{C}} = \hat{\mathbf{D}}^{-1} \hat{\Sigma} \hat{\mathbf{D}}^{-1} = \begin{pmatrix} 1 & \hat{\rho}_{12} & \cdots & \hat{\rho}_{1N} \\ \hat{\rho}_{12} & 1 & \cdots & \hat{\rho}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\rho}_{1N} & \hat{\rho}_{2N} & \cdots & 1 \end{pmatrix}. \quad (5.13)$$

Example 5.23. Sample covariance and correlation between Microsoft and S&P 500 returns.

The scatterplots of Microsoft and S&P 500 returns in Figure 5.3.1 suggest positive linear relationships in the data. We can confirm this by computing the sample covariance and correlation using the R functions `cov()` and `cor()`. For the monthly returns, we have

```
cov(sp500MonthlyRetS, msftMonthlyRetS)

##           MSFT
## SP500 0.002977

cor(sp500MonthlyRetS, msftMonthlyRetS)

##           MSFT
## SP500 0.6138
```

Indeed, the sample covariance is positive and the sample correlation shows a moderately strong linear relationship. For the daily returns we have

```
cov(sp500DailyRetS, msftDailyRetS)

##           MSFT
## SP500 0.0001964

cor(sp500DailyRetS, msftDailyRetS)

##           MSFT
## SP500 0.671
```

Here, the daily sample covariance is about twenty times smaller than the monthly covariance (recall the square-root-of time rule), but the daily sample correlation is similar to the monthly sample correlation.

When passed a matrix of data, the `cov()` and `cor()` functions can also be used to compute the sample covariance and correlation matrices $\hat{\Sigma}$ and $\hat{\mathbf{C}}$, respectively. For example,

```
cov(msftSp500MonthlyRetS)

##          MSFT      SP500
## MSFT  0.010302 0.002977
## SP500 0.002977 0.002284

cor(msftSp500MonthlyRetS)

##          MSFT      SP500
## MSFT  1.0000 0.6138
## SP500 0.6138 1.0000
```

The function `cov2cor()` transforms a sample covariance matrix to a sample correlation matrix using (5.13):

```
cov2cor(cov(msftSp500MonthlyRetS))

##          MSFT      SP500
## MSFT  1.0000 0.6138
## SP500 0.6138 1.0000
```



Example 5.24. Visualizing correlation matrices.

The R package `corrplot` contains functions for visualizing correlation matrices. This is particularly useful for summarizing the linear dependencies among many data series. For example, Figure 5.3.2 shows the correlation plot from the `corrplot` function `corrplot.mixed()` created using:

```
dataToPlot = merge(gwnMonthly, msftMonthlyRetS, sp500MonthlyRetS)
cor.mat = cor(dataToPlot)
corrplot.mixed(cor.mat, lower = "number", upper = "ellipse")
```

The color scheme shows the magnitudes of the correlations (blue for positive and red for negative) and the orientation of the ellipses show the magnitude and direction of the linear associations.



5.3.3 Sample cross-lag covariances and correlations

The dynamic interactions between two observed time series $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$ can be measured using the sample cross-lag covariances and correlations

$$\begin{aligned}\hat{\gamma}_{xy}^k &= \widehat{\text{cov}}(X_t, Y_{t-k}), \\ \hat{\rho}_{xy}^k &= \widehat{\text{corr}}(X_t, Y_{t-k}) = \frac{\hat{\gamma}_{xy}^k}{\sqrt{\hat{\sigma}_x^2 \hat{\sigma}_y^2}}.\end{aligned}$$



Fig. 5.25 Correlation plot created with `corrplot()`.

When more than two data series are being analyzed, all pairwise cross-lag sample covariances and sample correlations can be computed at once using matrix algebra. For a time series of N data series $\{\mathbf{x}_t\}_{t=1}^T$, where $\mathbf{x}_t = (x_{1t}, \dots, x_{Nt})'$, the sample lag k cross-lag covariance and correlation matrices are computed using:

$$\hat{\mathbf{\Gamma}}_k = \frac{1}{T-1} \sum_{t=k+1}^T (\mathbf{x}_t - \hat{\mu})(\mathbf{x}_{t-k} - \hat{\mu})', \quad (5.14)$$

$$\hat{\mathbf{C}}_k = \hat{\mathbf{D}}^{-1} \hat{\mathbf{\Gamma}}_k \hat{\mathbf{D}}^{-1}, \quad (5.15)$$

where $\hat{\mathbf{D}}$ is defined in (5.12).

Example 5.25. Sample cross-lag covariances and correlations between Microsoft and S&P 500 returns.

Consider computing the cross-lag covariance and correlation matrices (5.14) and (5.15) for $k = 0, 1, \dots, 5$ between Microsoft and S&P 500 monthly returns. These matrices may be computed using the R function `acf()` as follows:

```
Ghat = acf(coredata(msftSp500MonthlyRetS), type = "covariance", lag.max = 5, plot = FALSE)
Chat = acf(coredata(msftSp500MonthlyRetS), type = "correlation", lag.max = 5, plot = FALSE)
class(Ghat)

## [1] "acf"

names(Ghat)

## [1] "acf"      "type"     "n.used"   "lag"      "series"   "snames"
```

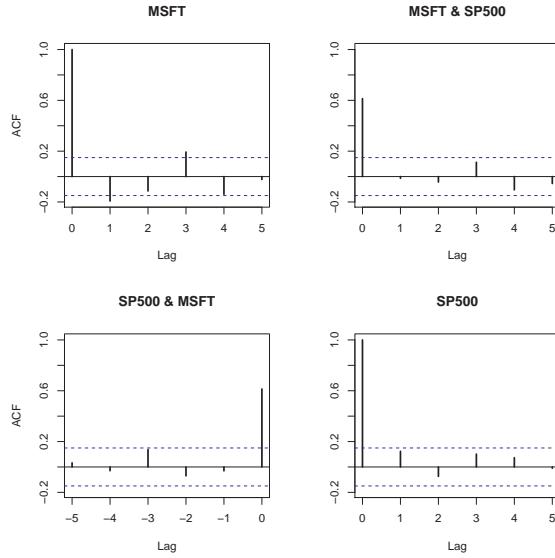


Fig. 5.26 Sample cross-lag correlations between Microsoft and S&P 500 returns.

Here, `Ghat` and `Chat` are objects of class “`acf`” for which there are print and plot methods. The `acf` components of `Ghat` and `Chat` are 3-dimensional arrays containing the cross lag matrices (5.14) and (5.15), respectively. For example, to extract $\hat{\mathbf{C}}_0$ and $\hat{\mathbf{C}}_1$ use:

```
# Chat0
Chat$acf[1, , ]
##      [,1]     [,2]
## [1,] 1.0000 0.6138
## [2,] 0.6138 1.0000

# Chat1
Chat$acf[2, , ]
##      [,1]     [,2]
## [1,] -0.19310 -0.01351
## [2,] -0.03069  0.12317
```

The print method shows the sample autocorrelations of each variable as well as the pairwise cross-lag correlations:

```
Chat

##
## Autocorrelations of series 'coredata(msftSp500MonthlyRetS)', by lag
##
## , , MSFT
##
##   MSFT      SP500
##   1.000 ( 0) 0.614 ( 0)
##  -0.193 ( 1) -0.031 (-1)
```

```

## -0.114 ( 2) -0.070 (-2)
##  0.193 ( 3)  0.137 (-3)
## -0.139 ( 4) -0.030 (-4)
## -0.023 ( 5)  0.032 (-5)
##
## , , SP500
##
## MSFT      SP500
##  0.614 ( 0)  1.000 ( 0)
## -0.014 ( 1)  0.123 ( 1)
## -0.043 ( 2) -0.074 ( 2)
##  0.112 ( 3)  0.101 ( 3)
## -0.105 ( 4)  0.073 ( 4)
## -0.055 ( 5) -0.010 ( 5)

```

These values can also be visualized using the plot method:

```
plot(Chat, lwd = 2)
```

Figure 5.26 shows the resulting four-panel plot. The top-left and bottom-right (diagonal) panels give the SACFs for Microsoft and S&P 500 returns. The top-right plot gives the cross-lag correlations $\hat{\rho}_{msft,sp500}^k$ for $k = 0, 1, \dots, 5$ and the bottom-left panel gives the corss-lag correlations $\hat{\rho}_{sp500,msft}^k$ for $k = 0, 1, \dots, 5$. The plots show no evidence of any dynamic feedback between the return series.

■

5.4 Stylized facts for daily and monthly asset returns

A stylized fact is something that is generally true but not always. From the data analysis of the daily and monthly returns on Microsoft and the S&P 500 index we observe a number of stylized facts that tend to be true for other individual assets and portfolios of assets. For monthly data we observe the following stylized facts:

1. M1. Prices appear to be random walk non-stationary and returns appear to be mostly covariance stationary. There is evidence that return volatility changes over time.
2. M2. Returns appear to approximately normally distributed. There is some negative skewness and excess kurtosis.
3. M3. Assets that have high average returns tend to have high standard deviations (volatilities) and vice-versa. This is the no free lunch principle.
4. M4. Returns on individual assets (stocks) have higher standard deviations than returns on diversified portfolios of assets (stocks).
5. M5. Returns on different assets tend to be positively correlated. It is unusual for returns on different assets to be negatively correlated.
6. M6. Returns are approximately uncorrelated over time. That is, there is little evidence of linear time dependence in asset returns.

7. M7. Returns do not exhibit dynamic feedback. That is, there are no lead-lag effects between pairs of returns.

Daily returns have some features in common with monthly returns and some not. The common features are M1 and M3-M7 above. The stylized facts that are specific to daily returns are:

1. D2. Returns are not normally distributed. Empirical distributions have much fatter tails than the normal distribution (excess kurtosis).
2. D7. Returns are not independent over time. Absolute and squared returns are positively auto correlated and the correlation dies out very slowly. Volatility appears to be auto correlated and, hence, predictable.

These stylized facts of daily and monthly asset returns are the main features of returns that models of assets returns should capture. A good model is one that can explain many stylized facts of the data. A bad model does not capture important stylized facts.

5.5 Further Reading

Fama (1976) is the first book that provides a comprehensive descriptive statistical analysis of asset returns. Much of the material in this chapter, and this book, is motivated by Fama's book. Several recent textbooks discuss exploratory data analysis of finance time series. This discussion in this chapter is similar to that presented in Carmona (2010), Jondeau et al. (2007), Ruppert and Matteson (2016), Tsay (2010), Tsay (2014), and Zivot (2016).

5.6 Problems

Exercise 5.1. In this question, you will compute descriptive statistics for the continuously compounded monthly return data on the Vanguard long term bond index fund (VBLTX), Fidelity Magellan stock mutual fund (FMAGX), and Starbucks stock (SBUX). You will use the R packages **corrplot**, **IntroCompFinR**, **PerformanceAnalytics**, **tseries** and **zoo**.

1. Downloading data and return calculations.
 - a. Use the **tseries** function **get.hist.quote()** to download from Yahoo! the monthly adjusted closing prices on VBLTX, FMAGX and SBUX over the period 1998-01-01 through 2009-12-31. Download these prices as “**zoo**” objects.
 - b. Use the **PerformanceAnalytics** function **Return.calculate()** to compute continuously compounded monthly returns and remove the first NA observation.
2. Univariate graphical analysis.
 - a. Make time plots of the return data using the **zoo** function **plot.zoo()**. Comment on any relationships between the returns suggested by the plots. Pay particular attention

- to the behavior of returns toward the end of 2008 at the beginning of the financial crisis.
- b. Make a equity curve/cumulative return plot (future of \$1 invested in each asset) and comment. Which assets gave the best and worst future values over the investment horizon?
 - c. For each return series, use the **IntroCompFinR** function `fourPanelPlot()` to make a four-panel plot containing a histogram, boxplot, normal QQ-plot, and SACF plot. Do the return series look normally distributed? Briefly compare the return distributions. Do you see any evidence of time dependence in the returns?
3. Univariate numerical summary statistics and historical VaR.
- a. Compute numerical descriptive statistics for all assets using the R functions `summary()`, `mean()`, `var()`, `stdev()`, and the **PerformanceAnalytics** functions `skewness()` and `kurtosis()`. Compare and contrast the descriptive statistics for the three assets. Which asset appears to be the riskiest asset?
 - b. Using the mean monthly return for each asset, compute an estimate of the annual continuously compounded return (i.e., recall the relationship between the expected monthly cc return and the expected annual cc return). Convert this annual continuously compounded return into a simple annual return. Are there any surprises?
 - c. Using the estimate of the monthly return standard deviation for each asset, compute an estimate of the annual return standard deviation. Briefly comment on the magnitude of the annual standard deviations.
 - d. For each asset compute the empirical 1% and 5% quantiles of the cc returns. Using these quantiles compute the 1% and 5% historical (monthly) VaR values based on an initial \$100,000 investment. Which asset has the highest and lowest VaR values? Are you surprised?
4. Bivariate graphical analysis and numerical summary statistics
- a. Use the R `pairs()` function to create all pair-wise scatterplots of returns. Comment on the direction and strength of the linear relationships in these plots.
 - b. Use the **corrplot** functions `corrplot()` and `corrplot.mixed()` to plot the correlation matrix of the returns on the three assets.
 - c. Use the R functions `var()`, `cov()`, and `cor()` to compute the sample covariance matrix and sample correlation matrix of the returns. Comment on the direction and strength of the linear relationships suggested by the values of the covariances and correlations.
- Exercise 5.2.** In this question, you will compute descriptive statistics for the continuously compounded weekly return data on the Vanguard long term bond index fund (VBLTX), Fidelity Magellan stock mutual fund (FMAGX), and Starbucks stock (SBUX). Repeat the analysis for the monthly returns but use instead weekly returns computed from end-of-week adjusted closing prices.
- Exercise 5.3.** In this question, you will compute descriptive statistics for the continuously compounded daily return data on the Vanguard long term bond index fund (VBLTX),

Fidelity Magellan stock mutual fund (FMAGX), and Starbucks stock (SBUX). Repeat the analysis for the monthly returns but use instead daily returns computed from end-of-day adjusted closing prices.

Exercise 5.4. In this problem you will analyze the monthly simple returns on the Vanguard S&P 500 index (VFINX) and Amazon stock (AMZN) over the 5-year period January 2010, through January 2015. For this period there are $T=60$ monthly returns. The monthly prices for VFINX are in the **IntroCompFinR** object `VanguardPrices` and the daily prices for AMZN are in the **IntroCompFinR** object `amznDailyPrices`.

1. Compute simple monthly returns on AMZN and VFINX and plot these returns together in a single plot. Do the monthly returns from the two assets look like realizations from a covariance stationary stochastic process? Why or why not?
2. Compare and contrast the return characteristics of the two assets. In addition, comment on any common features, if any, of the two return series.
3. Plot the cumulative simple returns (equity curve) for each fund, which represents the growth of \$1 invested in each fund over the sample period, and plot these returns in the same graph. Which fund performed better over the sample?
4. Using the **IntroCompFinR** function `fourPanelPlot()`, create descriptive statistics plots for each asset. In addition, compute Based on statistics, do the returns on VFINX and AMZN look normally distributed? Is there any evidence of linear time dependence? Briefly justify your answer.
5. Which asset appears to be riskier? Briefly justify your answer.

References

1. Carmona, R. (2014). *Statistical Analysis of Financial Data in R, Second Edition*. Springer, New York.
2. Fama, E.F. (1976). *Foundations of Finance*. Basic Books, New York.
3. Jondeau, R., Poon, S.-H., and Rockinger, M. (2007). *Financial Modeling Under Non-Gaussian Distributions*. Springer, New York.
4. Ruppert, D., and Matteson, D.S. (2015). *Statistics and Data Analysis for Financial Engineering with R examples*. Springer, New York.
5. Tsay, R. (2010). *Analysis of Financial Time Series*, Wiley.
6. Tsay, R. (2014). *An Introduction to Analysis of Financial Time Series*, Wiley.
7. Zivot, E. (2016). *Modeling Financial Time Series with R*. Springer, New York.

Chapter 6

The Constant Expected Return Model

Updated: July 22, 2016

Copyright © Eric Zivot 2015, 2016

The first model of asset returns we consider is the very simple *constant expected return* (CER) model. This model is motivated by the stylized facts for monthly asset returns. The CER model assumes that an asset's return (simple or continuously compounded) over time is independent and identically normally distributed with a constant (time invariant) mean and variance. The model allows for the returns on different assets to be contemporaneously correlated but that the correlations are constant over time. The CER model is widely used in finance. For example, it is used in risk analysis (e.g., computing Value-at-Risk) for assets and portfolios, in mean-variance portfolio analysis, in the Capital Asset Pricing Model (CAPM), and in the Black-Scholes option pricing model. Although this model is very simple, it provides important intuition about the statistical behavior of asset returns and prices and serves as a benchmark against which more complicated models can be compared and evaluated. It allows us to discuss and develop several important econometric topics such as Monte Carlo simulation, estimation, bootstrapping, hypothesis testing, forecasting and model evaluation that are discussed in later chapters.

The outline of this chapter is as follows. Section 6.1 reviews the assumptions of the CER model and presents several equivalent specifications of the model. Application of the model to continuously compounded and simple returns is also discussed. Section 6.2 illustrates Monte Carlo simulation of the CER model.

The R packages used in this chapter are **IntroCompFinR**, **mvtnorm**, and **PerformanceAnalytics**. Make sure these packages are installed and loaded before running the R examples in the chapter.

6.1 CER Model Assumptions

Let R_{it} denote the simple or continuously compounded return on asset i over the investment horizon between times $t - 1$ and t (e.g., monthly returns). We make the following assumption:

tions regarding the probability distribution of R_{it} for $i = 1, \dots, N$ assets for all times t .

Assumption 1

- (i) *Covariance stationarity and ergodicity:* $\{R_{i1}, \dots, R_{iT}\} = \{R_{it}\}_{t=1}^T$ is a covariance stationary and ergodic stochastic process with $E[R_{it}] = \mu_i$, $\text{var}(R_{it}) = \sigma_i^2$, $\text{cov}(R_{it}, R_{jt}) = \sigma_{ij}$ and $\text{cor}(R_{it}, R_{jt}) = \rho_{ij}$.
- (ii) *Normality:* $R_{it} \sim N(\mu_i, \sigma_i^2)$ for all i and t , and all joint distributions are normal.
- (iii) *No serial correlation:* $\text{cov}(R_{it}, R_{js}) = \text{cor}(R_{it}, R_{js}) = 0$ for $t \neq s$ and $i, j = 1, \dots, N$.

Assumptions 1 states that in every time period asset returns are jointly (multivariate) normally distributed, that the means and the variances of all asset returns, and all of the pairwise contemporaneous covariances and correlations between assets are constant over time. In addition, all of the asset returns are *serially uncorrelated*:

$$\text{cor}(R_{it}, R_{is}) = \text{cov}(R_{it}, R_{is}) = 0 \text{ for all } i \text{ and } t \neq s,$$

and the returns on all possible pairs of assets i and j are serially uncorrelated,

$$\text{cor}(R_{it}, R_{js}) = \text{cov}(R_{it}, R_{js}) = 0 \text{ for all } i \neq j \text{ and } t \neq s.$$

In addition, under the normal distribution assumption lack of serial correlation implies time independence of returns over time. Clearly, these are very strong assumptions. However, they allow us to develop a straightforward probabilistic model for asset returns as well as statistical tools for estimating the parameters of the model, testing hypotheses about the parameter values and assumptions.

6.1.1 Regression model representation

A convenient mathematical representation or *model* of asset returns can be given based on Assumption 1. This is the CER *regression model*. For assets $i = 1, \dots, N$ and time periods $t = 1, \dots, T$, the CER regression model is:

$$\begin{aligned} R_{it} &= \mu_i + \varepsilon_{it}, \\ \{\varepsilon_{it}\}_{t=1}^T &\sim \text{GWN}(0, \sigma_i^2), \\ \text{cov}(\varepsilon_{it}, \varepsilon_{js}) &= \begin{cases} \sigma_{ij} & t = s \\ 0 & t \neq s \end{cases}. \end{aligned} \tag{6.1}$$

The notation $\varepsilon_{it} \sim \text{GWN}(0, \sigma_i^2)$ stipulates that the stochastic process $\{\varepsilon_{it}\}_{t=1}^T$ is a Gaussian white noise process with $E[\varepsilon_{it}] = 0$ and $\text{var}(\varepsilon_{it}) = \sigma_i^2$. In addition, the random error term ε_{it} is independent of ε_{js} for all assets $i \neq j$ and all time periods $t \neq s$.

Using the basic properties of expectation, variance and covariance, we can derive the following properties of returns in the CER model:

$$\begin{aligned}
E[R_{it}] &= E[\mu_i + \varepsilon_{it}] = \mu_i + E[\varepsilon_{it}] = \mu_i, \\
\text{var}(R_{it}) &= \text{var}(\mu_i + \varepsilon_{it}) = \text{var}(\varepsilon_{it}) = \sigma_i^2, \\
\text{cov}(R_{it}, R_{jt}) &= \text{cov}(\mu_i + \varepsilon_{it}, \mu_j + \varepsilon_{jt}) = \text{cov}(\varepsilon_{it}, \varepsilon_{jt}) = \sigma_{ij}, \\
\text{cov}(R_{it}, R_{js}) &= \text{cov}(\mu_i + \varepsilon_{it}, \mu_j + \varepsilon_{js}) = \text{cov}(\varepsilon_{it}, \varepsilon_{js}) = 0, \quad t \neq s.
\end{aligned}$$

Given that covariances and variances of returns are constant over time implies that the correlations between returns over time are also constant:

$$\begin{aligned}
\text{cor}(R_{it}, R_{jt}) &= \frac{\text{cov}(R_{it}, R_{jt})}{\sqrt{\text{var}(R_{it})\text{var}(R_{jt})}} = \frac{\sigma_{ij}}{\sigma_i\sigma_j} = \rho_{ij}, \\
\text{cor}(R_{it}, R_{js}) &= \frac{\text{cov}(R_{it}, R_{js})}{\sqrt{\text{var}(R_{it})\text{var}(R_{js})}} = \frac{0}{\sigma_i\sigma_j} = 0, \quad i \neq j, t \neq s.
\end{aligned}$$

Finally, since $\{\varepsilon_{it}\}_{t=1}^T \sim \text{GWN}(0, \sigma_i^2)$ it follows that $\{R_{it}\}_{t=1}^T \sim \text{iid } N(\mu_i, \sigma_i^2)$. Hence, the CER regression model (6.1) for R_{it} is equivalent to the model implied by Assumption 1.

Interpretation of the CER regression model

The CER model has a very simple form and is identical to the *measurement error model* in the statistics literature.¹ In words, the model states that each asset return is equal to a constant μ_i (the expected return) plus a normally distributed random variable ε_{it} with mean zero and constant variance. The random variable ε_{it} can be interpreted as representing the *unexpected news* concerning the value of the asset that arrives between time $t - 1$ and time t . To see this, (6.1) implies that:

$$\varepsilon_{it} = R_{it} - \mu_i = R_{it} - E[R_{it}],$$

so that ε_{it} is defined as the deviation of the random return from its expected value. If the news between times $t - 1$ and t is good, then the realized value of ε_{it} is positive and the observed return is above its expected value μ_i . If the news is bad, then ε_{it} is negative and the observed return is less than expected. The assumption $E[\varepsilon_{it}] = 0$ means that news, on average, is neutral; neither good nor bad. The assumption that $\text{var}(\varepsilon_{it}) = \sigma_i^2$ can be interpreted as saying that volatility, or typical magnitude, of news arrival is constant over time. The random news variable affecting asset i , ε_{it} , is allowed to be contemporaneously correlated with the random news variable affecting asset j , ε_{jt} , to capture the idea that news about one asset may spill over and affect another asset. For example, if asset i is Microsoft stock and asset j is Apple Computer stock, then one interpretation of news in this context is general news about the computer industry and technology. Good news should lead to positive values of both ε_{it} and ε_{jt} . Hence these variables will be positively correlated due to a positive reaction to a common news component. Finally, the news on asset j at time s is unrelated to the news on asset i at time t for all times $t \neq s$. For example, this means that the news for Apple in January is not related to the news for Microsoft in February.

¹ In the measurement error model, r_{it} represents the t^{th} measurement of some physical quantity μ_i and ε_{it} represents the random measurement error associated with the measurement device. The value σ_i represents the typical size of a measurement error.

6.1.2 Location-Scale model representation

Sometimes it is convenient to re-express the regression form of the CER model (6.1) in *location-scale* form:

$$\begin{aligned} R_{it} &= \mu_i + \varepsilon_{it} = \mu_i + \sigma_i \cdot Z_{it} \\ \{Z_{it}\}_{t=1}^T &\sim \text{GWN}(0, 1), \end{aligned} \quad (6.2)$$

where we use the decomposition $\varepsilon_{it} = \sigma_i \cdot Z_{it}$. In this form, the random news shock is the *iid* standard normal random variable Z_{it} scaled by the “news” volatility σ_i . This form is particularly convenient for Value-at-Risk calculations because the $\alpha \times 100\%$ quantile of the return distribution has the simple form:

$$q_\alpha^{R_i} = \mu_i + \sigma_i \times q_\alpha^Z,$$

where q_α^Z is the $\alpha \times 100\%$ quantile of the standard normal random distribution. Let W_0 be the initial amount of wealth to be invested from time $t - 1$ to t . If R_{it} is the simple return then,

$$\text{VaR}_\alpha = W_0 \times q_\alpha^{R_i},$$

whereas if R_{it} is the continuously compounded return then,

$$\text{VaR}_\alpha = W_0 \times \left(e^{q_\alpha^{R_i}} - 1 \right).$$

6.1.3 The CER Model in matrix notation

Define the $N \times 1$ vectors $R_t = (R_{1t}, \dots, R_{Nt})'$, $\mu = (\mu_1, \dots, \mu_N)'$, $\varepsilon_t = (\varepsilon_{1t}, \dots, \varepsilon_{Nt})'$ and the $N \times N$ symmetric covariance matrix:

$$\text{var}(\varepsilon_t) = \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}.$$

Then the regression form of the CER model in matrix notation is:

$$\begin{aligned} \mathbf{R}_t &= \mu + \varepsilon_t, \\ \varepsilon_t &\sim \text{iid } N(\mathbf{0}, \boldsymbol{\Sigma}), \end{aligned} \quad (6.3)$$

which implies that $\mathbf{R}_t \sim \text{iid } N(\mu, \boldsymbol{\Sigma})$.

The location-scale form of the CER model in matrix notation makes use of the matrix square root factorization $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{1/2} \boldsymbol{\Sigma}^{1/2'}$ where $\boldsymbol{\Sigma}^{1/2}$ is the lower-triangular matrix square

root (usually the Cholesky factorization). Then (6.3) can be rewritten as:

$$\begin{aligned}\mathbf{R}_t &= \mu + \boldsymbol{\Sigma}^{1/2} \mathbf{Z}_t, \\ \mathbf{Z}_t &\sim \text{iid } N(\mathbf{0}, \mathbf{I}_N),\end{aligned}\tag{6.4}$$

where \mathbf{I}_N denotes the N -dimensional identity matrix.

6.1.4 The CER model for continuously compounded returns

The CER model is often used to describe continuously compounded returns defined as $R_{it} = \ln(P_{it}/P_{it-1})$ where P_{it} is the price of asset i at time t . This is particularly convenient for investment risk analysis. An advantage of the CER model for continuously compounded returns is that the model aggregates to any time horizon because multi-period continuously compounded returns are additive. The CER model for continuously compounded returns also gives rise to the random walk model for the logarithm of asset prices. The normal distribution assumption of the CER model for continuously compounded returns implies that single-period simple returns are log-normally distributed.

A disadvantage of the CER model for continuously compounded returns is that the model has some limitations for the analysis of portfolios because the continuously compounded return on a portfolio of assets is not a weighted average of the continuously compounded returns on the individual securities. As a result, for portfolio analysis the CER model is typically applied to simple returns.

Time Aggregation and the CER Model

The CER model for continuously compounded returns has the following nice aggregation property with respect to the interpretation of ε_{it} as news. For illustration purposes, suppose that t represents months so that R_{it} is the continuously compounded monthly return on asset i . Now, instead of the monthly return, suppose we are interested in the annual continuously compounded return $R_{it} = R_{it}(12)$. Since multi-period continuously compounded returns are additive, $R_{it}(12)$ is the sum of 12 monthly continuously compounded returns:

$$R_{it} = R_{it}(12) = \sum_{k=0}^{11} R_{it-k} = R_{it} + R_{it-1} + \cdots + R_{it-11}.$$

Using the CER regression model (6.1) for the monthly return R_{it} , we may express the annual return $R_{it}(12)$ as:

$$R_{it}(12) = \sum_{t=0}^{11} (\mu_i + \varepsilon_{it}) = 12 \cdot \mu_i + \sum_{t=0}^{11} \varepsilon_{it} = \mu_i(12) + \varepsilon_{it}(12),$$

where $\mu_i(12) = 12 \cdot \mu_i$ is the annual expected return on asset i , and $\varepsilon_{it}(12) = \sum_{k=0}^{11} \varepsilon_{it-k}$ is the annual random news component. The annual expected return, $\mu_i(12)$, is simply 12 times the monthly expected return, μ_i . The annual random news component, $\varepsilon_{it}(12)$, is the accumulation of news over the year. As a result, the variance of the annual news component, $(\sigma_i(12))^2$, is 12 times the variance of the monthly news component:

$$\begin{aligned}\text{var}(\varepsilon_{it}(12)) &= \text{var}\left(\sum_{k=0}^{11} \varepsilon_{it-k}\right) \\ &= \sum_{k=0}^{11} \text{var}(\varepsilon_{it-k}) \text{ since } \varepsilon_{it} \text{ is uncorrelated over time} \\ &= \sum_{k=0}^{11} \sigma_i^2 \text{ since } \text{var}(\varepsilon_{it}) \text{ is constant over time} \\ &= 12 \cdot \sigma_i^2 = \sigma_i^2(12).\end{aligned}$$

It follows that the standard deviation of the annual news is equal to $\sqrt{12}$ times the standard deviation of monthly news:

$$\text{SD}(\varepsilon_{it}(12)) = \sqrt{12} \times \text{SD}(\varepsilon_{it}) = \sqrt{12} \times \sigma_i.$$

Similarly, due to the additivity of covariances, the covariance between $\varepsilon_{it}(12)$ and $\varepsilon_{jt}(12)$ is 12 times the monthly covariance:

$$\begin{aligned}\text{cov}(\varepsilon_{it}(12), \varepsilon_{jt}(12)) &= \text{cov}\left(\sum_{k=0}^{11} \varepsilon_{it-k}, \sum_{k=0}^{11} \varepsilon_{jt-k}\right) \\ &= \sum_{k=0}^{11} \text{cov}(\varepsilon_{it-k}, \varepsilon_{jt-k}) \text{ since } \varepsilon_{it} \text{ and } \varepsilon_{jt} \text{ are uncorrelated over time} \\ &= \sum_{k=0}^{11} \sigma_{ij} \text{ since } \text{cov}(\varepsilon_{it}, \varepsilon_{jt}) \text{ is constant over time} \\ &= 12 \cdot \sigma_{ij} = \sigma_{ij}(12).\end{aligned}$$

The above results imply that the correlation between the annual errors $\varepsilon_{it}(12)$ and $\varepsilon_{jt}(12)$ is the same as the correlation between the monthly errors ε_{it} and ε_{jt} :

$$\begin{aligned}\text{cor}(\varepsilon_{it}(12), \varepsilon_{jt}(12)) &= \frac{\text{cov}(\varepsilon_{it}(12), \varepsilon_{jt}(12))}{\sqrt{\text{var}(\varepsilon_{it}(12)) \cdot \text{var}(\varepsilon_{jt}(12))}} \\ &= \frac{12 \cdot \sigma_{ij}}{\sqrt{12\sigma_i^2 \cdot 12\sigma_j^2}} \\ &= \frac{\sigma_{ij}}{\sigma_i \sigma_j} = \rho_{ij} = \text{cor}(\varepsilon_{it}, \varepsilon_{jt}).\end{aligned}$$

The above results generalize to aggregating returns to arbitrary time horizons. Let R_{it} denote the continuously compounded return between times $t-1$ and t , where t represents the general investment horizon, and let $R_{it}(k) = \sum_{j=0}^{k-1} R_{it-j}$ denote the k -period continuously

compounded return. Then the CER model for $R_{it}(k)$ has the form:

$$\begin{aligned} R_{it}(k) &= \mu_i(k) + \varepsilon_{it}(k), \\ \varepsilon_{it}(k) &\sim N(0, \sigma_i^2(k)), \end{aligned}$$

where $\mu_i(k) = k \times \mu_i$ is the k -period expected return, $\varepsilon_{it}(k) = \sum_{j=0}^{k-1} \varepsilon_{it-j}$ is the k -period error term, and $\sigma_i^2(k) = k \times \sigma_i^2$ is the k -period variance. The k -period volatility follows the *square-root-of-time rule*: $\sigma_i(k) = \sqrt{k} \times \sigma_i$. This aggregation result is exact for continuously compounded returns but it is often used as an approximation for simple returns.

The random walk model of asset prices

The CER model for continuously compounded returns (6.1) gives rise to the so-called *random walk* (RW) model for the *logarithm* of asset prices. To see this, recall that the continuously compounded return, R_{it} , is defined from asset prices via $R_{it} = \ln\left(\frac{P_{it}}{P_{it-1}}\right) = \ln(P_{it}) - \ln(P_{it-1})$. Letting $p_{it} = \ln(P_{it})$ and using the representation of R_{it} in the CER model (6.1), we can express the log-price as:

$$p_{it} = p_{it-1} + \mu_i + \varepsilon_{it}. \quad (6.5)$$

The representation in (6.5) is known as the RW model for log-prices.² It is a representation of the CER model in terms of log-prices.

In the RW model (6.5), μ_i represents the expected change in the log-price (continuously compounded return) between months $t-1$ and t , and ε_{it} represents the unexpected change in the log-price. That is,

$$\begin{aligned} E[\Delta p_{it}] &= E[R_{it}] = \mu_i, \\ \varepsilon_{it} &= \Delta p_{it} - E[\Delta p_{it}]. \end{aligned}$$

where $\Delta p_{it} = p_{it} - p_{it-1}$. Further, in the RW model, the unexpected changes in log-price, ε_{it} , are uncorrelated over time ($\text{cov}(\varepsilon_{it}, \varepsilon_{is}) = 0$ for $t \neq s$) so that future changes in log-price cannot be predicted from past changes in the log-price.³

The RW model gives the following interpretation for the evolution of log prices. Let p_{i0} denote the initial log price of asset i . The RW model says that the log-price at time $t=1$ is:

$$p_{i1} = p_{i0} + \mu_i + \varepsilon_{i1},$$

where ε_{i1} is the value of random news that arrives between times 0 and 1. At time $t=0$ the expected log-price at time $t=1$ is,

$$E[p_{i1}] = p_{i0} + \mu_i + E[\varepsilon_{i1}] = p_{i0} + \mu_i,$$

² The model (6.5) is technically a random walk with drift μ_i . A pure random walk has zero drift ($\mu_i = 0$)

³ The notion that future changes in asset prices cannot be predicted from past changes in asset prices is often referred to as the weak form of the efficient markets hypothesis.

which is the initial price plus the expected return between times 0 and 1. Similarly, by recursive substitution the log-price at time $t = 2$ is,

$$\begin{aligned} p_{i2} &= p_{i1} + \mu_i + \varepsilon_{i2} \\ &= p_{i0} + \mu_i + \mu_i + \varepsilon_{i1} + \varepsilon_{i2} \\ &= p_{i0} + 2 \cdot \mu_i + \sum_{t=1}^2 \varepsilon_{it}, \end{aligned}$$

which is equal to the initial log-price, p_{i0} , plus the two period expected return, $2 \cdot \mu_i$, plus the accumulated random news over the two periods, $\sum_{t=1}^2 \varepsilon_{it}$. By repeated recursive substitution, the log price at time $t = T$ is,

$$p_{iT} = p_{i0} + T \cdot \mu_i + \sum_{t=1}^T \varepsilon_{it}.$$

At time $t = 0$, the expected log-price at time $t = T$ is,

$$E[p_{iT}] = p_{i0} + T \cdot \mu_i,$$

which is the initial price plus the expected growth in prices over T periods. The actual price, p_{iT} , deviates from the expected price by the accumulated random news:

$$p_{iT} - E[p_{iT}] = \sum_{t=1}^T \varepsilon_{it}.$$

At time $t = 0$, the variance of the log-price at time T is,

$$\text{var}(p_{iT}) = \text{var}\left(\sum_{t=1}^T \varepsilon_{it}\right) = T \cdot \sigma_i^2$$

Hence, the RW model implies that the stochastic process of log-prices $\{p_{it}\}$ is non-stationary because the variance of p_{it} increases with t . Finally, because $\varepsilon_{it} \sim iidN(0, \sigma_i^2)$ it follows that (conditional on p_{i0}) $p_{iT} \sim N(p_{i0} + T\mu_i, T\sigma_i^2)$.

The term *random walk* was originally used to describe the unpredictable movements of a drunken sailor staggering down the street. The sailor starts at an initial position, p_0 , outside the bar. The sailor generally moves in the direction described by μ but randomly deviates from this direction after each step t by an amount equal to ε_t . After T steps the sailor ends up at position $p_T = p_0 + \mu \cdot T + \sum_{t=1}^T \varepsilon_t$. The sailor is expected to be at location μT , but where he actually ends up depends on the accumulation of the random changes in direction $\sum_{t=1}^T \varepsilon_t$. Because $\text{var}(p_T) = \sigma^2 T$, the uncertainty about where the sailor will be increases with each step.

The RW model for log-prices implies the following model for prices:

$$P_{it} = e^{p_{it}} = P_{i0} e^{\mu_i \cdot t + \sum_{s=1}^t \varepsilon_{is}} = P_{i0} e^{\mu_i t} e^{\sum_{s=1}^t \varepsilon_{is}},$$

where $p_{it} = p_{i0} + \mu_i t + \sum_{s=1}^t \varepsilon_s$. The term $e^{\mu_i t}$ represents the expected exponential growth rate in prices between times 0 and time t , and the term $e^{\sum_{s=1}^t \varepsilon_{is}}$ represents the unexpected exponential growth in prices. Here, conditional on P_{i0} , P_{it} is log-normally distributed because $p_{it} = \ln P_{it}$ is normally distributed.

6.1.5 CER model for simple returns

For simple returns, defined as $R_{it} = (P_{it} - P_{it-1})/P_{it-1}$, the CER model is often used for the analysis of portfolios as discussed in chapters 11 - 15. The reason is that the simple return on a portfolio of N assets is weighted average of the simple returns on the individual assets. Hence, the CER model for simple returns extends naturally to portfolios of assets.

CER Model and Portfolios

Consider the CER model in matrix form (6.3) for the $N \times 1$ vector of simple returns $\mathbf{R}_t = (R_{1t}, \dots, R_{Nt})'$. For a vector of portfolio weights $\mathbf{x} = (x_1, \dots, x_N)$ such that $\mathbf{x}'\mathbf{1} = \sum_{i=1}^N x_i = 1$, the simple return on the portfolio is:

$$R_{pt} = \mathbf{x}'\mathbf{R}_t = \sum_{i=1}^N x_i R_{it}.$$

Substituting in (6.1) gives the CER model for the portfolio returns,

$$R_{pt} = \mathbf{x}'(\mu + \varepsilon_t) = \mathbf{x}'\mu + \mathbf{x}'\varepsilon_t = \mu_p + \varepsilon_{pt} \quad (6.6)$$

where $\mu_p = \mathbf{x}'\mu = \sum_{i=1}^N x_i \mu_i$ is the portfolio expected return, and $\varepsilon_{pt} = \mathbf{x}'\varepsilon_t = \sum_{i=1}^N x_i \varepsilon_{it}$ is the portfolio error. The variance of R_{pt} is given by:

$$\text{var}(R_{pt}) = \text{var}(\mathbf{x}'\mathbf{R}_t) = \mathbf{x}'\Sigma\mathbf{x} = \sigma_p^2.$$

Therefore, the distribution of portfolio returns is normal,

$$R_{pt} \sim N(\mu_p, \sigma_p^2).$$

This result is exact for simple returns but is often used as an approximation for continuously compounded returns.

CER Model for Multi-Period Simple Returns

The CER model for single period simple returns does not extend exactly to multi-period simple returns because multi-period simple returns are not additive. Recall, the k -period simple return has a multiplicative relationship to single period returns:

$$\begin{aligned} R_t(k) &= (1 + R_t)(1 + R_{t-1}) \times \cdots \times (1 + R_{t-k+1}) - 1 \\ &= R_t + R_{t-1} + \cdots + R_{t-k+1} \\ &\quad + R_t R_{t-1} + R_t R_{t-2} + \cdots + R_{t-k+2} R_{t-k+1}. \end{aligned}$$

Even though single period returns are normally distributed in the CER model, multi-period returns are not normally distributed because the product of two normally distributed random variables is not normally distributed. Hence, the CER model does not exactly generalize to multi-period simple returns. However, if single period returns are small then all of the cross products of returns are approximately zero ($R_t R_{t-1} \approx \cdots \approx R_{t-k+2} R_{t-k+1} \approx 0$) and,

$$\begin{aligned} R_t(k) &\approx R_t + R_{t-1} + \cdots + R_{t-k+1} \\ &\approx \mu(k) + \varepsilon_t(k). \end{aligned}$$

where $\mu(k) = k\mu$ and $\varepsilon_t(k) = \sum_{j=0}^{k-1} \varepsilon_{t-j}$. Hence, the CER model is approximately true for multi-period simple returns when single period simple returns are not too big.

Some exact returns can be derived for the mean and variance of multi-period simple returns. For simplicity, let $k = 2$ so that,

$$R_t(2) = (1 + R_t)(1 + R_{t-1}) - 1 = R_t + R_{t-1} + R_t R_{t-1}.$$

Substituting in (6.1) then gives,

$$\begin{aligned} R_t(2) &= (\mu + \varepsilon_t) + (\mu + \varepsilon_{t-1}) + (\mu + \varepsilon_t)(\mu + \varepsilon_{t-1}) \\ &= 2\mu + \varepsilon_t + \varepsilon_{t-1} + \mu^2 + \mu\varepsilon_t + \mu\varepsilon_{t-1} + \varepsilon_t\varepsilon_{t-1} \\ &= 2\mu + \mu^2 + \varepsilon_t(1 + \mu) + \varepsilon_{t-1}(1 + \mu) + \varepsilon_t\varepsilon_{t-1}. \end{aligned}$$

The result for the expected return is easy,

$$\begin{aligned} E[R_t(2)] &= 2\mu + \mu^2 + (1 + \mu)E[\varepsilon_t] + (1 + \mu)E[\varepsilon_{t-1}] + E[\varepsilon_t\varepsilon_{t-1}] \\ &= 2\mu + \mu^2 = (1 + \mu)^2 - 1, \end{aligned}$$

The result uses the independence of ε_t and ε_{t-1} to get $E[\varepsilon_t\varepsilon_{t-1}] = E[\varepsilon_t]E[\varepsilon_{t-1}] = 0$. The result for the variance, however, is more work,

$$\begin{aligned} \text{var}(R_t(2)) &= \text{var}(\varepsilon_t(1 + \mu) + \varepsilon_{t-1}(1 + \mu) + \varepsilon_t\varepsilon_{t-1}) \\ &= (1 + \mu)^2 \text{var}(\varepsilon_t) + (1 + \mu)^2 \text{var}(\varepsilon_{t-1}) + \text{var}(\varepsilon_t\varepsilon_{t-1}) \\ &\quad + 2(1 + \mu)^2 \text{cov}(\varepsilon_t, \varepsilon_{t-1}) + 2(1 + \mu) \text{cov}(\varepsilon_t, \varepsilon_t\varepsilon_{t-1}) \\ &\quad + 2(1 + \mu) \text{cov}(\varepsilon_{t-1}, \varepsilon_t\varepsilon_{t-1}) \end{aligned}$$

Now, $\text{var}(\varepsilon_t) = \text{var}(\varepsilon_{t-1}) = \sigma^2$ and $\text{cov}(\varepsilon_t, \varepsilon_{t-1}) = 0$. Next, note that,

$$\text{var}(\varepsilon_t\varepsilon_{t-1}) = E[\varepsilon_t^2\varepsilon_{t-1}^2] - (E[\varepsilon_t\varepsilon_{t-1}])^2 = E[\varepsilon_t^2]E[\varepsilon_{t-1}^2] - (E[\varepsilon_t]E[\varepsilon_{t-1}])^2 = 2\sigma^2.$$

Finally,

$$\begin{aligned}\text{cov}(\varepsilon_t, \varepsilon_t \varepsilon_{t-1}) &= E[\varepsilon_t(\varepsilon_t \varepsilon_{t-1})] - E[\varepsilon_t]E[\varepsilon_t \varepsilon_{t-1}] \\ &= E[\varepsilon_t^2]E[\varepsilon_{t-1}] - E[\varepsilon_t]E[\varepsilon_t]E[\varepsilon_{t-1}] \\ &= 0.\end{aligned}$$

Then,

$$\begin{aligned}\text{var}(R_t(2)) &= (1 + \mu)^2\sigma^2 + (1 + \mu)^2\sigma^2 + 2\sigma^2 \\ &= 2\sigma^2[(1 + \mu)^2 + 1].\end{aligned}$$

If μ is close to zero then $E[R_t(2)] \approx 2\mu$ and $\text{var}(R_t(2)) \approx 2\sigma^2$ and so the square-root-of-time rule holds approximately.

6.2 Monte Carlo Simulation of the CER Model

A simple technique that can be used to understand the probabilistic behavior of a model involves using computer simulation methods to create pseudo data from the model. The process of creating such pseudo data is called *Monte Carlo simulation*.⁴ Monte Carlo simulation of a model can be used as a first pass “reality check” of the model. If simulated data from the model do not look like the data that the model is supposed to describe, then serious doubt is cast on the model. However, if simulated data look reasonably close to the actual data then the first step reality check is passed. Ideally, one should consider many simulated samples from the model because it is possible for a given simulated sample to look strange simply because of an unusual set of random numbers. Monte Carlo simulation can also be used to create “what if?” type scenarios for a model. Different scenarios typically correspond with different model parameter values. Finally, Monte Carlo simulation can be used to study properties of statistics computed from the pseudo data from the model. For example, Monte Carlo simulation can be used to illustrate the concepts of estimator bias and confidence interval coverage probabilities.

To illustrate the use of Monte Carlo simulation, consider creating pseudo return data from the CER model (6.1) for a single asset. The steps to create a Monte Carlo simulation from the CER model are:

1. Fix values for the CER model parameters μ and σ .
2. Determine the number of simulated values, T , to create.
3. Use a computer random number generator to simulate T *iid* values of ε_t from a $N(0, \sigma^2)$ distribution. Denote these simulated values as $\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_T$.
4. Create the simulated return data $\tilde{R}_t = \mu + \tilde{\varepsilon}_t$ for $t = 1, \dots, T$.

Example 6.1. Microsoft data to calibrate univariate Monte Carlo simulation of CER model.

To motivate plausible values for μ and σ in the simulation, Figure 6.1 shows the monthly continuously compounded returns on Microsoft stock over the period January 1998 through

⁴ Monte Carlo refers to the famous city in Monaco where gambling is legal.

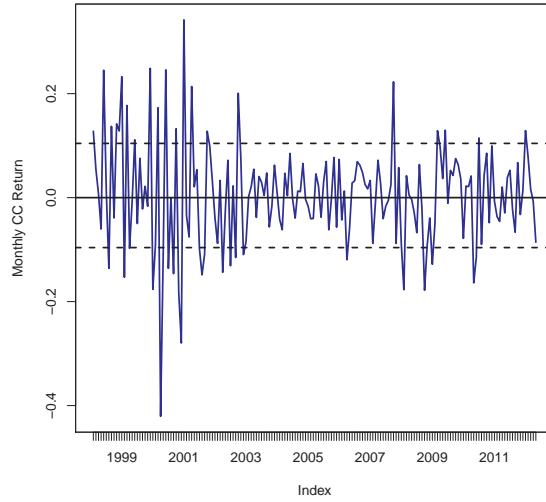


Fig. 6.1 Monthly continuously compounded returns on Microsoft. Dashed lines indicate $\hat{\mu} \pm \hat{\sigma}$.

May 2012. The data is the same as that used in Chapter 5 and is contructed from the **IntroCompFinR** “xts” object `msftDailyPrices` as follows:

```
library(IntroCompFinR)
library(PerformanceAnalytics)
Sys.setenv(TZ = "UTC")
data(msftDailyPrices)
msftPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
smp1 = "1998-01::2012-05"
msftPrices = msftPrices[smp1]
msftRetC = na.omit(Return.calculate(msftPrices, method = "log"))
head(msftRetC, n = 3)

##          MSFT
## Feb 1998 0.127864
## Mar 1998 0.054207
## Apr 1998 0.006886
```

The parameter $\mu = E[R_t]$ in the CER model is the expected monthly return, and σ represents the typical size of a deviation about μ . In Figure 6.1, the returns seem to fluctuate up and down about a central value near 0 and the typical size of a return deviation about 0 is roughly 0.10, or 10% (see dashed lines in Figure 6.1). The sample mean turns out to be $\hat{\mu} = 0.004$ (0.4%) and the sample standard deviation is $\hat{\sigma} = 0.100$ (10%). Figure 6.2 shows three distribution summaries (histogram, boxplot and normal qq-plot) and the SACF. The returns look to have slightly fatter tails than the normal distribution and show little evidence of linear time dependence (autocorrelation).

Example 6.2. Simulating observations from the CER model.

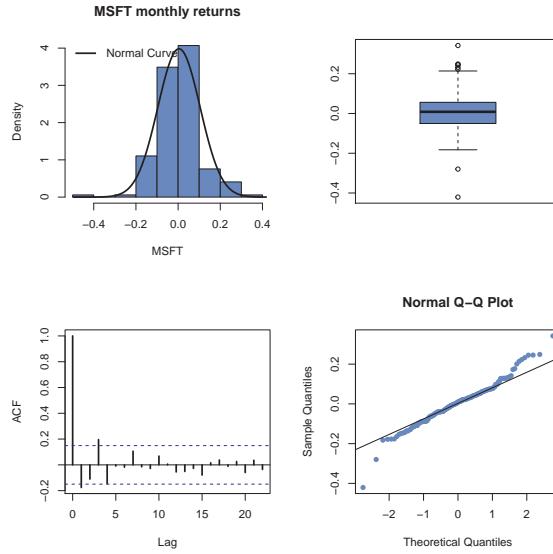


Fig. 6.2 Graphical descriptive statistics for the monthly continuously compounded returns on Microsoft.

To mimic the monthly return data on Microsoft in the Monte Carlo simulation, the values $\mu = 0.004$ and $\sigma = 0.10$ are used as the model's true parameters and $T = 172$ is the number of simulated values (sample size of actual data). Let $\{\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_{172}\}$ denote the 172 simulated values of the news variable $\varepsilon_t \sim \text{GWN}(0, (0.10)^2)$. The simulated returns are then computed using:⁵

$$\tilde{R}_t = 0.004 + \tilde{\varepsilon}_t, \quad t = 1, \dots, 172 \quad (6.7)$$

To create and plot the simulated returns from (6.7) use:

```
mu = 0.004
sd.e = 0.1
n.obs = length(msftRetC)
set.seed(111)
sim.e = rnorm(n.obs, mean = 0, sd = sd.e)
sim.ret = mu + sim.e
sim.ret = xts(sim.ret, index(msftRetC))
plot.zoo(sim.ret, main = "", lwd = 2, col = "blue", ylab = "Monthly CC Return")
abline(h = 0, lwd = 2)
abline(h = (mu + sd.e), lty = "dashed", lwd = 2)
abline(h = (mu - sd.e), lty = "dashed", lwd = 2)
```

The simulated returns $\{\tilde{R}_t\}_{t=1}^{172}$ (with the same time index as the Microsoft returns) are shown in Figure 6.3. The simulated return data fluctuate randomly about $\mu = 0.004$, and the typical size of the fluctuation is approximately equal to $\sigma = 0.10$. The simulated return data look somewhat like the actual monthly return data for Microsoft. The main difference is that the return volatility for Microsoft appears to have decreased in the latter part of the sample whereas the simulated data has constant volatility over the entire sample. Figure 6.4

⁵ Alternatively, the returns can be simulated directly by simulating observations from a normal distribution with mean 0.0 and standard deviation 0.10.

shows the distribution summaries (histogram, boxplot and normal qq-plot) and the SACF for the simulated returns. The simulated returns are normally distributed and show thinner tails than the actual returns. The simulated returns also show no evidence of linear time dependence (autocorrelation).

■

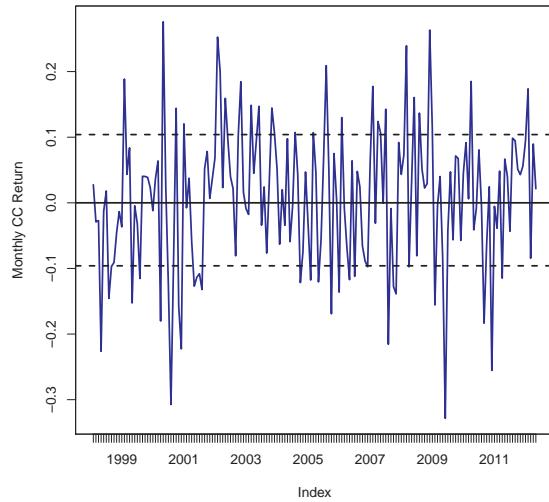


Fig. 6.3 Monte Carlo simulated returns from the CER model for Microsoft.

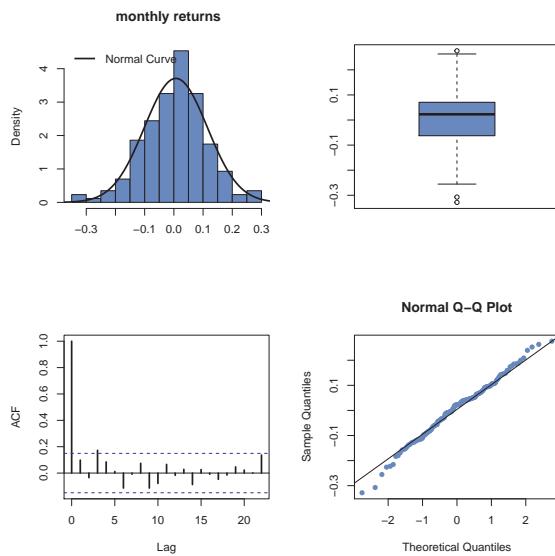


Fig. 6.4 Graphical descriptive statistics for the Monte Carlo simulated returns on Microsoft.

Example 6.3. Simulating log-prices from the RW model.

The RW model for log-price based on the CER model (6.7) calibrated to Microsoft log prices is:

$$p_t = 2.592 + 0.004 \cdot t + \sum_{j=1}^t \varepsilon_j, \varepsilon_t \sim \text{GWN}(0, (0.10)^2).$$

where $p_0 = 2.592 = \ln(13.36)$ is the log of first Microsoft Price. A Monte Carlo simulation of this RW model with drift can be created in R using:

```
sim.p = 2.592 + mu * seq(n.obs) + cumsum(sim.e)
sim.P = exp(sim.p)
# decompose RW into components
sim.sume = xts(cumsum(sim.e), index(sim.ret))
sim.p = xts(sim.p, index(sim.ret))
sim.mt = sim.p - sim.sume
sim.P = xts(sim.P, index(sim.ret))
dataToPlot = merge(sim.sume, sim.mt, sim.p, sim.P)
```

Figure 6.5 shows the simulated values as well as the individual random walk with drift model components. The top panel shows the simulated log price, \tilde{p}_t (blue solid line), the expected price $E[\tilde{p}_t] = 2.592 + 0.004 \cdot t$ (green dashed line) and the accumulated random news $\tilde{p}_t - E[\tilde{p}_t] = \sum_{s=1}^t \tilde{\varepsilon}_s$ (dotted red line). The bottom panel shows the simulated price levels $\tilde{P}_t = e^{\tilde{p}_t}$ (solid black line). Figure 6.6 shows the actual log prices and price levels for Microsoft stock. Notice the similarity between the simulated random walk data and the actual data.

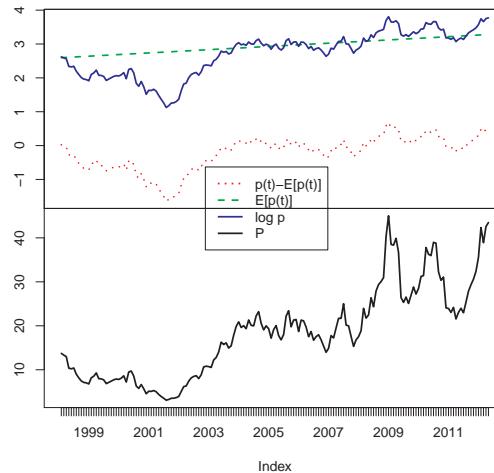


Fig. 6.5 Top panel: Random walk decomposition of simulated log monthly prices of Microsoft. Bottom panel: Simulated monthly random walk price of Microsoft stock.

6.2.1 Simulating returns on more than one asset

Creating a Monte Carlo simulation of more than one return from the CER model requires simulating observations from a multivariate normal distribution. This follows from the matrix representation of the CER model given in (6.3). The steps required to create a multivariate Monte Carlo simulation are:

1. Fix values for $N \times 1$ mean vector μ and the $N \times N$ covariance matrix Σ .
2. Determine the number of simulated values, T , to create.
3. Use a computer random number generator to simulate T iid values of the $N \times 1$ random vector ε_t from the multivariate normal distribution $N(\mathbf{0}, \Sigma)$. Denote these simulated vectors as $\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_T$.
4. Create the $N \times 1$ simulated return vector $\tilde{R}_t = \mu + \tilde{\varepsilon}_t$ for $t = 1, \dots, T$.

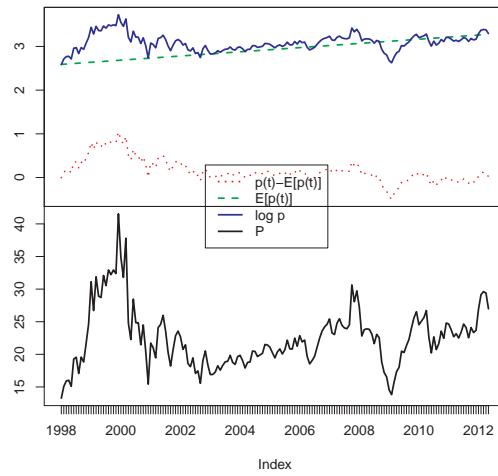


Fig. 6.6 Top panel: Random walk decomposition of the log monthly price of Microsoft. Bottom panel: Monthly price of Microsoft stock.

Example 6.4. Microsoft, Starbucks and S&P 500 data to calibrate multivariate Monte Carlo simulation of CER model.

To motivate the parameters for a multivariate simulation of the CER model, consider the monthly continuously compounded returns for Microsoft, Starbucks and the S&P 500 index over the period January 1998 through May 2012 illustrated in Figures 6.7 and 6.8. The data is assembled using the R commands:

```
# get data from IntroCompFin package
data(sp500DailyPrices, sbuxDailyPrices)
sp500Prices = to.monthly(sp500DailyPrices, OHLC = FALSE)
```

```

sbuxPrices = to.monthly(sbuxDailyPrices, OHLC = FALSE)
# set sample to match book chapter
sp500Prices = sp500Prices[smpl]
sbuxPrices = sbuxPrices[smpl]
# calculate returns
sp500RetC = na.omit(Return.calculate(sp500Prices, method = "log"))
sbuxRetC = na.omit(Return.calculate(sbuxPrices, method = "log"))
# merged data set
cerRetC = merge(msftRetC, sbuxRetC, sp500RetC)
colnames(cerRetC) = c("MSFT", "SBUX", "SP500")
head(cerRetC, n = 3)

##           MSFT      SBUX      SP500
## Feb 1998 0.127864 0.07932 0.068078
## Mar 1998 0.054207 0.13435 0.048738
## Apr 1998 0.006886 0.06096 0.009036

```

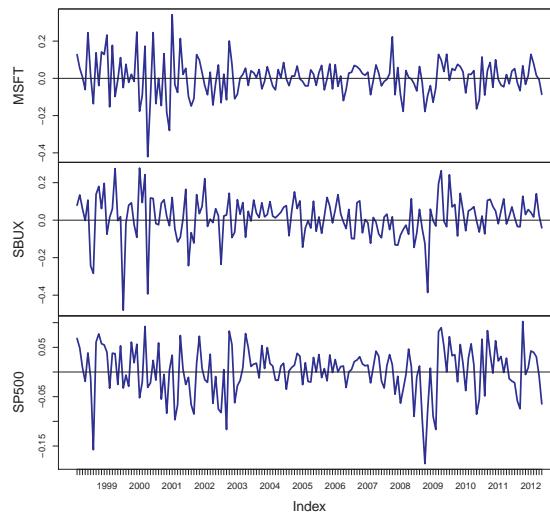


Fig. 6.7 Monthly continuously compounded returns on Microsoft, Starbucks, and the S&P 500 Index.

The multivariate sample descriptive statistics (mean vector, standard deviation vector, covariance matrix and correlation matrix) are:

```

apply(cerRetC, 2, mean)

##           MSFT      SBUX      SP500
## 0.004126 0.014657 0.001687

apply(cerRetC, 2, sd)

##           MSFT      SBUX      SP500
## 0.10021 0.11164 0.04847

cov(cerRetC)

```

```

##           MSFT      SBUX      SP500
## MSFT  0.010043 0.003814 0.002998
## SBUX  0.003814 0.012465 0.002476
## SP500 0.002998 0.002476 0.002349

cor(cerRetC)

##           MSFT      SBUX      SP500
## MSFT  1.0000 0.3409 0.6172
## SBUX  0.3409 1.0000 0.4575
## SP500 0.6172 0.4575 1.0000

```

All returns fluctuate around mean values close to zero. The volatilities of Microsoft and Starbucks are similar with typical magnitudes around 0.10, or 10%. The volatility of the S&P 500 index is considerably smaller at about 0.05, or 5%. The pairwise scatterplots show that all returns are positively related. The pairs (MSFT, SP500) and (SBUX, SP500) are the most correlated with sample correlation values around 0.5. The pair (MSFT, SBUX) has a moderate positive correlation around 0.3.

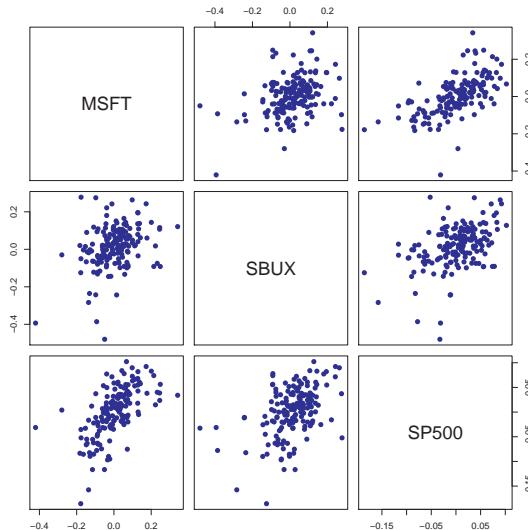


Fig. 6.8 Pairwise scatterplots of the monthly returns on Microsoft, Starbucks, and the S&P 500 Index.

Example 6.5. Monte Carlo simulation of CER model for three assets.

Simulating values from the multivariate CER model (6.3) requires simulating multivariate normal random variables. In R, this can be done using the function `rmvnorm()` from the package `mvtnorm`. The function `rmvnorm()` requires a vector of mean values and a covariance matrix. Define:

$$\mathbf{R}_t = \begin{pmatrix} R_{\text{msft},t} \\ R_{\text{sbux},t} \\ R_{\text{sp500},t} \end{pmatrix}, \mu = \begin{pmatrix} \mu_{\text{msft},t} \\ \mu_{\text{sbux},t} \\ \mu_{\text{sp500},t} \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_{\text{msft}}^2 & \sigma_{\text{msft,sbux}} & \sigma_{\text{msft,sp500}} \\ \sigma_{\text{msft,sbux}} & \sigma_{\text{sbux}}^2 & \sigma_{\text{sbux,sp500}} \\ \sigma_{\text{msft,sp500}} & \sigma_{\text{sbux,sp500}} & \sigma_{\text{sp500}}^2 \end{pmatrix}$$

The parameters μ and Σ of the multivariate CER model are set equal to the sample mean vector μ and sample covariance matrix Σ :

```
muVec = apply(cerRetC, 2, mean)
covMat = cov(cerRetC)
```

To create a Monte Carlo simulation from the CER model calibrated to the month continuously returns on Microsoft, Starbucks and the S&P 500 index use:

```
library(mvtnorm)
set.seed(123)
returns.sim = rmvnorm(n.obs, mean = muVec, sigma = covMat)
colnames(returns.sim) = colnames(cerRetC)
returns.sim = xts(returns.sim, index(cerRetC))
```

The simulated returns are shown in Figures 6.9 and 6.10. They look similar to the actual returns shown in Figures 6.7 and 6.8. The actual returns show periods of high and low volatility that the simulated returns do not. The sample statistics from the simulated returns, however, are close to the sample statistics of the actual data:

```
apply(returns.sim, 2, mean)

##      MSFT      SBUX      SP500
## 0.006707 0.013812 0.005080

apply(returns.sim, 2, sd)

##      MSFT      SBUX      SP500
## 0.09509 0.10512 0.04601

cov(returns.sim)

##            MSFT      SBUX      SP500
## MSFT 0.009043 0.003534 0.002463
## SBUX 0.003534 0.011050 0.001942
## SP500 0.002463 0.001942 0.002117

cor(returns.sim)

##            MSFT      SBUX      SP500
## MSFT 1.0000 0.3535 0.5629
## SBUX 0.3535 1.0000 0.4015
## SP500 0.5629 0.4015 1.0000
```

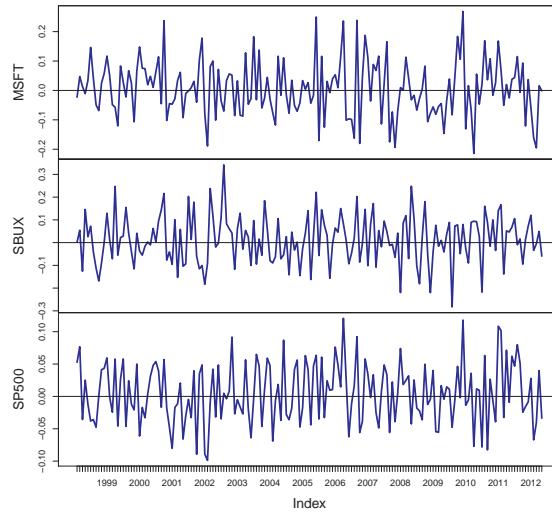


Fig. 6.9 Simulated CER model returns on Microsoft, Starbucks and the S&P500 Index.

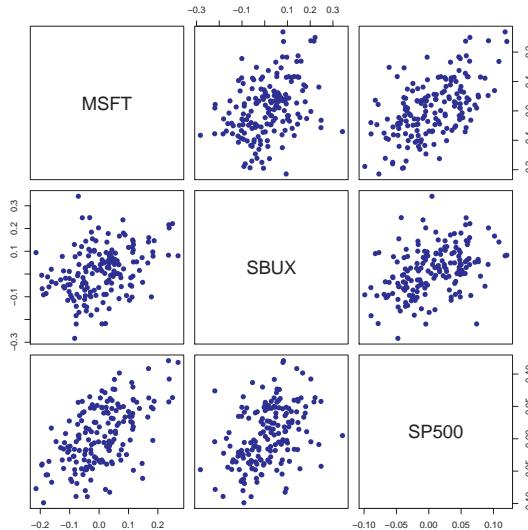


Fig. 6.10 Pairwise scatterplot of simulated returns

6.3 Further Reading

The model name “Constant Expected Return” is taken from Campbell, Lo and MacKinlay (1987) who use it to illustrate event study methodology. The location-scale form of the CER model (6.2) is commonly used as a starting point for the distributional analysis of returns. In this chapter, and throughout most of the book, we use the standard normal distribution for the standardized news shock. However, many other standardized distributions (e.g., skew-

normal, Student's t, skewed-t, etc.) can also be used as illustrated in Jondeau et al. (2007), Pfaff (2013), Carmona (2014), and Ruppert and Matteson (2015).

6.4 Problems

Exercise 6.1. Consider the CER model for continuously compounded returns:

$$r_t = \mu + \epsilon_t, \epsilon_t \sim \text{GWN}(0, \sigma^2).$$

It implies that the log price follows a random walk with drift:

$$\ln P_t = \ln P_{t-1} + r_t = \ln P_0 + \mu t + \sum_{s=1}^t \epsilon_s.$$

Show that $E[\ln P_t]$ and $\text{var}[\ln P_t]$ depend on t so that $\ln P_t$ is non-stationary.

Exercise 6.2. Consider using Monte Carlo simulation to evaluate the CER model for the monthly simple returns on Amazon stock and the S&P 500 index over the 5 year period January 2009, through January 2014.

1. Daily adjusted closing prices on Amazon stock and the S&P 500 index are in the **IntroCompFinR** "xts" objects `amznDailyPrices` and `sp500DailyPrices`, respectively. Using these prices, create monthly returns over the period January 2009, through January 2014.
2. Create a time plot showing the monthly returns on the two assets. Do the monthly returns from the two assets look like realizations from a covariance stationary stochastic process? Why or why not?
3. Compare and contrast the return characteristics of the two assets. In addition, comment on any common features, if any, of the two return series.
4. Using the **IntroCompFinR** function `fourPanelPlot()`, create graphical descriptive statistics for the monthly returns. Also, create the scatterplot between the two return series. Do the returns look normally distributed? Is there any evidence for linear time dependence? Do the return appear to contemporaneously correlated?
5. Compute sample descriptive statistics (mean, standard deviation, skewness, kurtosis, and correlation between Amazon and the S&P 500 index).

References

1. Campbell, Lo and MacKinley (1998). *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ.
2. Carmona, R. (2014). *Statistical Analysis of Financial Data in R, Second Edition*. Springer, New York
3. Jondeau, R., Poon, S.-H., and Rockinger, M. (2007). *Financial Modeling Under Non-Gaussian Distributions*. Springer, New York.
4. Pfaff, B. (2013). *Financial Risk Modelling and Portfolio Optimization with R*. Wiley, New York.
5. Ruppert, D., and Matteson, D.S. (2015). *Statistics and Data Analysis for Financial Engineering with R examples*. Springer, New York.

Chapter 7

Estimation of The CER Model

Updated: July 22, 2016

Copyright © Eric Zivot 2015, 2016

The CER model of asset returns presented in the previous chapter gives us a simple framework for interpreting the time series behavior of asset returns and prices. At the beginning of time $t - 1$, \mathbf{R}_t is an $N \times 1$ random vector representing the returns (simple or continuously compounded) on assets $i = 1, \dots, N$ to be realized at time t . The CER model states that $\mathbf{R}_t \sim iid N(\mu, \Sigma)$. Our best guess for the return at t on asset i is $E[R_{it}] = \mu_i$, our measure of uncertainty about our best guess is captured by $SD(R_{it}) = \sigma_i$, and our measures of the direction and strength of linear association between R_{it} and R_{jt} are $\sigma_{ij} = \text{cov}(R_{it}, R_{jt})$ and $\rho_{ij} = \text{cor}(R_{it}, R_{jt})$, respectively. The CER model assumes that the economic environment is constant over time so that the multivariate normal distribution characterizing returns is the same for all time periods t .

Our life would be very easy if we knew the exact values of μ_i , σ_i^2 , σ_{ij} and ρ_{ij} , the parameters of the CER model. Then we could use the CER model for risk and portfolio analysis. In actuality, however, we do not know these values with certainty. Therefore, a key task in financial econometrics is estimating these values from a history of observed return data. Given estimates of the CER model parameters we can then apply the model to risk and portfolio analysis.

Suppose we observe returns on N different assets over the sample $t = 1, \dots, T$. Denote this sample $\{\mathbf{r}_1, \dots, \mathbf{r}_T\} = \{\mathbf{r}_t\}_{t=1}^T$, where $\mathbf{r}_t = (\mathbf{r}_{1t}, \dots, \mathbf{r}_{Nt})'$ is the $N \times 1$ vector of returns on N assets observed at time t . It is assumed that the observed returns are realizations of the random variables $\{\mathbf{R}_t\}_{t=1}^T$, where $\mathbf{R}_t = (R_{1t}, \dots, R_{Nt})'$ is a vector of N asset returns described by the CER model:

$$\mathbf{R}_t = \mu + \varepsilon_t, \quad \varepsilon_t \sim iid N(\mathbf{0}, \Sigma). \quad (7.1)$$

Under these assumptions, we can use the observed returns $\{\mathbf{r}_t\}_{t=1}^T$ to estimate the unknown parameters in μ and Σ of the CER model. However, before we describe the estimation of the CER model in detail, it is necessary to review some fundamental concepts in the statistical theory of estimation. This is important because estimates of parameters have estimation

error and statistical theory allows us to evaluate characteristics of estimation error and to develop quantitative measures of the typical magnitude of estimation error. As we shall see, some parameters of the CER model are precisely estimated (small estimation error) and some parameters are not (large estimation error).

The chapter is outlined as follows. Section 7.1 reviews some of the statistical theory of estimation and discusses some important finite sample properties of estimators such as bias and precision. For many estimators it difficult to derive exact finite sample properties which motivates the need to look at asymptotic properties of estimators. These are properties that hold exactly as the sample size goes to infinity but are used as approximations for a finite sample size. Estimators of the CER model parameters are presented in section 7.2 and the statistical properties of these estimators are investigated in section 7.3. Section 7.4 illustrates how Monte Carlo simulation can be used to evaluate and understand the statistical properties of the CER model estimators. Section 7.5 wraps up the chapter by discussing estimators for value-at-risk using the CER model.

In this chapter we use the R packages **IntroCompFinR**, **mvtnorm**, and **PerformanceAnalytics**. Make sure these packages are installed and loaded prior to running the R examples in the chapter.

7.1 Estimators and Estimates

Let R_t be the return on a single asset described by the CER model and let θ denote some characteristic (parameter) of the CER model we are interested in estimating. For simplicity, assume that $\theta \in \mathbb{R}$ is a single parameter. For example, if we are interested in the expected return on the asset, then $\theta = \mu$; if we are interested in the variance of asset i returns, then $\theta = \sigma^2$; if we are interested in the first lag autocorrelation then $\theta = \rho_1$. The goal is to estimate θ based on a sample of size T of the observed data.

Definition 7.1. Let $\{R_1, \dots, R_T\}$ denote a collection of T random returns from the CER model, and let θ denote some characteristic of the model. An estimator of θ , denoted $\hat{\theta}$, is a rule or algorithm for estimating θ as a function of the random variables $\{R_1, \dots, R_T\}$. Here, $\hat{\theta}$ is a random variable.

Definition 7.2. Let $\{r_1, \dots, r_T\}$ denote an observed sample of size T from the CER model, and let θ denote some characteristic of the model. An estimate of θ , denoted $\hat{\theta}$, is simply the value of the estimator for θ based on the observed sample $\{r_1, \dots, r_T\}$. Here, $\hat{\theta}$ is a number.

Example 7.1. The sample average as an estimator and an estimate.

Let R_t be the return on a single asset described by the CER model, and suppose we are interested in estimating $\theta = \mu = E[R_t]$ from the sample of observed returns $\{r_t\}_{t=1}^T$. The sample average $\hat{\mu} = \frac{1}{T} \sum_{t=1}^T r_t$ is an algorithm for computing an estimate of the expected return μ . Before the sample is observed, we can think of $\hat{\mu} = \frac{1}{T} \sum_{t=1}^T R_t$ as a simple linear function of the random variables $\{R_t\}_{t=1}^T$ and so is itself a random variable. After the

sample is observed, the sample average can be evaluated using the observed data $\{r_t\}_{t=1}^T$ which produces the estimate of μ . For example, suppose $T = 5$ and the realized values of the returns are $r_1 = 0.1, r_2 = 0.05, r_3 = 0.025, r_4 = -0.1, r_5 = -0.05$. Then the estimate of μ using the sample average is:

$$\hat{\mu} = \frac{1}{5}(0.1 + 0.05 + 0.025 + -0.1 + -0.05) = 0.005.$$

■

The example above illustrates the distinction between an estimator and an estimate of a parameter θ . However, typically in the statistics literature we use the same symbol, $\hat{\theta}$, to denote both an estimator and an estimate. When $\hat{\theta}$ is treated as a function of the random returns it denotes an estimator and is a random variable. When $\hat{\theta}$ is evaluated using the observed data it denotes an estimate and is simply a number. The context in which we discuss $\hat{\theta}$ will determine how to interpret it.

7.1.1 Properties of Estimators

Consider $\hat{\theta}$ as a random variable. In general, the pdf of $\hat{\theta}$, $f(\hat{\theta})$, depends on the pdf's of the random variables $\{R_t\}_{t=1}^T$. The exact form of $f(\hat{\theta})$ may be very complicated. Sometimes we can use analytic calculations to determine the exact form of $f(\hat{\theta})$. In general, the exact form of $f(\hat{\theta})$ is often too difficult to derive exactly. When $f(\hat{\theta})$ is too difficult to compute we can often approximate $f(\hat{\theta})$ using either Monte Carlo simulation techniques or the *Central Limit Theorem* (CLT). In Monte Carlo simulation, we use the computer to simulate many different realizations of the random returns $\{R_t\}_{t=1}^T$ and on each simulated sample we evaluate the estimator $\hat{\theta}$. The Monte Carlo approximation of $f(\hat{\theta})$ is the empirical distribution $\hat{\theta}$ over the different simulated samples. For a given sample size T , Monte Carlo simulation gives a very accurate approximation to $f(\hat{\theta})$ if the number of simulated samples is very large. The CLT approximation of $f(\hat{\theta})$ is a normal distribution approximation that becomes more accurate as the sample size T gets very large. An advantage of the CLT approximation is that it is often easy to compute. The disadvantage is that the accuracy of the approximation depends on the estimator $\hat{\theta}$ and sample size T .

For analysis purposes, we often focus on certain characteristics of $f(\hat{\theta})$, like its expected value (center), variance and standard deviation (spread about expected value). The expected value of an estimator is related to the concept of estimator *bias*, and the variance/standard deviation of an estimator is related to the concept of estimator *precision*. Different realizations of the random variables $\{R_t\}_{t=1}^T$ will produce different values of $\hat{\theta}$. Some values of $\hat{\theta}$ will be bigger than θ and some will be smaller. Intuitively, a *good* estimator of θ is one that is on average correct (unbiased) and never gets too far away from θ (small variance). That is, a good estimator will have small bias and high precision.

Bias

Bias concerns the location or center of $f(\hat{\theta})$ in relation to θ . If $f(\hat{\theta})$ is centered away from θ , then we say $\hat{\theta}$ is a *biased* estimator of θ . If $f(\hat{\theta})$ is centered at θ , then we say that $\hat{\theta}$ is an *unbiased* estimator of θ . Formally, we have the following definitions:

Definition 7.3. The estimation error is the difference between the estimator and the parameter being estimated:

$$\text{error}(\hat{\theta}, \theta) = \hat{\theta} - \theta. \quad (7.2)$$

Definition 7.4. The bias of an estimator $\hat{\theta}$ of θ is the expected estimation error:

$$\text{bias}(\hat{\theta}, \theta) = E[\text{error}(\hat{\theta}, \theta)] = E[\hat{\theta}] - \theta. \quad (7.3)$$

Definition 7.5. An estimator $\hat{\theta}$ of θ is unbiased if $\text{bias}(\hat{\theta}, \theta) = 0$; i.e., if $E[\hat{\theta}] = \theta$ or $E[\text{error}(\hat{\theta}, \theta)] = 0$.

Unbiasedness is a desirable property of an estimator. It means that the estimator produces the correct answer "on average", where "on average" means over many hypothetical realizations of the random variables $\{R_t\}_{t=1}^T$. It is important to keep in mind that an unbiased estimator for θ may not be very close to θ for a particular sample, and that a biased estimator may actually be quite close to θ . For example, consider two estimators of θ , $\hat{\theta}_1$ and $\hat{\theta}_2$. The pdfs of $\hat{\theta}_1$ and $\hat{\theta}_2$ are illustrated in Figure 7.1. $\hat{\theta}_1$ is an unbiased estimator of θ with a large variance, and $\hat{\theta}_2$ is a biased estimator of θ with a small variance. Consider first, the pdf of $\hat{\theta}_1$. The center of the distribution is at the true value $\theta = 0$, $E[\hat{\theta}_1] = 0$, but the distribution is very widely spread out about $\theta = 0$. That is, $\text{var}(\hat{\theta}_1)$ is large. On average (over many hypothetical samples) the value of $\hat{\theta}_1$ will be close to θ , but in any given sample the value of $\hat{\theta}_1$ can be quite a bit above or below θ . Hence, unbiasedness by itself does not guarantee a good estimator of θ . Now consider the pdf for $\hat{\theta}_2$. The center of the pdf is slightly higher than $\theta = 0$, i.e., $\text{bias}(\hat{\theta}_2, \theta) > 0$, but the spread of the distribution is small. Although the value of $\hat{\theta}_2$ is not equal to 0 *on average* we might prefer the estimator $\hat{\theta}_2$ over $\hat{\theta}_1$ because it is generally closer to $\theta = 0$ *on average* than $\hat{\theta}_1$.

While unbiasedness is a desirable property of an estimator $\hat{\theta}$ of θ , it, by itself, is not enough to determine if $\hat{\theta}$ is a good estimator. Being correct on average means that $\hat{\theta}$ is seldom exactly correct for any given sample. In some samples $\hat{\theta}$ is less than θ , and some samples $\hat{\theta}$ is greater than θ . More importantly, we need to know how far $\hat{\theta}$ typically is from θ . That is, we need to know about the magnitude of the spread of the distribution of $\hat{\theta}$ about its average value. This will tell us the precision of $\hat{\theta}$.

Precision

An estimate is, hopefully, our best guess of the true (but unknown) value of θ . Our guess most certainly will be wrong, but we hope it will not be too far off. A precise estimate is one in which the variability of the estimation error is small. The variability of the estimation error is captured by the *mean squared error*.

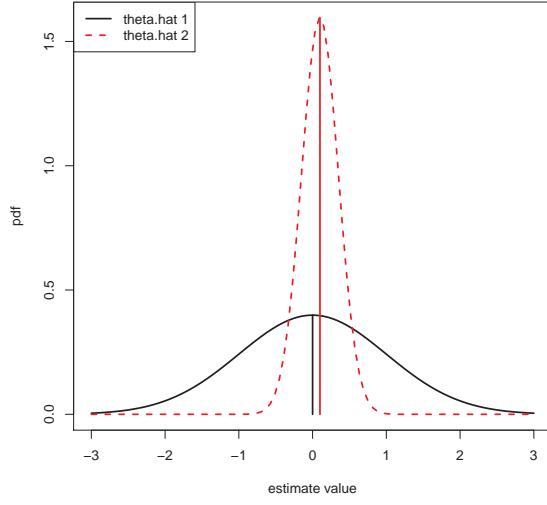


Fig. 7.1 Distributions of competing estimators for $\theta = 0$. $\hat{\theta}_1$ is unbiased but has high variance, and $\hat{\theta}_2$ is biased but has low variance.

Definition 7.6. The mean squared error of an estimator $\hat{\theta}$ of θ is given by:

$$\text{mse}(\hat{\theta}, \theta) = E[(\hat{\theta} - \theta)^2] = E[\text{error}(\hat{\theta}, \theta)^2] \quad (7.4)$$

The mean squared error measures the expected squared deviation of $\hat{\theta}$ from θ . If this expected deviation is small, then we know that $\hat{\theta}$ will almost always be close to θ . Alternatively, if the mean squared is large then it is possible to see samples for which $\hat{\theta}$ is quite far from θ . A useful decomposition of $\text{mse}(\hat{\theta}, \theta)$ is:

$$\text{mse}(\hat{\theta}, \theta) = E[(\hat{\theta} - E[\hat{\theta}])^2] + (E[\hat{\theta}] - \theta)^2 = \text{var}(\hat{\theta}) + \text{bias}(\hat{\theta}, \theta)^2$$

The derivation of this result is straightforward. Write,

$$\hat{\theta} - \theta = \hat{\theta} - E[\hat{\theta}] + E[\hat{\theta}] - \theta.$$

Then,

$$(\hat{\theta} - \theta)^2 = (\hat{\theta} - E[\hat{\theta}])^2 + 2(\hat{\theta} - E[\hat{\theta}])(E[\hat{\theta}] - \theta) + (E[\hat{\theta}] - \theta)^2.$$

Taking expectations of both sides gives,

$$\begin{aligned} \text{mse}(\hat{\theta}, \theta) &= E[(\hat{\theta} - E[\hat{\theta}])^2] + 2(E[\hat{\theta}] - E[\hat{\theta}])(E[\hat{\theta}] - \theta) + E[(E[\hat{\theta}] - \theta)^2] \\ &= E[(\hat{\theta} - E[\hat{\theta}])^2] + E[(E[\hat{\theta}] - \theta)^2] \\ &= \text{var}(\hat{\theta}) + \text{bias}(\hat{\theta}, \theta)^2. \end{aligned}$$

The result states that for any estimator $\hat{\theta}$ of θ , $\text{mse}(\hat{\theta}, \theta)$ can be split into a variance component, $\text{var}(\hat{\theta})$, and a bias component, $\text{bias}(\hat{\theta}, \theta)^2$. Clearly, $\text{mse}(\hat{\theta}, \theta)$ will be small only if both components are small. If an estimator is unbiased then $\text{mse}(\hat{\theta}, \theta) = \text{var}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$ is just the squared deviation of $\hat{\theta}$ about θ . Hence, an unbiased estimator $\hat{\theta}$ of θ is *good*, if it has a small variance.

The $\text{mse}(\hat{\theta}, \theta)$ and $\text{var}(\hat{\theta})$ are based on squared deviations and so are not in the same units of measurement as θ . Measures of precision that are in the same units as θ are the *root mean square error*:

$$\text{rmse}(\hat{\theta}, \theta) = \sqrt{\text{mse}(\hat{\theta}, \theta)}, \quad (7.5)$$

and the *standard error*,

$$\text{se}(\hat{\theta}) = \sqrt{\text{var}(\hat{\theta})}. \quad (7.6)$$

If $\text{bias}(\hat{\theta}, \theta) \approx 0$ then the precision of $\hat{\theta}$ is typically measured by $\text{se}(\hat{\theta})$.

Good Estimators

With the concepts of bias and precision in hand, we can state what defines a good estimator.

Definition 7.7. A good estimator $\hat{\theta}$ of θ has a small bias (7.3) and a small standard error (7.6).

7.1.2 Asymptotic Properties of Estimators

Estimator bias and precision are finite sample properties. That is, they are properties that hold for a fixed sample size T . Very often we are also interested in properties of estimators when the sample size T gets very large. For example, analytic calculations may show that the bias and mse of an estimator $\hat{\theta}$ depend on T in a decreasing way. That is, as T gets very large the bias and mse approach zero. So for a very large sample, $\hat{\theta}$ is effectively unbiased with high precision. In this case we say that $\hat{\theta}$ is a *consistent estimator* of θ . In addition, for large samples the Central Limit Theorem (CLT) says that $f(\hat{\theta})$ can often be well approximated by a normal distribution. In this case, we say that $\hat{\theta}$ is *asymptotically normally distributed*. The word "asymptotic" means "in an infinitely large sample" or "as the sample size T goes to infinity". Of course, in the real world we don't have an infinitely large sample and so the asymptotic results are only approximations. How good these approximations are for a given sample size T depends on the context. Monte Carlo simulations (see section 7.4) can often be used to evaluate asymptotic approximations in a given context.

Consistency

Let $\hat{\theta}$ be an estimator of θ based on the random returns $\{R_t\}_{t=1}^T$.

Definition 7.8. $\hat{\theta}$ is consistent for θ (converges in probability to θ) if for any $\varepsilon > 0$:

$$\lim_{T \rightarrow \infty} \Pr(|\hat{\theta} - \theta| > \varepsilon) = 0.$$

Intuitively, consistency says that as we get enough data then $\hat{\theta}$ will eventually equal θ . In other words, if we have enough data then we know the truth.

Theorems in probability theory known as *Laws of Large Numbers* are used to determine if an estimator is consistent or not. In general, we have the following result: an estimator $\hat{\theta}$ is consistent for θ if:

- $\text{bias}(\hat{\theta}, \theta) = 0$ as $T \rightarrow \infty$.
- $\text{se}(\hat{\theta}) = 0$ as $T \rightarrow \infty$.

Equivalently, $\hat{\theta}$ is consistent for θ if $\text{mse}(\hat{\theta}, \theta) \rightarrow 0$ as $T \rightarrow \infty$. Intuitively, if $f(\hat{\theta})$ collapses to θ as $T \rightarrow \infty$ then $\hat{\theta}$ is consistent for θ .

Asymptotic Normality

Let $\hat{\theta}$ be an estimator of θ based on the random returns $\{R_t\}_{t=1}^T$.

Definition 7.9. An estimator $\hat{\theta}$ is asymptotically normally distributed if:

$$\hat{\theta} \sim N(\theta, \text{se}(\hat{\theta})^2) \tag{7.7}$$

for large enough T .

Asymptotic normality means that $f(\hat{\theta})$ is well approximated by a normal distribution with mean θ and variance $\text{se}(\hat{\theta})^2$. The justification for asymptotic normality comes from the famous *Central Limit Theorem*.

Central Limit Theorem

There are actually many versions of the CLT with different assumptions.¹ In its simplest form, the CLT says that the sample average of a collection of iid random variables X_1, \dots, X_T with $E[X_i] = \mu$ and $\text{var}(X_i) = \sigma^2$ is asymptotically normal with mean μ and variance σ^2/T . In particular, the CDF of the standardized sample mean:

$$\frac{\bar{X} - \mu}{\text{se}(\bar{X})} = \frac{\bar{X} - \mu}{\sigma/\sqrt{T}} = \sqrt{T} \left(\frac{\bar{X} - \mu}{\sigma} \right),$$

converges to the CDF of a standard normal random variable Z as $T \rightarrow \infty$. This result can be expressed as:

¹ White (1984) gives a comprehensive discussion of CLTs useful in econometrics.

$$\sqrt{T} \left(\frac{\bar{X} - \mu}{\sigma} \right) \sim Z \sim N(0, 1),$$

for large enough T . Equivalently,

$$\bar{X} \sim \mu + \frac{\sigma}{\sqrt{T}} \times Z \sim N \left(\mu, \frac{\sigma^2}{T} \right) = N \left(\mu, \text{se}(\bar{X})^2 \right),$$

for large enough T . This form shows that \bar{X} is asymptotically normal with mean μ and variance σ^2/T .

Asymptotic Confidence Intervals

For an asymptotically normal estimator $\hat{\theta}$ of θ , the precision of $\hat{\theta}$ is measured by $\text{se}(\hat{\theta})$ but is best communicated by computing a (asymptotic) *confidence interval* for the unknown value of θ . A confidence interval is an *interval estimate* of θ such that we can put an explicit probability statement about the likelihood that the interval covers θ .

The construction of an asymptotic confidence interval for θ uses the asymptotic normality result:

$$\frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})} = Z \sim N(0, 1). \quad (7.8)$$

Then, for $\alpha \in (0, 1)$, we compute a $(1 - \alpha) \cdot 100\%$ confidence interval for θ using (7.8) and the $1 - \alpha/2$ standard normal quantile (critical value) $q_{(1-\alpha/2)}^Z$ to give:

$$\Pr \left(-q_{(1-\alpha/2)}^Z \leq \frac{\hat{\theta} - \theta}{\text{se}(\hat{\theta})} \leq q_{(1-\alpha/2)}^Z \right) = 1 - \alpha,$$

which can be rearranged as,

$$\Pr \left(\hat{\theta} - q_{(1-\alpha/2)}^Z \cdot \text{se}(\hat{\theta}) \leq \mu_i \leq \hat{\theta} + q_{(1-\alpha/2)}^Z \cdot \text{se}(\hat{\theta}) \right) = 1 - \alpha.$$

Hence, the random interval,

$$[\hat{\theta} - q_{(1-\alpha/2)}^Z \cdot \text{se}(\hat{\theta}), \hat{\theta} + q_{(1-\alpha/2)}^Z \cdot \text{se}(\hat{\theta})] = \hat{\theta} \pm q_{(1-\alpha/2)}^Z \cdot \text{se}(\hat{\theta}) \quad (7.9)$$

covers the true unknown value of θ with probability $1 - \alpha$.

In practice, typical values for α are 0.05 and 0.01 for which $q_{(0.975)}^Z = 1.96$ and $q_{(0.995)}^Z = 2.58$. Then, approximate 95% and 99% asymptotic confidence intervals for θ have the form $\hat{\theta} \pm 2 \cdot \text{se}(\hat{\theta})$ and $\hat{\theta} \pm 2.5 \cdot \text{se}(\hat{\theta})$, respectively.

7.2 Estimators for the Parameters of the CER Model

Let $\{\mathbf{r}_t\}_{t=1}^T$ denote a sample of size T of observed returns on N assets from the CER model (7.1). To estimate the unknown CER model parameters $\mu_i, \sigma_i^2, \sigma_{ij}$ and ρ_{ij} from $\{\mathbf{r}_t\}_{t=1}^T$ we can use the *plug-in principle* from statistics:

Plug-in-Principle: Estimate model parameters using corresponding sample statistics.

For the CER model parameters, the plug-in principle estimates are the following sample descriptive statistics discussed in Chapter 5:

$$\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^T r_{it}, \quad (7.10)$$

$$\hat{\sigma}_i^2 = \frac{1}{T-1} \sum_{t=1}^T (r_{it} - \hat{\mu}_i)^2, \quad (7.11)$$

$$\hat{\sigma}_i = \sqrt{\hat{\sigma}_i^2}, \quad (7.12)$$

$$\hat{\sigma}_{ij} = \frac{1}{T-1} \sum_{t=1}^T (r_{it} - \hat{\mu}_i)(r_{jt} - \hat{\mu}_j), \quad (7.13)$$

$$\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\hat{\sigma}_i \hat{\sigma}_j}. \quad (7.14)$$

The plug-in principle is appropriate because the CER model parameters $\mu_i, \sigma_i^2, \sigma_{ij}$ and ρ_{ij} are characteristics of the underlying distribution of returns that are naturally estimated using sample statistics.

The plug-in principle sample statistics (7.10) - (7.14) are given for a single asset and the statistics (7.13) - (7.14) are given for one pair of assets. However, these statistics can be computed for a collection of N assets using the matrix sample statistics:

$$\hat{\mu}_{(N \times 1)} = \frac{1}{T} \sum_{t=1}^T \mathbf{r}_t = \begin{pmatrix} \hat{\mu}_1 \\ \vdots \\ \hat{\mu}_N \end{pmatrix}, \quad (7.15)$$

$$\hat{\Sigma}_{(N \times N)} = \frac{1}{T-1} \sum_{t=1}^T (\mathbf{r}_t - \hat{\mu})(\mathbf{r}_t - \hat{\mu})' = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1N} \\ \hat{\sigma}_{12} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{1N} & \hat{\sigma}_{2N} & \cdots & \hat{\sigma}_N^2 \end{pmatrix}. \quad (7.16)$$

Here $\hat{\mu}$ is called the sample mean vector and $\hat{\Sigma}$ is called the sample covariance matrix. The sample variances are the diagonal elements of $\hat{\Sigma}$ and the sample covariances are the off-diagonal elements of $\hat{\Sigma}$. To get the sample correlations, define the $N \times N$ diagonal matrix:

$$\hat{\mathbf{D}} = \begin{pmatrix} \hat{\sigma}_1 & 0 & \cdots & 0 \\ 0 & \hat{\sigma}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}_N \end{pmatrix}.$$

Then the sample correlation matrix $\hat{\mathbf{C}}$ is computed as:

$$\hat{\mathbf{C}} = \hat{\mathbf{D}}^{-1} \hat{\Sigma} \hat{\mathbf{D}}^{-1} = \begin{pmatrix} 1 & \hat{\rho}_{12} & \cdots & \hat{\rho}_{1N} \\ \hat{\rho}_{12} & 1 & \cdots & \hat{\rho}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\rho}_{1N} & \hat{\rho}_{2N} & \cdots & 1 \end{pmatrix}. \quad (7.17)$$

Here, the sample correlations are the off diagonal elements of $\hat{\mathbf{C}}$.

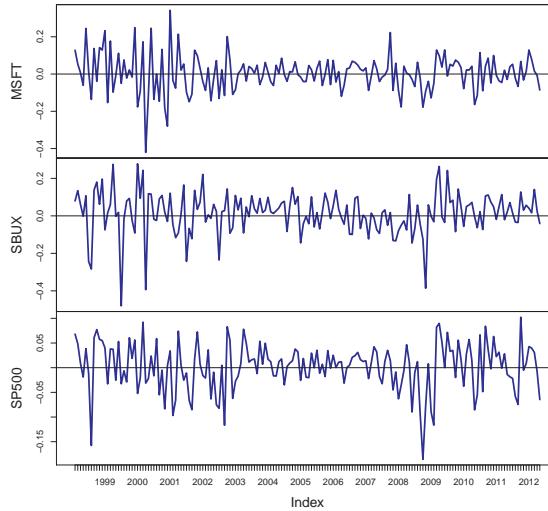


Fig. 7.2 Monthly cc returns on Microsoft stock, Starbucks stock, and the S&P 500 index, over the period January 1998 through May 2012.

Example 7.2. Estimating the CER model parameters for Microsoft, Starbucks and the S&P 500 index.

To illustrate typical estimates of the CER model parameters, we use data on monthly continuously compounded returns for Microsoft, Starbucks and the S & P 500 index over the period January 1998 through May 2012 from the **IntroCompFinR** package. The data is the same as that used in Chapter 5 and is constructed as follows:

```

library(IntroCompFinR)
library(PerformanceAnalytics)
data(msftDailyPrices, sp500DailyPrices, sbuxDailyPrices)
msftPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
sp500Prices = to.monthly(sp500DailyPrices, OHLC = FALSE)
sbuxPrices = to.monthly(sbuxDailyPrices, OHLC = FALSE)
# set sample to match examples in other chapters
smpl = "1998-01::2012-05"
msftPrices = msftPrices[smpl]
sp500Prices = sp500Prices[smpl]
sbuxPrices = sbuxPrices[smpl]
# calculate returns
msftRetS = na.omit(Return.calculate(msftPrices, method = "simple"))
sp500RetS = na.omit(Return.calculate(sp500Prices, method = "simple"))
sbuxRetS = na.omit(Return.calculate(sbuxPrices, method = "simple"))
msftRetC = log(1 + msftRetS)
sp500RetC = log(1 + sp500RetS)
sbuxRetC = log(1 + sbuxRetS)
# merged data set
cerRetS = merge(msftRetS, sbuxRetS, sp500RetS)
cerRetC = merge(msftRetC, sbuxRetC, sp500RetC)
colnames(cerRetS) = colnames(cerRetC) = c("MSFT", "SBUX", "SP500")
head(cerRetC, n = 3)

##           MSFT      SBUX      SP500
## Feb 1998 0.127864 0.07932 0.068078
## Mar 1998 0.054207 0.13435 0.048738
## Apr 1998 0.006886 0.06096 0.009036

```

These data are illustrated in Figures 7.2 and 7.3.

The estimates of μ_i ($i = \text{msft}, \text{sbux}, \text{sp500}$) using (7.10) or (7.15) can be computed using the R functions `apply()` and `mean()`:

```

muhat = apply(cerRetC, 2, mean)
muhat

##      MSFT      SBUX      SP500
## 0.004126 0.014657 0.001687

```

Starbucks has the highest average monthly return at 1.5% and the S&P 500 index has the lowest at 0.2%.

The estimates of the parameters σ_i^2, σ_i , using (7.11) and (7.12) can be computed using `apply()`, `var()` and `sd()`:

```

sigma2hat = apply(cerRetC, 2, var)
sigma2hat

##      MSFT      SBUX      SP500
## 0.010043 0.012465 0.002349

sigmahat = apply(cerRetC, 2, sd)
sigmahat

```

```
##      MSFT      SBUX      SP500
## 0.10021 0.11164 0.04847
```

Starbucks has the most variable monthly returns at 11%, and the S&P 500 index has the smallest at 5%.

The scatterplots of the returns are illustrated in Figure 7.3. All returns appear to be positively related. The covariance and correlation matrix estimates using (7.16) and (7.17) can be computed using the functions `var()` (or `cov()`) and `cor()`:

```
covmat = var(cerRetC)
covmat

##           MSFT      SBUX      SP500
## MSFT  0.010043 0.003814 0.002998
## SBUX  0.003814 0.012465 0.002476
## SP500 0.002998 0.002476 0.002349

cormat = cor(cerRetC)
cormat

##           MSFT      SBUX      SP500
## MSFT  1.0000 0.3409 0.6172
## SBUX  0.3409 1.0000 0.4575
## SP500 0.6172 0.4575 1.0000
```

To extract the unique pairwise values of σ_{ij} and ρ_{ij} from the matrix objects `covmat` and `cormat` use:

```
covhat = covmat[lower.tri(covmat)]
rhohat = cormat[lower.tri(cormat)]
names(covhat) <- names(rhohat) <- c("msft,sbux", "msft,sp500", "sbux,sp500")
covhat

## msft,sbux msft,sp500 sbux,sp500
## 0.003814 0.002998 0.002476

rhohat

## msft,sbux msft,sp500 sbux,sp500
## 0.3409 0.6172 0.4575
```

The pairs (MSFT, SP500) and (SBUX, SP500) are the most correlated. These estimates confirm the visual results from the scatterplot matrix in Figure 7.3.



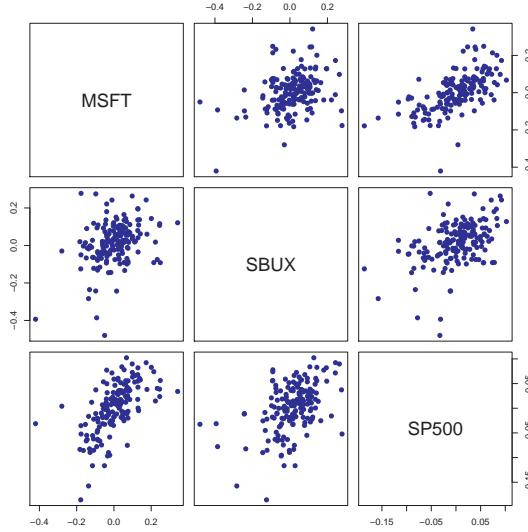


Fig. 7.3 Monthly cc returns on Microsoft stock, Starbucks stock, and the S&P 500 index, over the period January 1998 through May 2012.

7.3 Statistical Properties of the CER Model Estimates

To determine the statistical properties of plug-in principle estimators $\hat{\mu}_i, \hat{\sigma}_i^2, \hat{\sigma}_i, \hat{\sigma}_{ij}$ and $\hat{\rho}_{ij}$ in the CER model, we treat them as functions of the random variables $\{\mathbf{R}_t\}_{t=1}^T$ where \mathbf{R}_t is assumed to be generated by the CER model (7.1).

7.3.1 Bias

Assuming that returns are generated by the CER model (7.1), $\hat{\mu}_i$, $\hat{\sigma}_i^2$ and $\hat{\sigma}_{ij}$ are unbiased estimators,

$$\begin{aligned} E[\hat{\mu}_i] &= \mu_i \\ E[\hat{\sigma}_i^2] &= \sigma_i^2, \\ E[\hat{\sigma}_{ij}] &= \sigma_{ij}, \end{aligned}$$

but $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$ are biased estimators,

$$\begin{aligned} E[\hat{\sigma}_i] &\neq \sigma_i, \\ E[\hat{\rho}_{ij}] &\neq \rho_{ij}. \end{aligned}$$

It can be shown that the biases in $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$, are very small and decreasing in T such that $\text{bias}(\hat{\sigma}_i, \sigma_i) = \text{bias}(\hat{\rho}_{ij}, \rho_{ij}) = 0$ as $T \rightarrow \infty$. The proofs of these results are beyond the scope of this book and may be found, for example, in Goldberger (1991). As we shall see, these

results about bias can be easily verified using Monte Carlo methods.

It is instructive to illustrate how to derive the result $E[\hat{\mu}_i] = \mu_i$. Using results about the expectation of a linear combination of random variables, it follows that:

$$\begin{aligned} E[\hat{\mu}_i] &= E\left[\frac{1}{T} \sum_{t=1}^T R_{it}\right] \\ &= E\left[\frac{1}{T} \sum_{t=1}^T (\mu_i + \varepsilon_{it})\right] \quad (\text{since } R_{it} = \mu_i + \varepsilon_{it}) \\ &= \frac{1}{T} \sum_{t=1}^T \mu_i + \frac{1}{T} \sum_{t=1}^T E[\varepsilon_{it}] \quad (\text{by the linearity of } E[\cdot]) \\ &= \frac{1}{T} \sum_{t=1}^T \mu_i \quad (\text{since } E[\varepsilon_{it}] = 0, t = 1, \dots, T) \\ &= \frac{1}{T} T \cdot \mu_i = \mu_i. \end{aligned}$$

The derivation of the results $E[\hat{\sigma}_i^2] = \sigma_i^2$ and $E[\hat{\sigma}_{ij}] = \sigma_{ij}$ are similar but are considerably more involved and so are omitted.

7.3.2 Precision

Because the CER model estimators are either unbiased or the bias is very small, the precision of these estimators is measured by their standard errors. The standard error for $\hat{\mu}_i$, $\text{se}(\hat{\mu}_i)$, can be calculated exactly and is given by:

$$\text{se}(\hat{\mu}_i) = \frac{\sigma_i}{\sqrt{T}}. \quad (7.18)$$

The derivation of this result is straightforward. Using the results for the variance of a linear combination of uncorrelated random variables, we have:

$$\begin{aligned}
\text{var}(\hat{\mu}_i) &= \text{var}\left(\frac{1}{T} \sum_{t=1}^T R_{it}\right) \\
&= \text{var}\left(\frac{1}{T} \sum_{t=1}^T (\mu_i + \varepsilon_{it})\right) \quad (\text{since } R_{it} = \mu_i + \varepsilon_{it}) \\
&= \text{var}\left(\frac{1}{T} \sum_{t=1}^T \varepsilon_{it}\right) \quad (\text{since } \mu_i \text{ is a constant}) \\
&= \frac{1}{T^2} \sum_{t=1}^T \text{var}(\varepsilon_{it}) \quad (\text{since } \varepsilon_{it} \text{ is independent over time}) \\
&= \frac{1}{T^2} \sum_{t=1}^T \sigma_i^2 \quad (\text{since } \text{var}(\varepsilon_{it}) = \sigma_i^2, t = 1, \dots, T) \\
&= \frac{1}{T^2} T \sigma_i^2 = \frac{\sigma_i^2}{T}.
\end{aligned}$$

Then $\text{se}(\hat{\mu}_i) = \text{SD}(\hat{\mu}_i) = \frac{\sigma_i}{\sqrt{T}}$. We make the following remarks:

1. The value of $\text{se}(\hat{\mu}_i)$ is in the same units as $\hat{\mu}_i$ and measures the precision of $\hat{\mu}_i$ as an estimate. If $\text{se}(\hat{\mu}_i)$ is small relative to $\hat{\mu}_i$ then $\hat{\mu}_i$ is a relatively precise estimate of μ_i because $f(\hat{\mu}_i)$ will be tightly concentrated around μ_i ; if $\text{se}(\hat{\mu}_i)$ is large relative to $\hat{\mu}_i$ then $\hat{\mu}_i$ is a relatively imprecise estimate of μ_i because $f(\hat{\mu}_i)$ will be spread out about μ_i .
2. The magnitude of $\text{se}(\hat{\mu}_i)$ depends positively on the volatility of returns, $\sigma_i = \text{SD}(R_{it})$. For a given sample size T , assets with higher return volatility have larger values of $\text{se}(\hat{\mu}_i)$ than assets with lower return volatility. In other words, estimates of expected return for high volatility assets are less precise than estimates of expected returns for low volatility assets.
3. For a given return volatility σ_i , $\text{se}(\hat{\mu}_i)$ is smaller for larger sample sizes T . In other words, $\hat{\mu}_i$ is more precisely estimated for larger samples. Moreover, $\text{se}(\hat{\mu}_i) \rightarrow 0$ as $T \rightarrow \infty$ at rate \sqrt{T} .

The derivations of the standard errors for $\hat{\sigma}_i^2$, $\hat{\sigma}_i$, $\hat{\sigma}_{ij}$ and $\hat{\rho}_{ij}$ are complicated, and the exact results are extremely messy and hard to work with. However, there are simple approximate formulas for the standard errors of $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$ based on the CLT that are valid if the sample size, T , is reasonably large.² These large sample approximate formulas are given by:

$$\text{se}(\hat{\sigma}_i^2) \approx \frac{\sqrt{2}\sigma_i^2}{\sqrt{T}} = \frac{\sigma_i^2}{\sqrt{T/2}}, \quad (7.19)$$

$$\text{se}(\hat{\sigma}_i) \approx \frac{\sigma_i}{\sqrt{2T}}, \quad (7.20)$$

$$\text{se}(\hat{\rho}_{ij}) \approx \frac{(1 - \rho_{ij}^2)}{\sqrt{T}}, \quad (7.21)$$

where “ \approx ” denotes approximately equal. The approximations are such that the approximation error goes to zero as the sample size T gets very large. We make the following remarks:

² The large sample approximate formula for the variance of $\hat{\sigma}_{ij}$ is too messy to work with so we omit it here. In practice, we can use the bootstrap to provide an estimated standard error for $\hat{\sigma}_{ij}$.

1. As with the formula for the standard error of the sample mean, the formulas for $\text{se}(\hat{\sigma}_i^2)$ and $\text{se}(\hat{\sigma}_i)$ depend on σ_i^2 . Larger values of σ_i^2 imply less precise estimates of $\hat{\sigma}_i^2$ and $\hat{\sigma}_i$.
2. The formula for $\text{se}(\rho_{ij})$, does not depend on σ_i^2 but rather depends on ρ_{ij}^2 and is smaller the closer ρ_{ij}^2 is to unity. Intuitively, this makes sense because as ρ_{ij}^2 approaches one the linear dependence between R_{it} and R_{jt} becomes almost perfect and this will be easily recognizable in the data (scatterplot will almost follow a straight line).
3. The formulas for the standard errors above are inversely related to the square root of the sample size, \sqrt{T} , which means that larger sample sizes imply smaller values of the standard errors.
4. Interestingly, $\text{se}(\hat{\sigma}_i)$ goes to zero the fastest and $\text{se}(\hat{\sigma}_i^2)$ goes to zero the slowest. Hence, for a fixed sample size, these formulas suggest that σ_i is generally estimated more precisely than σ_i^2 and ρ_{ij} , and ρ_{ij} is estimated generally more precisely than σ_i^2 .

The above formulas (7.19) - (7.21) are not practically useful, however, because they depend on the unknown quantities σ_i^2 , σ_i and ρ_{ij} . Practically useful formulas replace σ_i^2 , σ_i and ρ_{ij} by the estimates $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$ and give rise to the *estimated standard errors*:

$$\widehat{\text{se}}(\hat{\mu}_i) = \frac{\hat{\sigma}_i}{\sqrt{T}} \quad (7.22)$$

$$\widehat{\text{se}}(\hat{\sigma}_i^2) \approx \frac{\hat{\sigma}_i^2}{\sqrt{T/2}}, \quad (7.23)$$

$$\widehat{\text{se}}(\hat{\sigma}_i) \approx \frac{\hat{\sigma}_i}{\sqrt{2T}}, \quad (7.24)$$

$$\widehat{\text{se}}(\rho_{ij}) \approx \frac{(1 - \hat{\rho}_{ij}^2)}{\sqrt{T}}. \quad (7.25)$$

It is good practice to report estimates together with their estimated standard errors. In this way the precision of the estimates is transparent to the user. Typically, estimates are reported in a table with the estimates in one column and the estimated standard errors in an adjacent column.

Example 7.3. $\widehat{\text{se}}(\hat{\mu}_i)$ values for Microsoft, Starbucks and the S&P 500 index.

For Microsoft, Starbucks and S&P 500, the values of $\widehat{\text{se}}(\hat{\mu}_i)$ are easily computed in R using:

```
n.obs = nrow(cerRetC)
seMuhat = sigmahat/sqrt(n.obs)
```

The values of $\hat{\mu}_i$ and $\widehat{\text{se}}(\hat{\mu}_i)$ shown together are:

```
cbind(muhat, seMuhat)

##          muhat    seMuhat
## MSFT  0.004126 0.007641
## SBUX  0.014657 0.008513
## SP500 0.001687 0.003696
```

For Microsoft and Starbucks, the values of $\widehat{\text{se}}(\hat{\mu}_i)$ are similar because the values of $\hat{\sigma}_i$ are similar, and $\widehat{\text{se}}(\hat{\mu}_i)$ is smallest for the S&P 500 index. This occurs because $\hat{\sigma}_{\text{SP500}}$ is much

smaller than $\hat{\sigma}_{\text{msft}}$ and $\hat{\sigma}_{\text{sbux}}$. Hence, $\hat{\mu}_i$ is estimated more precisely for the S&P 500 index (a highly diversified portfolio) than it is for Microsoft and Starbucks stock (individual assets).

It is tempting to compare the magnitude of $\hat{s.e.}(\hat{\mu}_i)$ to the value of $\hat{\mu}_i$ to evaluate if $\hat{\mu}_i$ is a precise estimate:

```
seMuhat/muhat

##   MSFT    SBUX   SP500
## 1.8522 0.5808 2.1905
```

Here we see that $\hat{s.e.}(\hat{\mu}_{\text{msft}})$ and $\hat{s.e.}(\hat{\mu}_{\text{sp500}})$ are about twice as large as $\hat{\mu}_{\text{msft}}$ and $\hat{\mu}_{\text{sp500}}$, respectively, whereas $\hat{s.e.}(\hat{\mu}_{\text{sbux}})$ is about half the size of $\hat{\mu}_{\text{sbux}}$. This seems to indicate that $\hat{\mu}_{\text{sbux}}$ is more precisely estimated than $\hat{\mu}_{\text{sp500}}$. However, this comparison is misleading. To see why, consider the range of values determined by $\hat{\mu}_i \pm 2 \times \hat{s.e.}(\hat{\mu}_i)$.

```
upper = muhat + 2 * seMuhat
lower = muhat - 2 * seMuhat
width = upper - lower
cbind(lower, upper, width)

##           lower      upper     width
## MSFT    -0.011157 0.019408 0.03056
## SBUX    -0.002369 0.031683 0.03405
## SP500   -0.005705 0.009079 0.01478
```

For normally distributed $\hat{\mu}_i$, this range contains the true value of μ_i with probability around 0.95. For all assets, this range contains both positive and negative values but the range is smallest for the S&P 500 index.

Example 7.4. Computing $\hat{s.e.}(\hat{\sigma}_i^2)$, $\hat{s.e.}(\hat{\sigma}_i)$, and $\hat{s.e.}(\hat{\rho}_{ij})$ for Microsoft, Starbucks and the S&P 500.

For Microsoft, Starbucks and S&P 500, the values of $\hat{s.e.}(\hat{\sigma}_i^2)$, $\hat{s.e.}(\hat{\sigma}_i)$ and $\hat{s.e.}(\hat{\rho}_{ij})$ (together with the estimates $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$) are:

```
seSigma2hat = sigma2hat/sqrt(n.obs/2)
seSigmahat = sigmahat/sqrt(2 * n.obs)
cbind(sigma2hat, seSigma2hat, sigmahat, seSigmahat)

##           sigma2hat      seSigma2hat      sigmahat      seSigmahat
## MSFT    0.010043    0.0010829    0.10021    0.005403
## SBUX    0.012465    0.0013441    0.11164    0.006019
## SP500   0.002349    0.0002533    0.04847    0.002613
```

Notice that σ^2 and σ for the S&P 500 index are estimated much more precisely than the values for Microsoft and Starbucks. Also notice that σ_i is estimated more precisely than μ_i for all assets: the values of $\hat{s.e.}(\hat{\sigma}_i)$ relative to $\hat{\sigma}_i$ are much smaller than the values of $\hat{s.e.}(\hat{\mu}_i)$ to $\hat{\mu}_i$.

The values of $\hat{s.e.}(\hat{\rho}_{ij})$ (together with $\hat{\rho}_{ij}$) are:

```

seRhohat = (1 - rhohat^2)/sqrt(n.obs)
cbind(rhohat, seRhohat)

##          rhohat    seRhohat
## msft,sbux 0.3409  0.06739
## msft,sp500 0.6172  0.04720
## sbux,sp500 0.4575  0.06029

```

The values of $\widehat{\text{se}}(\hat{\rho}_{ij})$ are moderate in size (relative to $\hat{\rho}_{ij}$). Notice that $\hat{\rho}_{\text{sbux,sp500}}$ has the smallest estimated standard error because $\hat{\rho}_{\text{sbux,sp500}}^2$ is closest to one.

■

7.3.3 Sampling Distributions and Confidence Intervals

Sampling Distribution for $\hat{\mu}_i$

In the CER model, $R_{it} \sim \text{iid } N(\mu_i, \sigma_i^2)$ and since $\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^T R_{it}$ is an average of these normal random variables, it is also normally distributed. The mean of $\hat{\mu}_i$ is μ_i and its variance is $\frac{\sigma_i^2}{T}$. Therefore, the exact probability distribution of $\hat{\mu}_i$, $f(\hat{\mu}_i)$, for a fixed sample size T is the normal distribution $\hat{\mu}_i \sim N\left(\mu_i, \frac{\sigma_i^2}{T}\right)$ where:

$$f(\hat{\mu}_i) = \left(\frac{2\pi\sigma_i^2}{T} \right)^{-1/2} \exp \left\{ -\frac{1}{2\sigma_i^2/T} (\hat{\mu}_i - \mu_i)^2 \right\}. \quad (7.26)$$

The probability curve $f(\hat{\mu}_i)$ is centered at the true value μ_i , and the spread about μ_i depends on the magnitude of σ_i^2 , the variability of R_{it} , and the sample size, T . For a fixed sample size, T , the uncertainty in $\hat{\mu}_i$ is larger for larger values of σ_i^2 . Notice that the variance of $\hat{\mu}_i$ is inversely related to the sample size T . Given σ_i^2 , $\text{var}(\hat{\mu}_i)$ is smaller for larger sample sizes than for smaller sample sizes. This makes sense since we expect to have a more precise estimator when we have more data. If the sample size is very large (as $T \rightarrow \infty$) then $\text{var}(\hat{\mu}_i)$ will be approximately zero and the normal distribution of $\hat{\mu}_i$ given by (7.26) will be essentially a spike at μ_i . In other words, if the sample size is very large then we essentially know the true value of μ_i . Hence, we have established that $\hat{\mu}_i$ is a *consistent* estimator of μ_i as the sample size goes to infinity.

Example 7.5. Sampling distribution of $\hat{\mu}$ with different sample sizes.

The distribution of $\hat{\mu}_i$, with $\mu_i = 0$ and $\sigma_i^2 = 1$ for various sample sizes is illustrated in figure 7.4. Notice how fast the distribution collapses at $\mu_i = 0$ as T increases.

■

Confidence intervals for μ_i

The precision of $\hat{\mu}_i$ is measured by $\widehat{\text{se}}(\hat{\mu}_i)$ but is best communicated by computing a *confidence interval* for the unknown value of μ_i . A confidence interval is an *interval estimate*

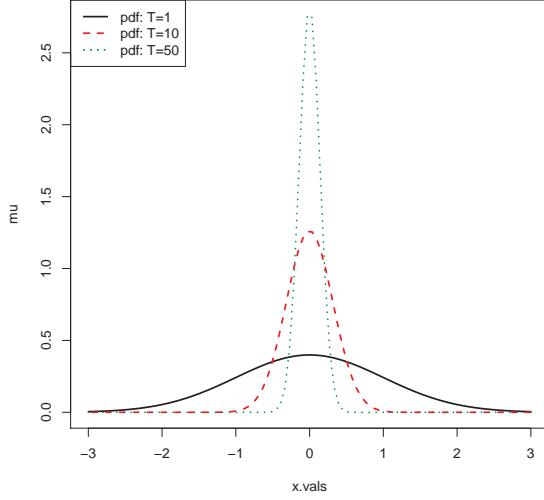


Fig. 7.4 $N(0, 1/T)$ sampling distributions for $\hat{\mu}$ for $T = 1, 10$ and 50 .

of μ_i such that we can put an explicit probability statement about the likelihood that the interval covers μ_i .

The construction of an exact confidence interval for μ_i is based on the following statistical result (see the appendix for details).

Result: Let $\{R_{it}\}_{t=1}^T$ be generated from the CER model (7.1). Define the t -ratio as:

$$t_i = \frac{\hat{\mu}_i - \mu_i}{\widehat{\text{se}}(\hat{\mu}_i)} = \frac{\hat{\mu}_i - \mu_i}{\hat{\sigma}/\sqrt{T}}, \quad (7.27)$$

Then $t_i \sim t_{T-1}$ where t_{T-1} denotes a Student's t random variable with $T - 1$ degrees of freedom.

The Student's t distribution with $v > 0$ degrees of freedom is a symmetric distribution centered at zero, like the standard normal. The tail-thickness (kurtosis) of the distribution is determined by the degrees of freedom parameter v . For values of v close to zero, the tails of the Student's t distribution are much fatter than the tails of the standard normal distribution. As v gets large, the Student's t distribution approaches the standard normal distribution.

For $\alpha \in (0, 1)$, we compute a $(1 - \alpha) \cdot 100\%$ confidence interval for μ_i using (7.27) and the $1 - \alpha/2$ quantile (critical value) $t_{T-1}(1 - \alpha/2)$ to give:

$$\Pr \left(-t_{T-1}(1 - \alpha/2) \leq \frac{\hat{\mu}_i - \mu_i}{\widehat{\text{se}}(\hat{\mu}_i)} \leq t_{T-1}(1 - \alpha/2) \right) = 1 - \alpha,$$

which can be rearranged as,

$$\Pr(\hat{\mu}_i - t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i) \leq \mu_i \leq \hat{\mu}_i + t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i)) = 1 - \alpha.$$

Hence, the interval,

$$\begin{aligned} & [\hat{\mu}_i - t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i), \hat{\mu}_i + t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i)] \\ &= \hat{\mu}_i \pm t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i) \end{aligned} \quad (7.28)$$

covers the true unknown value of μ_i with probability $1 - \alpha$.

Example 7.6. Computing 95% confidence intervals for μ_i

Suppose we want to compute a 95% confidence interval for μ_i . In this case $\alpha = 0.05$ and $1 - \alpha = 0.95$. Suppose further that $T - 1 = 60$ (e.g., five years of monthly return data) so that $t_{T-1}(1 - \alpha/2) = t_{60}(0.975) = 2$. This can be verified in R using the function `qt()`.

Then the 95% confidence for μ_i is given by:

$$\hat{\mu}_i \pm 2 \cdot \hat{se}(\hat{\mu}_i). \quad (7.29)$$

The above formula for a 95% confidence interval is often used as a rule of thumb for computing an approximate 95% confidence interval for moderate sample sizes. It is easy to remember and does not require the computation of the quantile $t_{T-1}(1 - \alpha/2)$ from the Student's t distribution. It is also an approximate 95% confidence interval that is based the asymptotic normality of $\hat{\mu}_i$. Recall, for a normal distribution with mean μ and variance σ^2 approximately 95% of the probability lies between $\mu \pm 2\sigma$. ■

The coverage probability associated with the confidence interval for μ_i is based on the fact that the estimator $\hat{\mu}_i$ is a random variable. Since confidence interval is constructed as $\hat{\mu}_i \pm t_{T-1}(1 - \alpha/2) \cdot \hat{se}(\hat{\mu}_i)$ it is also a random variable. An intuitive way to think about the coverage probability associated with the confidence interval is to think about the game of "horseshoes"³. The horseshoe is the confidence interval and the parameter μ_i is the post at which the horse shoe is tossed. Think of playing game 100 times (i.e. simulate 100 samples of the CER model). If the thrower is 95% accurate (if the coverage probability is 0.95) then 95 of the 100 tosses should ring the post (95 of the constructed confidence intervals should contain the true value μ_i).

Example 7.7. 95% confidence intervals for μ_i for Microsoft, Starbucks and the S& P 500 index.

Consider computing 95% confidence intervals for μ_i using (7.28) based on the estimated results for the Microsoft, Starbucks and S&P 500 data. The degrees of freedom for the Student's t distribution is $T - 1 = 171$. The 97.5% quantile, $t_{99}(0.975)$, can be computed using the R function `qt()`:

³ Horse shoes is a game commonly played at county fairs. See <http://en.wikipedia.org/wiki/Horseshoes> for a complete description of the game.

```
t.975 = qt(0.975, df = (n.obs - 1))
t.975

## [1] 1.974
```

Notice that this quantile is very close to 2. Then the exact 95% confidence intervals are given by:

```
lower = muhat - t.975 * seMuhat
upper = muhat + t.975 * seMuhat
width = upper - lower
cbind(lower, upper, width)

##          lower     upper   width
## MSFT    -0.010958 0.019209 0.03017
## SBUX    -0.002147 0.031461 0.03361
## SP500   -0.005608 0.008983 0.01459
```

With probability 0.95, the above intervals will contain the true mean values assuming the CER model is valid. The 95% confidence intervals for Microsoft and Starbucks are fairly wide (about 3%) and contain both negative and positive values. The confidence interval for the S&P 500 index is tighter but also contains negative and positive values. For Microsoft, the confidence interval is $[-1.1\%, 1.9\%]$. This means that with probability 0.95, the true monthly expected return is somewhere between -1.1% and 1.9%. The economic implications of a -1.1% expected monthly return and a 1.9% expected return are vastly different. In contrast, the 95% confidence interval for the SP500 is about half the width of the intervals for Microsoft or Starbucks. The lower limit is near -0.5% and the upper limit is near 1%. This result clearly shows that the monthly mean return for the S&P 500 index is estimated much more precisely than the monthly mean returns for Microsoft or Starbucks.

Sampling distributions for $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$

The exact distributions of $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ and $\hat{\rho}_{ij}$ based on a fixed sample size T are not normal distributions and are difficult to derive⁴. However, approximate normal distributions of the form (7.7) based on the CLT are readily available:

$$\hat{\sigma}_i^2 \sim N(\sigma_i^2, \text{se}(\hat{\sigma}_i^2)^2) = N\left(\sigma_i^2, \frac{4\sigma_i^4}{T}\right), \quad (7.30)$$

$$\hat{\sigma}_i \sim N(\sigma_i, \text{se}(\hat{\sigma}_i)^2) = N\left(\sigma_i, \frac{\sigma_i^2}{2T}\right), \quad (7.31)$$

$$\hat{\rho}_{ij} \sim N(\rho_{ij}, \text{se}(\hat{\rho}_{ij})^2) = N\left(\rho_{ij}, \frac{(1 - \hat{\rho}_{ij}^2)^2}{T}\right). \quad (7.32)$$

These approximate normal distributions can be used to compute approximate confidence intervals for σ_i^2 , σ_i and ρ_{ij} .

⁴ For example, the exact sampling distribution of $(T-1)\hat{\sigma}_i^2/\sigma_i^2$ is chi-square with $T-1$ degrees of freedom.

Approximate Confidence Intervals for σ_i^2 , σ_i and ρ_{ij}

Approximate 95% confidence intervals for σ_i^2 , σ_i and ρ_{ij} are given by:

$$\hat{\sigma}_i^2 \pm 2 \cdot \widehat{se}(\hat{\sigma}_i^2) = \hat{\sigma}_i^2 \pm 2 \cdot \frac{\hat{\sigma}_i^2}{\sqrt{T/2}}, \quad (7.33)$$

$$\hat{\sigma}_i \pm 2 \cdot \widehat{se}(\hat{\sigma}_i) = \hat{\sigma}_i \pm 2 \cdot \frac{\hat{\sigma}_i}{\sqrt{2T}}, \quad (7.34)$$

$$\hat{\rho}_{ij} \pm 2 \cdot \widehat{se}(\hat{\rho}_{ij}) = \hat{\rho}_{ij} \pm 2 \cdot \frac{(1 - \hat{\rho}_{ij}^2)}{\sqrt{T}}. \quad (7.35)$$

Example 7.8. Approximate 95% confidence intervals for σ_i^2 , σ_i and ρ_{ij} for Microsoft, Starbucks and the S&P 500.

Using (7.33) - (7.34), the approximate 95% confidence intervals for σ_i^2 and σ_i (i = Microsoft, Starbucks, S&P 500) are:

```
# 95% confidence interval for variance
lowerSigma2 = sigma2hat - 2 * seSigma2hat
upperSigma2 = sigma2hat + 2 * seSigma2hat
widthSigma2 = upperSigma2 - lowerSigma2
cbind(lowerSigma2, upperSigma2, widthSigma2)

##      lowerSigma2 upperSigma2 widthSigma2
## MSFT    0.007877   0.012208   0.004332
## SBUX    0.009776   0.015153   0.005376
## SP500   0.001843   0.002856   0.001013

# 95% confidence interval for volatility
lowerSigma = sigmahat - 2 * seSigmahat
upperSigma = sigmahat + 2 * seSigmahat
widthSigma = upperSigma - lowerSigma
cbind(lowerSigma, upperSigma, widthSigma)

##      lowerSigma upperSigma widthSigma
## MSFT    0.08941    0.1110    0.02161
## SBUX    0.09961    0.1237    0.02408
## SP500   0.04324    0.0537    0.01045
```

The 95% confidence intervals for σ and σ^2 are larger for Microsoft and Starbucks than for the S&P 500 index. For all assets, the intervals for σ are fairly narrow (2% for Microsoft and Starbucks and 1% for S&P 500 index) indicating that σ is precisely estimated.

The approximate 95% confidence intervals for ρ_{ij} are:

```
lowerRho = rhohat - 2 * seRhohat
upperRho = rhohat + 2 * seRhohat
widthRho = upperRho - lowerRho
cbind(lowerRho, upperRho, widthRho)

##      lowerRho upperRho widthRho
## msft,sbux 0.2061  0.4757  0.2696
```

```
## msft,sp500  0.5228  0.7116  0.1888
## sbux,sp500  0.3369  0.5780  0.2412
```

The 95% confidence intervals for ρ_{ij} are not too wide and all contain just positive values away from zero. The smallest interval is for $\rho_{\text{msft},\text{sp500}}$ because $\hat{\rho}_{\text{msft},\text{sp500}}$ is closest to 1. ■

7.4 Using Monte Carlo Simulation to Understand the Statistical Properties of Estimators

Let R_t be the return on a single asset described by the CER model, let θ denote some characteristic (parameter) of the CER model we are interested in estimating, and let $\hat{\theta}$ denote an estimator for θ based on a sample of size T . The exact meaning of estimator bias, $\text{bias}(\hat{\theta}, \theta)$, the interpretation of $\text{se}(\hat{\theta})$ as a measure of precision, the sampling distribution $f(\hat{\theta})$, and the interpretation of the coverage probability of a confidence interval for θ , can all be a bit hard to grasp at first. If $\text{bias}(\hat{\theta}, \theta) = 0$ so that $E[\hat{\theta}] = \theta$ then over an infinite number of repeated samples of $\{R_{it}\}_{t=1}^T$ the average of the $\hat{\theta}$ values computed over the infinite samples is equal to the true value θ . The value of $\text{se}(\hat{\theta})$ represents the standard deviation of these $\hat{\theta}$ values. The sampling distribution $f(\hat{\theta})$ is the smoothed histogram of these $\hat{\theta}$ values. And the 95% confidence intervals for θ will actually contain θ in 95% of the samples. We can think of these hypothetical samples as different Monte Carlo simulations of the CER model. In this way we can approximate the computations involved in evaluating $E[\hat{\theta}]$, $\text{se}(\hat{\theta})$, $f(\hat{\theta})$, and the coverage probability of a confidence interval using a large, but finite, number of Monte Carlo simulations.

7.4.1 Evaluating the Statistical Properties of $\hat{\mu}$ Using Monte Carlo Simulation

Consider the CER model:

$$\begin{aligned} R_t &= 0.05 + \varepsilon_t, t = 1, \dots, 100 \\ \varepsilon_t &\sim \text{GWN}(0, (0.10)^2). \end{aligned} \tag{7.36}$$

Here, the true parameter values are $\mu = 0.05$ and $\sigma = 0.10$. Using Monte Carlo simulation, we can simulate $N = 1000$ different samples of size $T = 100$ from (7.36) giving the sample realizations $\{r_t^j\}_{t=1}^{100}$ for $j = 1, \dots, 1000$. The first 10 of these simulated samples are illustrated in Figure 7.5. Notice that there is considerable variation in the appearance of the simulated samples, but that all of the simulated samples fluctuate about the true mean value of $\mu = 0.05$ and have a typical deviation from the mean of about 0.10. For each of the 1000 simulated samples we can estimate $\hat{\mu}$ giving 1000 mean estimates $\{\hat{\mu}^1, \dots, \hat{\mu}^{1000}\}$. A histogram of these 1000 mean values is illustrated in Figure ???. The histogram of the estimated means, $\hat{\mu}^j$, can be thought of as an estimate of the underlying pdf, $f(\hat{\mu})$, of the

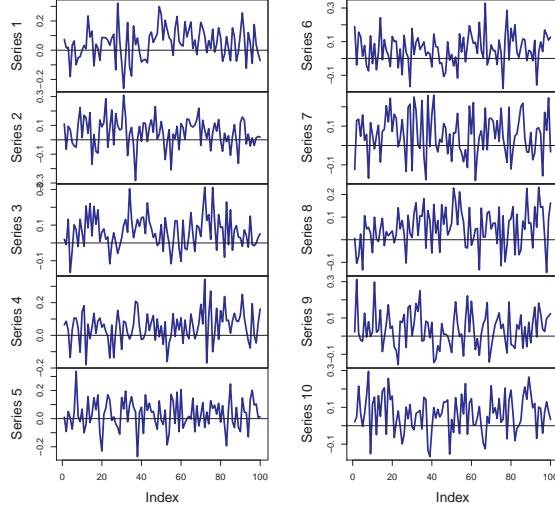


Fig. 7.5 Ten simulated samples of size $T = 100$ from the CER model $R_t = 0.05 + \varepsilon_t$, $\varepsilon_t \sim \text{iid } N(0, (0.10)^2)$.

estimator $\hat{\mu}$ which we know from (7.26) is a normal pdf centered at $E[\hat{\mu}] = \mu = 0.05$ with $\text{se}(\hat{\mu}_i) = 0.10/\sqrt{100} = 0.01$. This normal curve (solid orange line) is superimposed on the histogram in Figure 7.6. Notice that the center of the histogram (white dashed vertical line) is very close to the true mean value $\mu = 0.05$. That is, on average over the 1000 Monte Carlo samples the value of $\hat{\mu}$ is about 0.05. In some samples, the estimate is too big and in some samples the estimate is too small but on average the estimate is correct. In fact, the average value of $\{\hat{\mu}^1, \dots, \hat{\mu}^{1000}\}$ from the 1000 simulated samples is:

$$\bar{\hat{\mu}} = \frac{1}{1000} \sum_{j=1}^{1000} \hat{\mu}^j = 0.0497,$$

which is very close to the true value 0.05. If the number of simulated samples is allowed to go to infinity then the sample average $\bar{\hat{\mu}}$ will be exactly equal to $\mu = 0.05$:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N \hat{\mu}^j = E[\hat{\mu}] = \mu = 0.05.$$

The typical size of the spread about the center of the histogram represents $\text{se}(\hat{\mu}_i)$ and gives an indication of the precision of $\hat{\mu}_i$. The value of $\text{se}(\hat{\mu}_i)$ may be approximated by computing the sample standard deviation of the 1000 $\hat{\mu}^j$ values:

$$\hat{\sigma}_{\hat{\mu}} = \sqrt{\frac{1}{999} \sum_{j=1}^{1000} (\hat{\mu}^j - 0.04969)^2} = 0.0104.$$

Notice that this value is very close to $\text{se}(\hat{\mu}_i) = 0.10/\sqrt{100} = 0.01$. If the number of simulated sample goes to infinity then:

$$\lim_{N \rightarrow \infty} \sqrt{\frac{1}{N-1} \sum_{j=1}^N (\hat{\mu}^j - \bar{\hat{\mu}})^2} = \text{se}(\hat{\mu}_i) = 0.10.$$

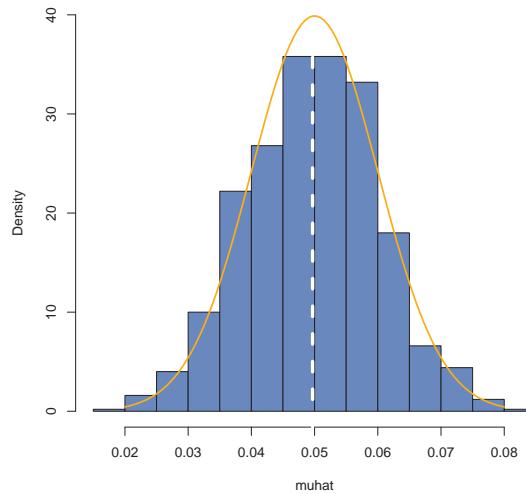


Fig. 7.6 Distribution of $\hat{\mu}$ computed from 1000 Monte Carlo simulations from the CER model (7.36). White dashed line is the average of the μ values, and orange curve is the true $f(\hat{\mu})$.

The coverage probability of the 95% confidence interval for μ can also be illustrated using Monte Carlo simulation. For each simulation j , the interval $\hat{\mu} \pm t_{100}(0.975) \times \widehat{\text{se}}(\hat{\mu}^j)$ is computed. The coverage probability is approximated by the fraction of intervals that contain (cover) the true $\mu = 0.05$. For the 1000 simulated samples, this fraction turns out to be 0.931. As the number of simulations goes to infinity, the Monte Carlo coverage probability will be equal to 0.95.

Example 7.9. Monte Carlo simulation to evaluate $E[\hat{\mu}]$, $\text{se}(\hat{\mu})$ and 95% confidence intervals for μ .

The R code to perform the Monte Carlo simulation presented in this section is:

```
mu = 0.05
sigma = 0.1
n.obs = 100
n.sim = 1000
set.seed(111)
sim.means = rep(0, n.sim)
mu.lower = rep(0, n.sim)
mu.upper = rep(0, n.sim)
qt.975 = qt(0.975, n.obs - 1)
```

```

for (sim in 1:n.sim) {
  sim.ret = rnorm(n.obs, mean = mu, sd = sigma)
  sim.means[sim] = mean(sim.ret)
  se.muhat = sd(sim.ret)/sqrt(n.obs)
  mu.lower[sim] = sim.means[sim] - qt.975 * se.muhat
  mu.upper[sim] = sim.means[sim] + qt.975 * se.muhat
}

```

The 1000×1 vectors `sim.means`, `mu.lower` and `mu.upper` contain the values of $\hat{\mu}^j$, $\hat{\mu}^j - t_{100}(0.975) \times \hat{s.e}(\hat{\mu}^j)$ and $\hat{\mu}^j + t_{100}(0.975) \times \hat{s.e}(\hat{\mu}^j)$ computed from each of the simulated samples $j = 1, \dots, 1000$. The mean and standard deviation of $\{\hat{\mu}^1, \dots, \hat{\mu}^{1000}\}$ are:

```

mean(sim.means)

## [1] 0.04969

sd(sim.means)

## [1] 0.01041

```

To evaluate the coverage probability of the 95% confidence intervals, we count the number of times each interval actually contains the true value of μ :

```

in.interval = mu >= mu.lower & mu <= mu.upper
sum(in.interval)/n.sim

## [1] 0.931

```

■

7.4.2 Evaluating the Statistical Properties of $\hat{\sigma}_i^2$ and $\hat{\sigma}_i$ Using Monte Carlo simulation

We can evaluate the statistical properties of $\hat{\sigma}_i^2$ and $\hat{\sigma}_i$ by Monte Carlo simulation in the same way that we evaluated the statistical properties of $\hat{\mu}_i$. We use the simulation model (7.36) and $N = 1000$ simulated samples of size $T = 100$ to compute the estimates $\{(\hat{\sigma}^2)^1, \dots, (\hat{\sigma}^2)^{1000}\}$ and $\{\hat{\sigma}^1, \dots, \hat{\sigma}^{1000}\}$. The R code for the simulation loop is

```

mu = 0.05
sigma = 0.1
n.obs = 100
n.sim = 1000
set.seed(111)
sim.sigma2 = rep(0, n.sim)
sigma2.lower = rep(0, n.sim)
sigma2.upper = rep(0, n.sim)
sim.sigma = rep(0, n.sim)
sigma.lower = rep(0, n.sim)
sigma.upper = rep(0, n.sim)
for (sim in 1:n.sim) {

```

```

sim.ret = rnorm(n.obs, mean = mu, sd = sigma)
sim.sigma2[sim] = var(sim.ret)
sim.sigma[sim] = sqrt(sim.sigma2[sim])
sigma2.lower[sim] = sim.sigma2[sim] - 2 * sim.sigma2[sim]/sqrt(n.obs/2)
sigma2.upper[sim] = sim.sigma2[sim] + 2 * sim.sigma2[sim]/sqrt(n.obs/2)
sigma.lower[sim] = sim.sigma[sim] - 2 * sim.sigma[sim]/sqrt(n.obs * 2)
sigma.upper[sim] = sim.sigma[sim] + 2 * sim.sigma[sim]/sqrt(n.obs * 2)
}
# compute expected value of estimates and bias
mean(sim.sigma2)

## [1] 0.00999

mean(sim.sigma2) - sigma^2

## [1] -9.865e-06

mean(sim.sigma)

## [1] 0.09972

mean(sim.sigma) - sigma

## [1] -0.0002782

# compute MC and asymptotic SE values
sd(sim.sigma2)

## [1] 0.001352

sigma^2/sqrt(n.obs/2)

## [1] 0.001414

sd(sim.sigma)

## [1] 0.006764

sigma/sqrt(2 * n.obs)

## [1] 0.007071

```

The histograms of these values, with the asymptotic normal distributions overlayed, are displayed in Figure 7.7. The histogram for the $\hat{\sigma}^2$ values is bell-shaped and slightly right skewed but is centered very close to $\sigma^2 = 0.010$. The histogram for the $\hat{\sigma}$ values is more symmetric and is centered near $\sigma = 0.10$. The average values of $\hat{\sigma}^2$ and $\hat{\sigma}$ from the 1000 simulations are:

$$\overline{\hat{\sigma}^2} = \frac{1}{1000} \sum_{j=1}^{1000} (\hat{\sigma}^2)^j = 0.00999,$$

$$\overline{\hat{\sigma}} = \frac{1}{1000} \sum_{j=1}^{1000} \hat{\sigma}^j = 0.0997.$$

The Monte Carlo estimate of the bias for $\hat{\sigma}^2$ is $0.00999 - 0.01 = -0.0000$, and the estimate of bias for $\hat{\sigma}$ is $0.0997 - 0.010 = -0.0003$. This confirms that $\hat{\sigma}^2$ is unbiased and that the bias in $\hat{\sigma}$ is extremely small. If the number of simulated samples, N , goes to infinity then $\overline{\hat{\sigma}^2} \rightarrow E[\hat{\sigma}^2] = \sigma^2 = 0.01$, and $\overline{\hat{\sigma}} \rightarrow E[\hat{\sigma}] = \sigma + \text{bias}(\hat{\sigma}, \sigma)$.

The sample standard deviation values of the Monte Carlo estimates of σ^2 and σ give approximations to $\text{se}(\hat{\sigma}^2)$ and $\text{se}(\hat{\sigma})$:

$$\hat{\sigma}_{\hat{\sigma}^2} = \sqrt{\frac{1}{999} \sum_{j=1}^{1000} ((\hat{\sigma}^2)^j - 0.00999)^2} = 0.00135,$$

$$\hat{\sigma}_{\hat{\sigma}} = \sqrt{\frac{1}{999} \sum_{j=1}^{1000} (\hat{\sigma}^j - 0.0997)^2} = 0.00676.$$

The approximate values for $\text{se}(\hat{\sigma}^2)$ and $\text{se}(\hat{\sigma})$ based on the CLT are:

$$\text{se}(\hat{\sigma}^2) = \frac{(0.10)^2}{\sqrt{100/2}} = 0.00141,$$

$$\text{se}(\hat{\sigma}^2) = \frac{0.10}{\sqrt{2 \times 100}} = 0.00707.$$

Notice that the Monte Carlo estimates of $\text{se}(\hat{\sigma}^2)$ and $\text{se}(\hat{\sigma})$ are a bit different from the CLT based estimates. The reason is that the CLT based estimates are approximations that hold when the sample size T is large. Because $T = 100$ is not too large, the Monte Carlo estimates of $\text{se}(\hat{\sigma}^2)$ and $\text{se}(\hat{\sigma})$ are likely more accurate (and will be more accurate if the number of simulations is larger).

For each simulation j , the approximate 95% confidence intervals $(\hat{\sigma}^2)^j \pm 2 \times \widehat{\text{se}}((\hat{\sigma}^2)^j)$ and $\hat{\sigma}^j \pm 2 \times \widehat{\text{se}}(\hat{\sigma}^j)$ are computed.

```
in.interval = sigma^2 >= sigma2.lower & sigma^2 <= sigma2.upper
sum(in.interval)/n.sim

## [1] 0.951

in.interval = sigma >= sigma.lower & sigma <= sigma.upper
sum(in.interval)/n.sim

## [1] 0.963
```

The coverage probabilities of these intervals is approximated by the fractions of intervals that contain (cover) the true values $\sigma^2 = 0.01$ and $\sigma = 0.10$ respectively. For the 1000 simulated samples, these fractions turn out to be 0.951 and 0.963, respectively. As the number of simulations and the sample size goes to infinity, the Monte Carlo coverage probability will be equal to 0.95.

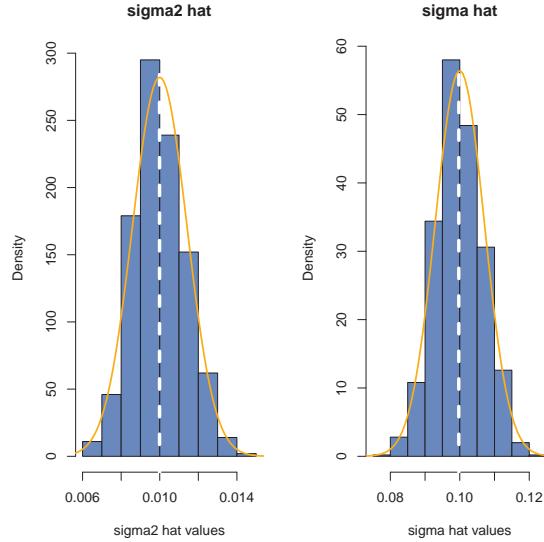


Fig. 7.7 Histograms of $\hat{\sigma}^2$ and $\hat{\sigma}$ computed from $N = 1000$ Monte Carlo samples from CER model.

7.4.3 Evaluating the Statistical Properties of $\hat{\rho}_{ij}$ by Monte Carlo simulation

To evaluate the statistical properties of $\hat{\rho}_{ij} = \text{cor}(R_{it}, R_{jt})$, we must simulate from the CER model in matrix form (7.1). For example, consider the bivariate CER model:

$$\begin{pmatrix} R_{1t} \\ R_{2t} \end{pmatrix} = \begin{pmatrix} 0.05 \\ 0.03 \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}, \quad t = 1, \dots, 100, \quad (7.37)$$

$$\begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \sim iid N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} (0.10)^2 & (0.75)(0.10)(0.05) \\ (0.75)(0.10)(0.05) & (0.05)^2 \end{pmatrix} \right), \quad (7.38)$$

where $\mu_1 = 0.05$, $\mu_2 = 0.03$, $\sigma_1 = 0.10$, $\sigma_2 = 0.05$, and $\rho_{12} = 0.75$. We use the simulation model (7.37) - (7.38) with $N = 1000$ simulated samples of size $T = 100$ to compute the estimates $\{\hat{\rho}_{12}^1, \dots, \hat{\rho}_{12}^{1000}\}$.

```
library(mvtnorm)
n.obs = 100
n.sim = 1000
set.seed(111)
sim.corrs = rep(0, n.sim) # initialize vectors
muvec = c(0.05, 0.03)
sigmavec = c(0.1, 0.05)
names(muvec) = names(sigmavec) = c("Asset.1", "Asset.2")
rho12 = 0.75
Sigma = diag(sigmavec^2)
Sigma[1, 2] = Sigma[2, 1] = rho12 * sigmavec[1] * sigmavec[2]
```

```

rho.lower = rep(0, n.sim)
rho.upper = rep(0, n.sim)
for (sim in 1:n.sim) {
  sim.ret = rmvnorm(n.obs, mean = muvec, sigma = Sigma)
  sim.corrs[sim] = cor(sim.ret)[1, 2]
  se.rhohat = (1 - sim.corrs[sim]^2)/sqrt(n.obs)
  rho.lower[sim] = sim.corrs[sim] - 2 * se.rhohat
  rho.upper[sim] = sim.corrs[sim] + 2 * se.rhohat
}
mean(sim.corrs)

## [1] 0.7469

mean(sim.corrs) - rho12

## [1] -0.003147

sd(sim.corrs)

## [1] 0.04504

```

The histogram of these values, with the asymptotic normal distribution overlaid, is displayed in Figure 7.8. The histogram for the $\hat{\rho}_{12}$ values is bell-shaped with a mild left skewness and centered close to $\rho_{12} = 0.75$. The sample mean and standard deviation values of $\hat{\rho}_{12}$ across the 1000 simulations are, respectively,

$$\bar{\hat{\rho}}_{12} = \frac{1}{1000} \sum_{j=1}^{1000} \hat{\rho}_{12}^j = 0.747,$$

$$\hat{\sigma}_{\hat{\rho}_{12}} = \sqrt{\frac{1}{999} \sum_{j=1}^{1000} (\hat{\rho}_{12}^j - 0.747)^2} = 0.045.$$

There is very slight downward bias in $\hat{\rho}_{12}$. The Monte Carlo standard deviation, $\hat{\sigma}_{\hat{\rho}_{12}}$, is very close to the approximate standard error $\text{se}(\hat{\rho}_{12}) = (1 - 0.75^2)/\sqrt{100} = 0.044$. For each simulation j , the approximate 95% confidence interval $\hat{\rho}_{12}^j \pm 2 \times \hat{\text{se}}(\hat{\rho}_{12}^j)$ is computed.

```

in.interval = (rho12 >= rho.lower) & (rho12 <= rho.upper)
sum(in.interval)/n.sim

## [1] 0.952

```

The coverage probability of this interval is approximated by the fractions of intervals that contain (cover) the true values $\rho_{12} = 0.75$. For the 1000 simulated samples, this fraction turns out to be 0.952.

7.5 Estimating Value-at-Risk in the CER Model

Consider the CER model for the simple return R_t . From the location-scale representation:

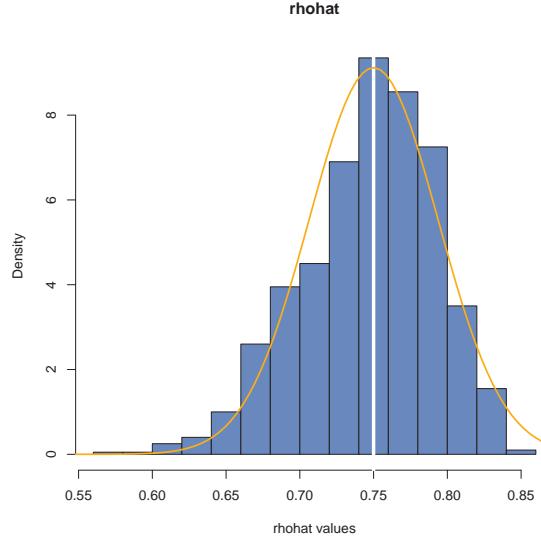


Fig. 7.8 Histograms of $\hat{\rho}_{12}$ computed from $N = 1000$ Monte Carlo samples from the bivariate CER model (7.37) - (7.38).

$$R_t = \mu + \sigma \times Z_t,$$

the α -quantile of R_t is,

$$q_\alpha^R = \mu + \sigma \times q_\alpha^Z,$$

where q_α^Z is the α -quantile of $Z_t \sim iid N(0, 1)$. Then, for an initial investment W_0 and $\alpha \in (0, 1)$ the $\alpha \cdot 100\%$ value-at-risk (VaR_α) is given by:

$$VaR_\alpha = W_0 \times q_\alpha^R = W_0(\mu + \sigma \times q_\alpha^Z).$$

Given the CER model estimators $\hat{\mu}$ and $\hat{\sigma}$, the plug-in principle estimator for VaR_α is:

$$\widehat{VaR}_\alpha = W_0 \times \hat{q}_\alpha^R = W_0(\hat{\mu} + \hat{\sigma} \times q_\alpha^Z). \quad (7.39)$$

Here \widehat{VaR}_α is a linear function of \hat{q}_α^R , which itself is a linear function of $\hat{\mu}$ and $\hat{\sigma}$. The statistical properties of \widehat{VaR}_α can then be easily derived from the statistical properties of $\hat{\mu}$ and $\hat{\sigma}$.

If the CER model is applied to the continuously compounded return R_t then:

$$VaR_\alpha = W_0 \times \left(e^{q_\alpha^R} - 1 \right) = W_0 \left(e^{\mu + \sigma \times q_\alpha^Z} - 1 \right),$$

and the estimate of VaR_α is:

$$\widehat{VaR}_\alpha = W_0 \times \left(e^{\hat{q}_\alpha^R} - 1 \right) = W_0 \left(e^{\hat{\mu} + \hat{\sigma} \times q_\alpha^Z} - 1 \right). \quad (7.40)$$

Here, \widehat{VaR}_α is a nonlinear function of \hat{q}_α^R (and $\hat{\mu}$ and $\hat{\sigma}$). In this case, the statistical properties of \widehat{VaR}_α cannot be easily derived from the statistical properties of $\hat{\mu}$ and $\hat{\sigma}$.

7.5.1 Statistical Properties of \hat{q}_α^R for Simple Returns

Regarding bias we have:

$$E[\hat{q}_\alpha^R] = E[\mu] + q_\alpha^Z \times E[\hat{\sigma}] \approx \mu + \sigma \times q_\alpha^Z. \quad (7.41)$$

Hence, \hat{q}_α^R is approximately unbiased for q_α^R because $\hat{\mu}$ is unbiased for μ and $\hat{\sigma}$ is approximately unbiased for σ .

To derive results regarding precision and the sampling distribution we require the following result.

Result. In the CER model:

$$\begin{pmatrix} \hat{\mu} \\ \hat{\sigma} \end{pmatrix} \sim N \left(\begin{pmatrix} \mu \\ \sigma \end{pmatrix}, \begin{pmatrix} \text{se}(\hat{\mu})^2 & 0 \\ 0 & \text{se}(\hat{\sigma})^2 \end{pmatrix} \right) \quad (7.42)$$

for large enough T . Hence, $\hat{\mu}$ and $\hat{\sigma}$ are (asymptotically) jointly normally distributed and $\text{cov}(\hat{\mu}, \hat{\sigma}) = 0$ which implies that they are also independent.

Using (7.42) we have:

$$\begin{aligned} \text{var}(\hat{q}_\alpha^R) &= \text{var}(\hat{\mu} + \hat{\sigma} \times q_\alpha^Z) \\ &= \text{var}(\hat{\mu}) + (q_\alpha^Z)^2 \text{var}(\hat{\sigma}) + 2q_\alpha^Z \text{cov}(\hat{\mu}, \hat{\sigma}) \\ &= \text{var}(\hat{\mu}) + (q_\alpha^Z)^2 \text{var}(\hat{\sigma}) \quad (\text{since } \text{cov}(\hat{\mu}, \hat{\sigma}) = 0) \\ &= \frac{\sigma^2}{T} + \frac{(q_\alpha^Z)^2 \sigma^2}{2T} \\ &= \frac{\sigma^2}{T} \left[1 + \frac{1}{2} (q_\alpha^Z)^2 \right]. \end{aligned}$$

Then,

$$\text{se}(\hat{q}_\alpha^R) = \sqrt{\text{var}(\hat{q}_\alpha^R)} = \frac{\sigma}{\sqrt{T}} \left[1 + \frac{1}{2} (q_\alpha^Z)^2 \right]^{1/2}. \quad (7.43)$$

Using the above results, the sampling distribution of \hat{q}_α^R can be approximated by the normal distribution:

$$\hat{q}_\alpha^R \sim N(q_\alpha^R, \text{se}(\hat{q}_\alpha^R)^2), \quad (7.44)$$

for large enough T .

Remarks,

1. $\text{se}(\hat{q}_\alpha^R)$ increases with σ and q_α^Z , and decreases with T . In particular, $\text{se}(\hat{q}_\alpha^R) \rightarrow \infty$ as $\alpha \rightarrow 0$ because $q_\alpha^Z \rightarrow -\infty$ as $\alpha \rightarrow 0$. Hence, we have very large estimation error in \hat{q}_α^R when α is very close to zero (for fixed values of σ and T).

2. The formula for $\text{se}(\hat{q}_\alpha^R)$ is not practically useful because it depends on the unknown value σ . The practically useful estimated standard error replaces the unknown value of σ with the estimate $\hat{\sigma}$ and is given by:

$$\widehat{\text{se}}(\hat{q}_\alpha^R) = \frac{\hat{\sigma}}{\sqrt{T}} \left[1 + \frac{1}{2} (q_\alpha^Z)^2 \right]^{1/2}. \quad (7.45)$$

Example 7.10. Estimating \hat{q}_α^R for Microsoft, Starbucks and the S&P 500 index.

The estimates of q_α^R and $\text{se}(\hat{q}_\alpha^R)$, for $\alpha = 0.05, 0.01$ and 0.001 , from the simple monthly returns for Microsoft, Starbucks and the S&P 500 index are:

```
n.obs = length(msftRetC)
muhatS = colMeans(cerRetS)
sigmahatS = apply(cerRetS, 2, sd)
qhat.05 = muhatS + sigmahatS * qnorm(0.05)
qhat.01 = muhatS + sigmahatS * qnorm(0.01)
qhat.001 = muhatS + sigmahatS * qnorm(0.001)
seQhat.05 = (sigmahatS/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.05)^2)
seQhat.01 = (sigmahatS/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.01)^2)
seQhat.001 = (sigmahatS/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.001)^2)
cbind(qhat.05, seQhat.05, qhat.01, seQhat.01, qhat.001, seQhat.001)

##          qhat.05 seQhat.05 qhat.01 seQhat.01 qhat.001 seQhat.001
## MSFT    -0.15780   0.01187 -0.2270  0.014898 -0.3045   0.018598
## SBUX    -0.15865   0.01276 -0.2330  0.016020 -0.3164   0.019997
## SP500   -0.07577   0.00559 -0.1083  0.007015 -0.1448   0.008757
```

For Microsoft and Starbucks, the values of $\widehat{\text{se}}(\hat{q}_\alpha^R)$ are about 1.2%, 1.5%, and 1.9% for $\alpha = 0.05$, $\alpha = 0.01$ and $\alpha = 0.001$, respectively. For the S&P 500 index, the corresponding values of $\widehat{\text{se}}(\hat{q}_\alpha^R)$ are 0.5%, 0.7%, and .9%, respectively. To see that these standard errors values are actually fairly large consider the 95% confidence intervals for $q_{0.05}^R$, $q_{0.01}^R$ and $q_{0.001}^R$:

```
# 95% CI for 5% quantile
lowerQhat.05 = qhat.05 - 2 * seQhat.05
upperQhat.05 = qhat.05 + 2 * seQhat.05
widthQhat.05 = upperQhat.05 - lowerQhat.05
cbind(lowerQhat.05, upperQhat.05, widthQhat.05)

##          lowerQhat.05 upperQhat.05 widthQhat.05
## MSFT     -0.18154    -0.13405     0.04748
## SBUX     -0.18418    -0.13312     0.05106
## SP500    -0.08695    -0.06459     0.02236

# 95% CI for 1% quantile
lowerQhat.01 = qhat.01 - 2 * seQhat.01
upperQhat.01 = qhat.01 + 2 * seQhat.01
widthQhat.01 = upperQhat.01 - lowerQhat.01
cbind(lowerQhat.01, upperQhat.01, widthQhat.01)

##          lowerQhat.01 upperQhat.01 widthQhat.01
## MSFT     -0.2568     -0.19717     0.05959
## SBUX     -0.2651     -0.20099     0.06408
## SP500    -0.1224     -0.09431     0.02806
```

```
# 95% CI for .1% quantile
lowerQhat.001 = qhat.001 - 2 * seQhat.001
upperQhat.001 = qhat.001 + 2 * seQhat.001
widthQhat.001 = upperQhat.001 - lowerQhat.001
cbind(lowerQhat.001, upperQhat.001, widthQhat.001)

##      lowerQhat.001 upperQhat.001 widthQhat.001
## MSFT      -0.3417      -0.2673     0.07439
## SBUX      -0.3564      -0.2764     0.07999
## SP500     -0.1624      -0.1273     0.03503
```

For example, the 95% confidence intervals for $q_{0.05}^R$, $q_{0.01}^R$ and $q_{0.001}^R$ for Microsoft are $[-19.7\%, -11.8\%]$, $[-27.7\%, -17.7\%]$, and $[-34.2\%, -26.7\%]$, respectively, which are quite large. Hence, the 5% and 1% simple monthly return quantiles have moderate estimation error, and the .1% quantile has very large estimation error. ■

7.5.2 Statistical Properties of $\widehat{\text{VaR}}_\alpha$ for Simple Returns

First, consider estimated $\widehat{\text{VaR}}_\alpha$ for simple returns (7.39). Here, we see that estimation error in $\widehat{\text{VaR}}_\alpha$ is directly linked to estimation error in \widehat{q}_α^R . Using (7.41) - (7.44) we have:

$$\begin{aligned} E[\widehat{\text{VaR}}_\alpha] &= W_0 \times E[\widehat{q}_\alpha^R] \approx W_0 \times q_\alpha^R = \text{VaR}_\alpha, \\ \text{se}(\widehat{\text{VaR}}_\alpha) &= W_0 \times \text{se}(\widehat{q}_\alpha^R), \end{aligned}$$

which suggests the following approximate normal distribution for $\widehat{\text{VaR}}_\alpha$:

$$\widehat{\text{VaR}}_\alpha \sim N(\text{VaR}_\alpha, \text{se}(\widehat{\text{VaR}}_\alpha)^2). \quad (7.46)$$

Hence, $\widehat{\text{VaR}}_\alpha$ is approximately unbiased and normally distributed (for large enough T) with:

$$\text{se}(\widehat{\text{VaR}}_\alpha) = W_0 \times \frac{\sigma}{\sqrt{T}} \left[1 + \frac{1}{2} (q_\alpha^Z)^2 \right]^{1/2}.$$

Remarks,

- $\text{se}(\widehat{\text{VaR}}_\alpha)$ increases with W_0 , σ and q_α^Z , and decreases with T .
- The practically useful estimated standard error replaces the unknown value of σ with the estimate $\hat{\sigma}$ and is given by:

$$\text{se}(\widehat{\text{VaR}}_\alpha) = W_0 \times \frac{\hat{\sigma}}{\sqrt{T}} \left[1 + \frac{1}{2} (q_\alpha^Z)^2 \right]^{1/2}. \quad (7.47)$$

Example 7.11. Estimating VaR_α for Microsoft, Starbucks and the S&P 500 index.

Consider a \$100,000 investment for one month in Microsoft, Starbucks and the S&P 500 index. The estimates of VaR_α and $\text{se}(\widehat{\text{VaR}}_\alpha)$, for $\alpha = 0.05$ and $\alpha = 0.01$, from the simple monthly returns for Microsoft, Starbucks and the S&P 500 index are:

```
w0 = 1e+05
VaR.05 = qhat.05 * w0
seVaR.05 = w0 * seQhat.05
VaR.01 = qhat.01 * w0
seVaR.01 = w0 * seQhat.01
cbind(VaR.05, seVaR.05, VaR.01, seVaR.01)

##           VaR.05 seVaR.05 VaR.01 seVaR.01
## MSFT     -15780      1187 -22696   1489.8
## SBUX     -15865      1276 -23303   1602.0
## SP500    -7577       559 -10834   701.5
```

The 95% confidence intervals for $\text{VaR}_{0.05}$ and $\text{VaR}_{0.01}$ are:

```
# 95% CI for 5% normal VaR
lowerVaR.05 = VaR.05 - 2 * seVaR.05
upperVaR.05 = VaR.05 + 2 * seVaR.05
widthVaR.05 = upperVaR.05 - lowerVaR.05
cbind(lowerVaR.05, upperVaR.05, widthVaR.05)

##           lowerVaR.05 upperVaR.05 widthVaR.05
## MSFT      -18154      -13405      4748
## SBUX      -18418      -13312      5106
## SP500     -8695       -6459      2236

# 95% CI for 1% VaR
lowerVaR.01 = VaR.01 - 2 * seVaR.01
upperVaR.01 = VaR.01 + 2 * seVaR.01
widthVaR.01 = upperVaR.01 - lowerVaR.01
cbind(lowerVaR.01, upperVaR.01, widthVaR.01)

##           lowerVaR.01 upperVaR.01 widthVaR.01
## MSFT      -25676      -19717      5959
## SBUX      -26507      -20099      6408
## SP500     -12237      -9431      2806
```

7.5.3 Using Monte Carlo Simulation to Understand the Statistical Properties of \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ for Simple Returns

We can evaluate the statistical properties of \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ by Monte Carlo simulation in the same way that we evaluated the statistical properties of $\hat{\mu}_i$. We use the simulation model (7.36) and $N = 1000$ simulated samples of size $T = 100$ to compute the estimates $\{(\hat{q}_\alpha^R)^1, \dots, (\hat{q}_\alpha^R)^{1000}\}$ and $\{(\widehat{\text{VaR}}_\alpha)^1, \dots, (\widehat{\text{VaR}}_\alpha)^{1000}\}$ for $\alpha = 0.05, 0.01$ and $W_0 = \$100,000$. The true values for q_α^R are:

$$\begin{aligned} q_{.05}^R &= 0.05 + (0.10) \times (-1.645) = -0.114, \\ q_{.01}^R &= 0.05 + (0.10) \times (-2.33) = -0.183, \end{aligned}$$

and the true values for VaR_α are:

$$\begin{aligned}\text{VaR}_{0.05} &= \$100,000 \times (-0.114) = -\$11,449, \\ \text{VaR}_{0.01} &= \$100,000 \times (-0.183) = -\$18,263.\end{aligned}$$

The R code for the simulation loop is:

```
mu = 0.05
sigma = 0.1
W0 = 1e+05
q.05 = mu + sigma * qnorm(0.05)
q.01 = mu + sigma * qnorm(0.01)
VaR.05 = W0 * q.05
VaR.01 = W0 * q.01
n.obs = 100
n.sim = 1000
set.seed(111)
sim.q = matrix(0, n.sim, 2)
colnames(sim.q) = c("q.05", "q.01")
sim.VaR = matrix(0, n.sim, 2)
colnames(sim.VaR) = c("VaR.05", "VaR.01")
for (sim in 1:n.sim) {
  sim.ret = rnorm(n.obs, mean = mu, sd = sigma)
  muhat = mean(sim.ret)
  sigmahat = sd(sim.ret)
  sim.q[sim, "q.05"] = muhat + sigmahat * qnorm(0.05)
  sim.q[sim, "q.01"] = muhat + sigmahat * qnorm(0.01)
  sim.VaR[sim, "VaR.05"] = W0 * sim.q[sim, "q.05"]
  sim.VaR[sim, "VaR.01"] = W0 * sim.q[sim, "q.01"]
}
```

The histograms of these values, with the asymptotic normal distributions overlayed, are displayed in Figure 7.9. The histograms for the \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ values (for $\alpha = 0.05, 0.01$) are all bell-shaped and centered very close to the respective true values. The asymptotic normal distributions based on (7.44) and (7.46), respectively, fit the histograms very well suggesting that the CLT works well for the sample size $T = 100$. The Monte Carlo estimates of bias are:

```
colMeans(sim.q) - c(q.05, q.01)
##      q.05      q.01
## 0.0001471 0.0003367

colMeans(sim.VaR) - c(VaR.05, VaR.01)
## VaR.05 VaR.01
## 14.71 33.67
```

which confirm that \hat{q}_α^R and $\widehat{\text{VaR}}_\alpha$ are essentially unbiased estimators. The sample standard deviations of the Monte Carlo estimates are also very close to the analytic standard error formulas (7.45) and (7.47), respectively:

```

se.qhat.05 = (sigma/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.05)^2)
se.qhat.01 = (sigma/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.01)^2)
se.VaRhat.05 = (W0 * sigma/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.05)^2)
se.VaRhat.01 = (W0 * sigma/sqrt(n.obs)) * sqrt(1 + 0.5 * qnorm(0.01)^2)
apply(sim.q, 2, sd)

##      q.05      q.01
## 0.01536 0.01901

c(se.qhat.05, se.qhat.01)

## [1] 0.01534 0.01925

apply(sim.VaR, 2, sd)

## VaR.05 VaR.01
##    1536    1901

c(se.VaRhat.05, se.VaRhat.01)

## [1] 1534 1925

```

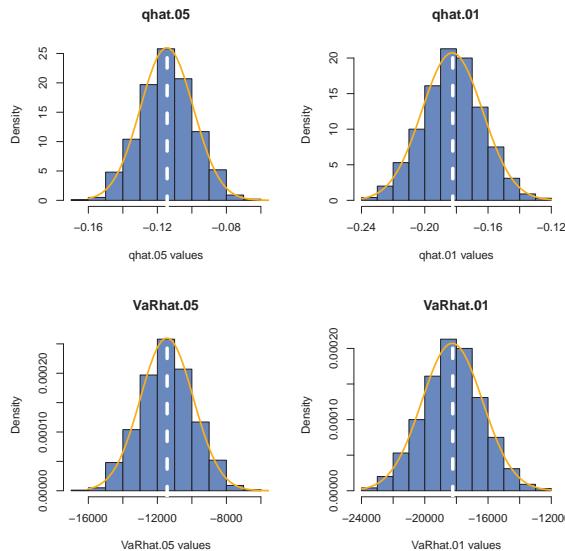


Fig. 7.9 Histograms of \widehat{q}_{α}^R and $\widehat{\text{VaR}}_{\alpha}$ values (for $\alpha = 0.05, 0.01$) computed from $N = 1000$ Monte Carlo samples from CER model.

7.5.4 Statistical Properties of \widehat{q}_{α}^R and $\widehat{\text{VaR}}_{\alpha}$ for Continuously Compounded Returns

TO BE COMPLTED!!!

The Delta Method

TO BE COMPLTED!!!

7.6 Further Reading

TO BE COMPLTED!!!

References

1. Campbell, Lo and MacKinley (1998). *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ.

Chapter 8

The Bootstrap

Updated: July 22, 2016

Copyright © Eric Zivot 2015, 2016

Before computers were cheap and widely available, calculating standard errors for estimators involved using mathematics and probability theory to derive analytic formulas. The formulas are specific to each estimator and are often approximations that rely on large samples and the validity of the Central Limit Theorem to be accurate. Today with modern computers and statistical software such as R, computer intensive resampling methods - *e.g., bootstrapping* - can be used to produce numerical estimator standard errors without the use of mathematical formulas. These bootstrap standard errors are easy to compute and are often more reliable than analytic formulas that rely on approximations.

There are many advantages of the bootstrap over analytic formulas. The most important are the following:

- *Fewer assumptions.* The bootstrap procedure typically requires fewer assumptions than are needed to derive analytic formulas. In particular, the bootstrap does not require the data to be normally distributed or the sample size to be large enough so that the Central Limit Theorem holds.
- *Greater accuracy.* In small samples the bootstrap is often more reliable than analytic approximations based on the Central Limit Theorem. In large samples when the Central Limit Theorem holds the bootstrap can be even be more accurate.
- *Generality.* In many situations, the bootstrap procedure is applied in the same way regardless of the estimator under consideration. That is, you don't need a different bootstrap procedure for each estimator.

This chapter introduces the bootstrap and we show how it can be used to numerically estimate the biases and standard errors of estimators, and to compute approximate confidence intervals for model parameters. Section 8.1 describes how to use the basic nonparametric bootstrap procedure to estimate the biases, standard errors and confidence intervals for the CER model parameters. Section 8.2 discusses some modifications of the nonparametric bootstrap for dealing with time dependence in time series data. Section 8.3 describes the parametric bootstrap procedure.

The R packages used in this chapter are **boot** and **IntroCompFinR**. Have these packages installed and loaded in R before replicating the examples.

8.1 The Nonparametric Bootstrap

In this section, we describe the easiest and most common form of the bootstrap: the *nonparametric bootstrap*. As we shall see, the nonparametric bootstrap procedure is very similar to a Monte Carlo simulation experiment. The main difference is how the random data is generated. In a Monte Carlo experiment, the random data is created from a computer random number generator for a specific probability distribution (e.g., normal distribution). In the nonparametric bootstrap, the random data is created by resampling with replacement from the original data.

To motivate the nonparametric bootstrap procedure, consider the CER model for continuously compounded returns

$$\begin{aligned} R_t &= \mu + \varepsilon_t, \quad t = 1, \dots, T, \\ \varepsilon_t &\sim \text{GWN}(0, \sigma^2). \end{aligned}$$

The observed sample of size T is $\{r_1, \dots, r_T\}$. The goal is to compute $\widehat{\text{bias}}(\hat{\mu})$, $\widehat{\text{bias}}(\hat{\sigma})$, $\widehat{\text{se}}(\hat{\mu})$, $\widehat{\text{se}}(\hat{\sigma})$ and 95% confidence intervals for μ and for σ using the nonparametric bootstrap. Under the assumptions of the CER model, we know that $\hat{\mu}$ is unbiased and that $\hat{\sigma}$ is effectively unbiased. We have the following analytic formulas for the estimated standard errors:

$$\widehat{\text{se}}(\hat{\mu}) = \frac{\hat{\sigma}}{\sqrt{T}}, \quad \widehat{\text{se}}(\hat{\sigma}) \approx \frac{\hat{\sigma}}{\sqrt{2T}},$$

where $\widehat{\text{se}}(\hat{\mu})$ is an exact calculation and $\widehat{\text{se}}(\hat{\sigma})$ is an approximation based on the CLT. In addition, the formula for an approximate 95% confidence interval for $\theta = \mu, \sigma$ is

$$\hat{\theta} \pm 2 \cdot \widehat{\text{se}}(\hat{\theta}).$$

This formula relies on the assumption that $\hat{\theta}$ is either normally distributed or is asymptotically normally distributed. As we shall see, the nonparametric bootstrap will give results that match the analytic results very closely.

We will also show how to use the nonparametric bootstrap to compute the bias, estimated standard error for an estimator of Value-at-Risk. Recall, for an initial investment of W_0 the estimate of VaR_α is given by

$$\widehat{\text{VaR}}_\alpha = W_0 \left(e^{\hat{q}_\alpha^r} - 1 \right), \quad (8.1)$$

where $\hat{q}_\alpha^r = \hat{\mu} + \hat{\sigma} q_\alpha^Z$ and q_α^Z is the α -quantile for $Z \sim N(0, 1)$. Analytic formulas for $\widehat{\text{bias}}(\widehat{\text{VaR}}_\alpha)$ and $\widehat{\text{se}}(\widehat{\text{VaR}}_\alpha)$ are not easy to compute. However, computing numeric estimates of these values from the nonparametric bootstrap is almost trivial.

Example 8.1. Data for bootstrap examples

To illustrate the bootstrap for estimating bias, standard errors, and confidence intervals for estimates of CER model parameters (and functions of CER parameters such as VaR), we use data on monthly continuously compounded returns for Microsoft over the period January 1998, through May 2012, from the **IntroCompFinR** package. The data is the same as that used in Chapter 5 and is constructed as follows:

```
library(IntroCompFinR)
library(PerformanceAnalytics)
data(msftDailyPrices)
msftPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
smpl = "1998-01::2012-05"
msftPrices = msftPrices[smpl]
msftRetC = na.omit(Return.calculate(msftPrices, method = "log"))
head(msftRetC, n = 3)

##           MSFT
## Feb 1998 0.12786
## Mar 1998 0.05421
## Apr 1998 0.00689
```

The CER model estimates for μ and σ , with estimated standard errors, are

```
n.obs = nrow(msftRetC)
# compute estimates
muhat = mean(msftRetC)
sigmahat = sd(msftRetC)
estimates = c(muhat, sigmahat)
# compute estimated standard errors
se.muhat = sigmahat/sqrt(n.obs)
se.sigmahat = sigmahat/sqrt(2 * n.obs)
stdErrors = c(se.muhat, se.sigmahat)
# show estimates with estimated standard errors
ans = rbind(estimates, stdErrors)
colnames(ans) = c("Mu", "Sigma")
ans

##           Mu   Sigma
## estimates 0.00413 0.1002
## stdErrors 0.00764 0.0054
```

8.1.1 Procedure for the nonparametric bootstrap

The procedure for the nonparametric bootstrap is as follows:

1. *Resample.* Create B bootstrap samples by sampling with replacement from the original data $\{r_1, \dots, r_T\}$. Each bootstrap sample has T observations (same as the original sample)

$$\{r_{11}^*, r_{12}^*, \dots, r_{1T}^*\} = 1\text{st bootstrap sample}$$

⋮

$$\{r_{B1}^*, r_{B2}^*, \dots, r_{BT}^*\} = B\text{th bootstrap sample}$$

2. Estimate θ . From each bootstrap sample estimate θ and denote the resulting estimate $\hat{\theta}^*$. There will be B values of $\hat{\theta}^*$: $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$.
3. Compute statistics. Using $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ compute estimate of bias, standard error, and approximate 95% confidence interval.

This procedure is very similar to the procedure used to perform a Monte Carlo experiment described in Chapter 7. The main difference is how the hypothetical samples are created. With the nonparametric bootstrap, the hypothetical samples are created by resampling with replacement from the original data. In this regard the bootstrap treats the sample as if it were the population. This ensures that the bootstrap samples inherit the same distribution as the original data - *whatever that distribution may be*. If the original data is normally distributed, then the nonparametric bootstrap samples will also be normally distributed. If the data is Student's t distributed, then the nonparametric bootstrap samples will be Student's t distributed. With Monte Carlo simulation, the hypothetical samples are simulated under an assumed model and distribution. This requires one to specify values for the model parameters and the distribution from which to simulate.

Bootstrap bias estimate

The nonparametric bootstrap can be used to estimate the bias of an estimator $\hat{\theta}$ using the bootstrap estimates $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$. The key idea is to treat the empirical distribution (i.e., histogram) of the bootstrap estimates $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ as an approximate to the unknown distribution of $\hat{\theta}$.

Recall, the bias of an estimator is defined as

$$E[\hat{\theta}] - \theta.$$

The bootstrap estimate of bias is given by

$$\widehat{\text{bias}}_{\text{boot}}(\hat{\theta}, \theta) = \frac{1}{B} \sum_{j=1}^B \hat{\theta}_j^* - \hat{\theta}. \quad (8.2)$$

(bootstrap mean - estimate)

The bootstrap bias estimate (8.2) is the difference between the mean of the bootstrap estimates of θ and the sample estimate of θ . This is similar to the Monte Carlo estimate of bias discussed in Chapter 7. However, the Monte Carlo estimate of bias is the difference between the mean of the Monte Carlo estimates of θ and the true value of θ . The bootstrap estimate of bias does not require knowing the true value of θ . Effectively, the bootstrap treats

the sample estimate $\hat{\theta}$ as the population value θ and the bootstrap mean $\frac{1}{B} \sum_{j=1}^B \hat{\theta}_j^*$ as an approximation to $E[\hat{\theta}]$. Here, $\widehat{bias}_{boot}(\hat{\theta}, \theta) = 0$ if the center of the histogram of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ is at $\hat{\theta}$.

Bootstrap standard error estimate

The bootstrap estimate of $\widehat{se}(\hat{\theta})$ is given by the sample standard deviation of the bootstrap estimates $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$:

$$\widehat{se}_{boot}(\hat{\theta}) = \sqrt{\frac{1}{B-1} \sum_{j=1}^B \left(\hat{\theta}_j^* - \frac{1}{B} \sum_{j=1}^B \hat{\theta}_j^* \right)^2}. \quad (8.3)$$

Here, $\widehat{se}_{boot}(\hat{\theta})$ is the size of a typical deviation of a bootstrap estimate, $\hat{\theta}^*$, from the mean of the bootstrap estimates (typical deviation from the middle of the histogram of the bootstrap estimates). This is very closely related to the Monte Carlo estimate of $\widehat{se}(\hat{\theta})$, which is the sample standard deviation of the estimates of θ from the Monte Carlo samples.

Bootstrap confidence intervals

Bootstrap confidence intervals are computed in two different ways depending on whether the distribution of the bootstrap estimates $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ appears to be normal. If the bootstrap distribution appears to be normal then compute a 95% confidence interval for θ using

$$\hat{\theta} \pm 2 \cdot \widehat{se}_{boot}(\hat{\theta}). \quad (8.4)$$

This is the same rule-of-thumb that is justified by the CLT applied to $\hat{\theta}$. If the bootstrap distribution does not appear to be normal then compute a 95% confidence interval for θ using

$$[\hat{q}_{.025}^*, \hat{q}_{.975}^*], \quad (8.5)$$

where $\hat{q}_{.025}^*$ and $\hat{q}_{.975}^*$ are the 2.5% and 97.5% empirical quantiles of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$, respectively. By construction, 95% of the bootstrap estimates lie in the quantile-based interval (8.5).

Whether to use (8.4) or (8.5) depends on the empirical distribution of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$. The basic shape of this distribution can be visualized by the histogram of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ and the normality of the distribution can be visually evaluated using the normal QQ-plot of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$. If the histogram is asymmetric and/or not bell-shaped or if the normal QQ-plot is not linear then normality is suspect. In this case (8.5) should be used to compute a 95% confidence interval for θ .

8.1.2 Performing the Nonparametric Bootstrap in R

The nonparametric bootstrap procedure is easy to perform in R. You can implement the procedure by “brute force” in very much the same way as you perform a Monte Carlo experiment. In this approach you program all parts of the bootstrapping procedure. Alternatively, you can use the R package **boot** which contains functions for automating certain parts of the bootstrapping procedure.

Brute force implementation

In the brute force implementation you program all parts of the bootstrap procedure. This typically involves a three steps:

1. Sample with replacement from the original data using the R function `sample()` to create $\{r_1^*, r_2^*, \dots, r_T^*\}$. Do this B times.
2. Compute B values of the statistic of interest $\hat{\theta}$ from each bootstrap sample giving $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$
3. Compute the bootstrap bias and SE values from $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ using the R functions `mean()` and `sd()`, respectively.
4. Compute the histogram and normal QQ-plot of $\{\hat{\theta}_1^*, \dots, \hat{\theta}_B^*\}$ using the R functions `hist()` and `qqnorm()`, respectively, to see if the bootstrap distribution looks normal.

Typically steps 1 and 2 are performed inside a “for loop” or within the R function `apply()`. In step 1 sampling with replacement for the original data is performed with the R function `sample()`. To illustrate, consider sampling with replacement from the 5×1 return vector $\mathbf{r} = (0.1, 0.05, -0.02, 0.03, -0.04)'$.

```
r = c(0.1, 0.05, -0.02, 0.03, -0.04)
r

## [1] 0.10 0.05 -0.02 0.03 -0.04
```

Recall, in R we can extract elements from a vector by subsetting based on their location, or index, in the vector. Since `r` has five elements, its index is the integer vector `1:5`. Then to extract the first, second and fifth elements of `r` use the index vector `c(1, 2, 5)`:

```
r[c(1, 2, 5)]

## [1] 0.10 0.05 -0.04
```

Using this idea, you can extract a random sample (of any given size) with replacement from `r` by creating a random sample with replacement of the integers $\{1, 2, \dots, 5\}$ and using this set of integers to extract the sample from `r`. The R function `sample()` can be used to do this process. When you pass a positive integer value `n` to `sample()`, with the optional argument `replace=TRUE`, it returns a random sample from the set of integers from 1 to `n`. For example, to create a random sample with replacement of size 5 from the integers $\{1, 2, \dots, 5\}$ use:

```
set.seed(123)
idx = sample(5, replace = TRUE)
idx

## [1] 2 4 3 5 5
```

We can then get a random sample with replacement from the vector `r` by subsetting using the index vector `idx`:

```
r[idx]

## [1] 0.05 0.03 -0.02 -0.04 -0.04
```

This two step process is automated when you pass a vector of observations to `sample()`:

```
set.seed(123)
sample(r, replace = TRUE)

## [1] 0.05 0.03 -0.02 -0.04 -0.04
```

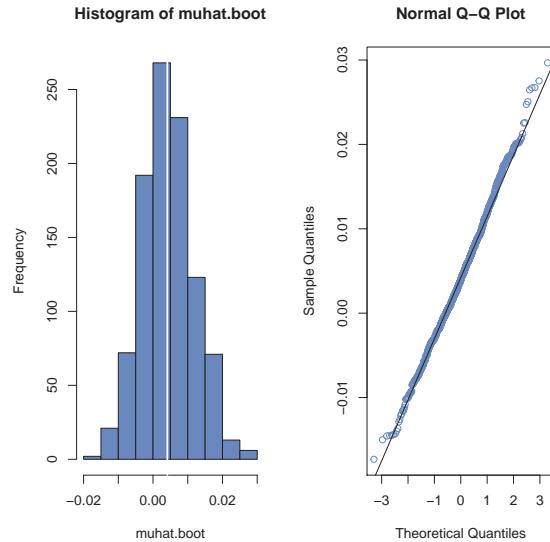


Fig. 8.1 Histogram and normal QQ-plot of bootstrap estimates of μ for Microsoft.

Example 8.2. Brute force nonparametric bootstrap of CER model in R

Consider using the nonparametric bootstrap to compute estimates of the bias and standard error for $\hat{\mu}$ in the CER model for Microsoft. The R code for the brute force “for loop” to implement the nonparametric bootstrap is

```
B = 999
muhat.boot = rep(0, B)
n.obs = nrow(msftRetC)
```

```
set.seed(123)
for (i in 1:B) {
  boot.data = sample(msftRetC, n.obs, replace = TRUE)
  muhat.boot[i] = mean(boot.data)
}
```

The bootstrap bias estimate is

```
mean(muhat.boot) - muhat
## [1] 0.000199
```

which is very close to zero and confirms that $\hat{\mu}$ is unbiased. The bootstrap standard error estimate is

```
sd(muhat.boot)
## [1] 0.00746
```

and is equal (to three decimals) to the analytic standard error estimate computed earlier. This confirms that the nonparametric bootstrap accurately computes $\widehat{SE}(\hat{\mu})$. Figure 8.1 shows the histogram and normal QQ-plot of the bootstrap estimates created with

```
par(mfrow = c(1, 2))
hist(muhat.boot, col = "cornflowerblue")
abline(v = muhat, col = "white", lwd = 2)
qqnorm(muhat.boot, col = "cornflowerblue")
qqline(muhat.boot)
par(mfrow = c(1, 1))
```

The distribution of the bootstrap estimates looks normal. As a result, the bootstrap 95% confidence interval for μ has the form

```
se.boot = sd(muhat.boot)
lower = muhat + se.boot
upper = muhat - se.boot
width = upper - lower
cbind(lower, upper, width)

##      lower     upper    width
## [1,] 0.0116 -0.00333 -0.0149
```

R package boot

The R package **boot** implements a variety of bootstrapping techniques including the basic non-parametric bootstrap described above. The **boot** package was written to accompany the Davidson and Hinkley (1997) textbook *Bootstrap Methods and Their Application*. There two main functions in **boot** are **boot()** and **boot.ci()**, respectively. The **boot()** function implements the bootstrap for a statistic computed from a user-supplied function. The **boot.ci()**

function computes bootstrap confidence intervals given the output from the `boot()` function.

The arguments to `boot()` are

```
library(boot)
args(boot)

## function (data, statistic, R, sim = "ordinary", stype = c("i",
##      "f", "w"), strata = rep(1, n), L = NULL, m = 0, weights = NULL,
##      ran.gen = function(d, p) d, mle = NULL, simple = FALSE, ...,
##      parallel = c("no", "multicore", "snow"), ncpus = getOption("boot.ncpus",
##      1L), cl = NULL)
## NULL
```

where `data` is a data object (typically a vector or matrix but can be a time series object too), `statistic` is a user-specified function to compute the statistic of interest, and `R` is the number of bootstrap replications. The remaining arguments are not important for the basic non-parametric bootstrap. The function assigned to the argument `statistic` has to be written in a specific form, which is illustrated in the next example. The `boot()` function returns an object of class “`boot`” for which there are print and plot methods.

The arguments to `boot.ci()` are

```
args(boot.ci)

## function (boot.out, conf = 0.95, type = "all", index = 1L:min(2L,
##      length(boot.out$t0)), var.t0 = NULL, var.t = NULL, t0 = NULL,
##      t = NULL, L = NULL, h = function(t) t, hdot = function(t) rep(1,
##      length(t)), hinvt = function(t) t, ...)
## NULL
```

where `boot.out` is an object of class “`boot`”, `conf` specifies the confidence level, and `type` is a subset from `c("norm", "basic", "stud", "perc", "bca")` indicating the type of confidence interval to compute. The choices “`norm`” and “`perc`” compute the normal confidence interval (8.4) and the percentile confidence interval (8.5), respectively. The remaining arguments are not important for the computation of basic bootstrap confidence intervals.

Example 8.3. Nonparametric bootstrap of CER model using the R package `boot`

To use the `boot()` function to implement the bootstrap for $\hat{\mu}$, a function must be specified to compute $\hat{\mu}$ for each bootstrap sample. The function must have two arguments: `x` and `idx`. Here, `x` represents the original data and `idx` represents the random integer index (created internally by `boot()`) to subset `x` for each bootstrap sample. For example, a function to be passed to `boot()` for $\hat{\mu}$ is

```
mean.boot = function(x, idx) {
  # arguments: x data to be resampled idx vector of scrambled indices created by
  # boot() function value: ans mean value computed using resampled data
  ans = mean(x[idx])
  ans
}
```

To implement the nonparametric bootstrap for $\hat{\mu}$ with 999 samples use

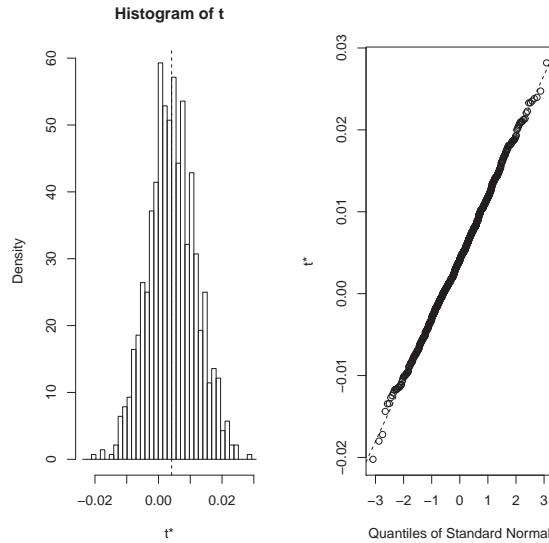


Fig. 8.2 plot method for objects of class “boot”

```
set.seed(123)
muhat.boot = boot(msftRetC, statistic = mean.boot, R = 999)
class(muhat.boot)

## [1] "boot"

names(muhat.boot)

## [1] "t0"         "t"          "R"          "data"       "seed"
## [6] "statistic"  "sim"        "call"       "stype"      "strata"
## [11] "weights"
```

The returned object `muhat.boot` is of class “boot”. The component `t0` is the sample estimate $\hat{\mu}$, and the component `t` is a 999×1 matrix containing the bootstrap estimates $\{\hat{\mu}_1^*, \dots, \hat{\mu}_{999}^*\}$. The print method shows the sample estimate, the bootstrap bias and the bootstrap standard error:

```
muhat.boot

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = msftRetC, statistic = mean.boot, R = 999)
##
##
## Bootstrap Statistics :
##      original   bias   std. error
## t1*  0.00413  0.000199    0.00745
```

These statistics can be computed directly from the components of `muhat.boot`

```
muhat.boot$t0

## [1] 0.00413

mean(muhat.boot$t) - muhat.boot$t0

## [1] 0.000199

sd(muhat.boot$t)

## [1] 0.00745
```

A histogram and normal QQ-plot of the bootstrap values, shown in Figure 8.2, can be created using the `plot` method

```
plot(muhat.boot)
```

Normal and percentile 95% confidence intervals can be computed using

```
boot.ci(muhat.boot, conf = 0.95, type = c("norm", "perc"))

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = muhat.boot, conf = 0.95, type = c("norm",
##           "perc"))
##
## Intervals :
## Level      Normal          Percentile
## 95%   (-0.0107,  0.0185 )  (-0.0101,  0.0189 )
## Calculations and Intervals on Original Scale
```

Because the bootstrap distribution looks normal, the normal and percentile confidence intervals are very similar.

The CER model estimate $\hat{\sigma}$ can be “bootstrapped” in a similar fashion. First, we write a function to compute $\hat{\sigma}$ for each bootstrap sample

```
sd.boot = function(x, idx) {
  ans = sd(x[idx])
  ans
}
```

Then we call `boot()` with `statistic=sd.boot`

```
set.seed(123)
sigmahat.boot = boot(msftRetC, statistic = sd.boot, R = 999)
sigmahat.boot

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
```

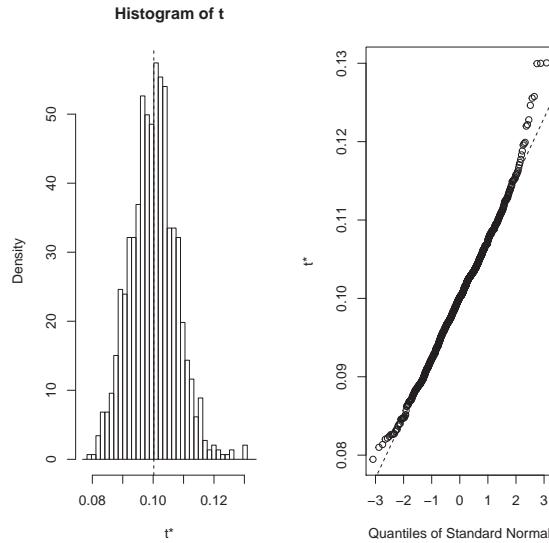


Fig. 8.3 Bootstrap distribution for $\hat{\sigma}$

```
##  
##  
## Call:  
## boot(data = msftRetC, statistic = sd.boot, R = 999)  
##  
##  
## Bootstrap Statistics :  
##      original    bias    std. error  
## t1*     0.1 -0.000173    0.00765
```

The bootstrap distribution, shown in Figure 8.3, looks a bit non-normal so we compute a 95% confidences interval using (8.4) and (8.5)

```
boot.ci(sigmahat.boot, conf = 0.95, type = c("norm", "perc"))

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = sigmahat.boot, conf = 0.95, type = c("norm",
##           "perc"))
##
## Intervals :
## Level      Normal          Percentile
## 95%   ( 0.0854,  0.1154 )  ( 0.0849,  0.1153 )
## Calculations and Intervals on Original Scale
```

Example 8.4. Nonparametric bootstrap for Value-at-Risk

The real power of the bootstrap procedure come when we apply it to an estimate like $\widehat{\text{VaR}}_\alpha$ for which there is no easy analytic formulas for bias($\widehat{\text{VaR}}_\alpha$) and se($\widehat{\text{VaR}}_\alpha$). In this situation, the bootstrap procedure easily computes numerical estimates of bias and the standard error. All we need is an R function for computing $\widehat{\text{VaR}}_\alpha$ from the formula (8.1):

```
VaR.boot = function(x, idx, alpha = 0.05, w0 = 1e+05) {
  # x.mat data to be resampled idx vector of scrambled indices created by boot()
  # function alpha probability value for VaR calculation w0 value of initial
  # investment value: ans Value-at-Risk computed using resampled data
  q = mean(x[idx]) + sd(x[idx]) * qnorm(alpha)
  VaR = (exp(q) - 1) * w0
  VaR
}
```

Here, the function `VaR.boot()` has additional arguments `alpha` and `w0`, which specify the VaR tail probability and the initial investment value, respectively. To compute the nonparametric bootstrap for $\widehat{\text{VaR}}_\alpha$ with $\alpha = 0.05$ and $W_0 = 100000$ use

```
VaR.boot = boot(msftRetC, statistic = VaR.boot, R = 999)
VaR.boot

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = msftRetC, statistic = VaR.boot, R = 999)
##
##
## Bootstrap Statistics :
##      original    bias    std. error
## t1*   -14846      78      1253
```

8.2 Bootstrapping Time Series Data

To be completed

- illustrate using overlapping returns
- Show how the basic non-parametric bootstrap does not preserve the time dependence in the data and motivate block resampling

8.2.1 Block bootstrap

To be completed

- Use the `tsboot` package

8.3 Parametric Bootstrap

To be completed

8.4 Further Reading

To be completed

8.5 Problems

References

1. Davidson. Bootstrap Methods and Their Applications.

Chapter 9

Hypothesis Testing in the CER Model

Updated: October 15, 2016

Copyright © Eric Zivot 2015, 2016

1. Introduction

- a. The core of statistical analysis involves estimation and inference. The previous chapters considered estimation of the CER model. In this chapter we discuss issues of statistical inference within the CER model. Statistical inference is about asking questions about a model and using statistical procedures to obtain answers to these questions with a given level of confidence. For example, we might find that our estimate the monthly mean of an asset return is positive but due to large estimation error we might ask the question: is the true monthly mean return positive? Statistical hypothesis testing gives us a rigorous framework for answering such a question.
- b. The R packages used in this chapter are **IntroCompFinR**, **PerformanceAnalytics**, **tseries**, and **zoo**. Make sure these packages are installed and loaded before replicating the examples.

2. Hypothesis Testing in the CER Model

- a. Coefficient tests: single assets
 - i. tests for specific values
 - ii. distribution of tests under normality of data
 - iii. distribution of tests based on asymptotic normality of estimators
 - iv. Duality between confidence intervals and hypothesis tests
 - v. tests for sign
 - vi. joint tests on parameters
- b. Using Monte Carlo Simulation to Understand Hypothesis tests
 - i. Size
 - ii. Power
- c. Coefficient tests: pairs of assets

- i. common mean, variance, correlation
- d. Specification tests
 - i. Normal distribution
 - A. Tests for skewness and kurtosis
 - B. JB test
 - C. What to do if returns are not normally distributed?
 - ii. No autocorrelation
 - A. individual and joint tests
 - iii. Constant parameters (covariance stationarity)
 - A. Formal tests
 - B. Informal diagnostics
- 3. Appendix: Distributions for test statistics
 - a. Student's t distribution
 - b. Chi-square distribution

9.1 Hypothesis Testing: General Principles

Hypothesis testing is discussed in almost all introductory textbooks on statistics, and the reader is assumed to be familiar with the basic concepts. In this section we provide a brief and concise review of the general principles of classical hypothesis testing. In the following sections we apply these principles to answer questions about the CER model parameters and assumptions.¹

The main steps for conducting a classical hypothesis test are as follows:

1. Specify the hypotheses to be tested:

$$H_0 : \text{null hypothesis versus } H_1 : \text{alternative hypothesis.}$$

The *null hypothesis* is the maintained hypothesis or the hypothesis to be tested against the data. The *alternative hypothesis* specifies what is assumed to be true if the null hypothesis is false. This can be specific or vague depending on the context.

2. Specify the significance level of the test:

$$\alpha = \text{significance level} = Pr(\text{Reject } H_0 | H_0 \text{ is true}).$$

The significance level of a test specifies the probability of making a certain type of decision error: rejecting the null hypothesis when the null hypothesis is, in fact, true. In practice, the significance level α is chosen to be a small number like 0.01 or 0.05.

¹ In this book, we do not consider evaluating hypotheses from a Bayesian perspective.

3. Construct a test statistic, S , from the observed data whose probability distribution is known when the H_0 is true.
4. Use the test statistic S to evaluate the data evidence regarding the validity of H_0 . Typically, if S is a big number then there is strong data evidence against H_0 and H_0 should be rejected; otherwise, there is not strong data evidence against H_0 and H_0 should not be rejected.
5. Decide to reject H_0 at the specified significance level α if the value of the test statistic S falls in the rejection region of the test. Usually, the rejection region for S is determined by a critical value cv_α such that:

$$\begin{aligned} S > cv_\alpha &\Rightarrow \text{reject } H_0, \\ S \leq cv_\alpha &\Rightarrow \text{do not reject } H_0. \end{aligned}$$

Smaller values of α make cv_α larger and require more data evidence (i.e., larger S) to reject H_0 .

6. Alternatively, decide to reject H_0 at the specified significance level α if the p-value of the test statistic S is less than α .

It is clear that hypothesis testing involves making a decision: reject H_0 or do not reject H_0 . Notice that if the data evidence strongly favors H_0 then we say “we do not reject H_0 ” instead of “accept H_0 ”. We can rarely know the truth with absolute certainty with a finite amount of data so it is more appropriate to say “do not reject H_0 ” than to say “accept H_0 ”.²

		Reality
Decision	H_0 is true	H_0 is false
Reject H_0	Type I error	No error
Do not reject H_0	No error	Type II error

Table 9.1 Decision table for hypothesis testing

Table 9.2 shows the 2×2 decision table associated with hypothesis testing. In reality there are two states of the world: either H_0 is true or it is not. There are two decisions to make: reject H_0 or don't reject H_0 . If the decision corresponds with reality then the correct decision is made and there is no decision error. This occurs in the off-diagonal elements of the table. If the decision and reality disagree then there is a decision error. These errors are in the diagonal elements of the table. Type I error results when the decision to reject H_0 occurs when, in fact, H_0 is true. Type II error happens when the decision not to reject H_0 occurs when H_0 is false. To put these types of errors in context, consider a jury in a capital murder trial in the US³. In the US a defendant is considered innocent until proven guilty. Here, the null hypothesis to be tested is:

² As an extreme example, consider testing H_0 : all swans are white. To logically accept this hypothesis you would have to show that every swan that ever existed is white.

³ A capital murder trial is one in which the defendant is eligible for the death penalty.

$$H_0 : \text{defendant is innocent.}$$

The alternative is:

$$H_1 : \text{defendant is guilty.}$$

The decision table for the jury is shown in Table 9.2. Type I error occurs when the jury convicts an innocent defendant and puts that person on “death row”. Clearly, this is a terrible mistake. To avoid this type of mistake the jury should have a very small (close to zero) significance level for evaluating evidence so that Type I error very rarely occurs. This is why typical jury instructions are to convict only if evidence of guilt is presented beyond reasonable doubt. Type II error happens when the jury does not convict (acquits) a guilty defendant and sets that person free. This is also a terrible mistake, but perhaps it is not as terrible as convicting the innocent person. Notice that there is a conflict between Type I error and Type II error. In order to completely eliminate Type I error, you can never reject H_0 . That is, to avoid ever sending innocent people to “death row” you must never convict anyone. But if you never convict anyone then you never convict guilty people either and this increases the occurrences of Type II errors.

The *significance level* of a test is the probability of Type I error:

$$\alpha = Pr(\text{Type I error}) = Pr(\text{Reject } H_0 | H_0 \text{ is true}).$$

The *power* of a test is one minus the probability of Type II error:

$$\pi = 1 - Pr(\text{Type II error}) = Pr(\text{Reject } H_0 | H_0 \text{ is false}).$$

In classical hypothesis tests, the goal is to construct a test that has a small significance level (α close to zero) that you can control and has high power (π close to one). In general, an optimal test is one that has the highest possible power for a given significance level. In the jury example, an optimal test would be the one in which the jury has the highest probability of convicting a guilty defendant while at the same time has a very low probability of convicting an innocent defendant. A difficult problem indeed!

		Reality	
		Defendant is innocent	Defendant is guilty
Decision	Convict	Type I error	No error
	Acquit	No error	Type II error

Table 9.2 Decision table jury on capital murder trial

9.2 Hypothesis Testing in the CER Model

In this section, we discuss hypothesis testing in the context of the CER model for asset returns:

$$\begin{aligned} R_{it} &= \mu_i + \varepsilon_{it}, \quad i = 1, \dots, N; \quad t = 1, \dots, T \\ \{\varepsilon_{it}\}_{t=1}^T &\sim \text{GWN}(0, \sigma_i^2), \\ \text{cov}(\varepsilon_{it}, \varepsilon_{js}) &= \begin{cases} \sigma_{ij} & t = s \\ 0 & t \neq s \end{cases}. \end{aligned} \tag{9.1}$$

In (9.1), R_{it} is the monthly return (continuously compounded or simple) on asset i in month t . Define $\mathbf{R}_t = (R_{1t}, \dots, R_{Nt})'$. The CER model in matrix form is:

$$\begin{aligned} \mathbf{R}_t &= \boldsymbol{\mu} + \boldsymbol{\varepsilon}_t, \\ \boldsymbol{\varepsilon}_t &\sim \text{iid } N(\mathbf{0}, \boldsymbol{\Sigma}). \end{aligned} \tag{9.2}$$

We consider two types of hypothesis tests: (1) tests for model coefficients; and (2) tests for model assumptions.⁴ These tests can be specified for the CER model for a single asset using (9.1), or they can be specified for the CER model for all assets using (9.2). The most common tests for model coefficients for a single asset are tests for a specific value. Typical hypotheses to be tested are:

$$\begin{aligned} H_0 : \mu_i &= \mu_i^0 \text{ vs. } H_1 : \mu_i \neq \mu_i^0, \\ H_0 : \sigma_i &= \sigma_i^0 \text{ vs. } H_1 : \sigma_i \neq \sigma_i^0, \\ H_0 : \rho_{ij} &= \rho_{ij}^0 \text{ vs. } H_1 : \rho_{ij} \neq \rho_{ij}^0. \end{aligned}$$

Here, the alternative hypotheses are *two-sided*.

Also common are tests for specific sign. Example hypotheses to be tested are:

$$\begin{aligned} H_0 : \mu_i &= 0 \text{ vs. } H_1 : \mu_i > 0 \text{ or } H_1 : \mu_i < 0, \\ H_0 : \rho_{ij} &= 0 \text{ vs. } H_1 : \rho_{ij} > 0 \text{ or } H_1 : \rho_{ij} < 0. \end{aligned}$$

Here, the alternative hypotheses *one-sided*. Sometimes we are interested in testing hypotheses about the relationship between model coefficients for two or more assets. For example,

$$H_0 : \mu_1 = \mu_2 \text{ vs. } H_1 : \mu_1 \neq \mu_2,$$

or

$$H_0 : \mu_1 = \mu_2 \text{ vs. } H_1 : \mu_1 > \mu_2.$$

⁴ Tests for model assumptions are sometimes called *model specification tests*.

In the CER model (9.1), asset returns are assumed to be normally distributed. The hypotheses to be tested for this assumption are:

$$H_0 : R_{it} \sim N(\mu_i, \sigma_i^2) \text{ vs. } H_1 : R_{it} \sim \text{not normal.}$$

In addition, it is assumed that asset returns are uncorrelated over time. One can test the hypothesis that each autocorrelation (up to some maximum lag J) is zero:

$$\begin{aligned} H_0 &: \rho_j = \text{cor}(r_{it}, r_{i,t-j}) = 0, j = 1, \dots, J \text{ vs.} \\ H_1 &: \rho_j = \text{cor}(r_{it}, r_{i,t-j}) \neq 0, \end{aligned}$$

or one can test the joint hypothesis that all autocorrelation coefficients (up to some maximum lag J) are zero:

$$\begin{aligned} H_0 &: \rho_1 = \rho_2 = \dots = \rho_J = 0 \text{ vs.} \\ H_1 &: \rho_j \neq 0 \text{ for some } j \leq J \end{aligned}$$

Lastly, it is assumed that asset returns are covariance stationary. This assumption implies that all model parameters are constant over time. The hypotheses of interest are then:

$$\begin{aligned} H_0 &: \mu_i, \sigma_i \text{ and } \rho_{ij} \text{ are constant over entire sample vs.} \\ H_1 &: \mu_i, \sigma_i \text{ or } \rho_{ij} \text{ changes in some sub-sample.} \end{aligned}$$

The following sections describe test statistics for the above listed hypotheses. The following data is used to illustrate each test.

Example 9.1. Data for hypothesis testing examples.

The examples in this chapter use the monthly continuously compounded returns on Microsoft, Starbucks and the S&P 500 index over the period January 1998 through May 2012 created using:

```
library(IntroCompFinR)
library(PerformanceAnalytics)
data(msftDailyPrices, sp500DailyPrices, sbuxDailyPrices)
msftPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
sbuxPrices = to.monthly(sbuxDailyPrices, OHLC = FALSE)
sp500Prices = to.monthly(sp500DailyPrices, OHLC = FALSE)
# set sample to match other chapters
smpl = "1998-01::2012-05"
msftPrices = msftPrices[smpl]
sbuxPrices = sbuxPrices[smpl]
sp500Prices = sp500Prices[smpl]
# calculate returns
msftRetC = na.omit(Return.calculate(msftPrices, method = "log"))
sp500RetC = na.omit(Return.calculate(sp500Prices, method = "log"))
```

```

sbuxRetC = na.omit(Return.calculate(sbuxPrices, method = "log"))
# merged data set
cerRetC = merge(msftRetC, sbuxRetC, sp500RetC)
colnames(cerRetC) = c("MSFT", "SBUX", "SP500")
head(cerRetC, n = 3)

##           MSFT    SBUX    SP500
## Feb 1998 0.12786 0.0793 0.06808
## Mar 1998 0.05421 0.1343 0.04874
## Apr 1998 0.00689 0.0610 0.00904

```

The monthly continuously compounded returns, with a common y-axis, are illustrated in Figure 9.1 created using:

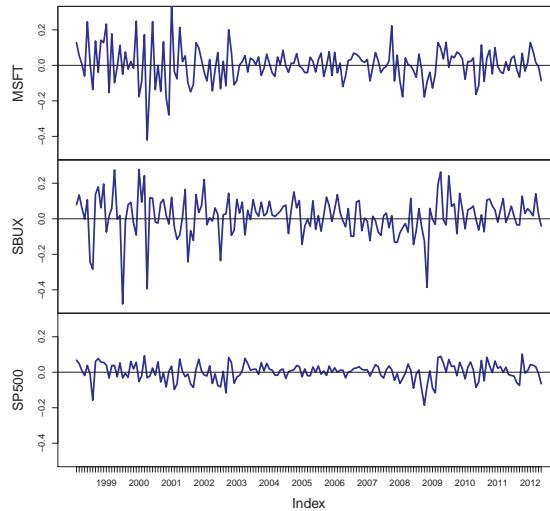


Fig. 9.1 Monthly cc returns on Microsoft, Starbucks and the S&P 500 Index.

9.3 Tests for Individual Parameters: t-tests

In this section, we present test statistics for testing hypotheses that certain model coefficients equal specific values. In some cases, we can derive *exact tests*. For an exact test, the pdf of the test statistic assuming the null hypothesis is true is known exactly for a finite sample size T . More generally, however, we rely on *asymptotic tests*. For an asymptotic test, the pdf of the test statistic assuming the null hypothesis is true is not known exactly for a finite sample size T , but can be approximated by a known pdf. The approximation is justified by

the Central Limit Theorem (CLT) and the approximation becomes exact as the sample size becomes infinitely large.

9.3.1 Exact tests under normality of data

In the CER model (9.1), consider testing the hypothesis that the mean return, μ_i , is equal to a specific value μ_i^0 :

$$H_0 : \mu_i = \mu_i^0. \quad (9.3)$$

The alternative hypothesis can be either two-sided or one-sided. The two-sided alternative is:

$$H_1 : \mu_i \neq \mu_i^0. \quad (9.4)$$

Two-sided alternatives are used when we don't care about the sign of $\mu_i - \mu_i^0$ under the alternative. With one-sided alternatives, we care about the sign of $\mu_i - \mu_i^0$ under the alternative:

$$H_1 : \mu_i > \mu_i^0 \text{ or } H_1 : \mu_i < \mu_i^0. \quad (9.5)$$

How do we come up with a test statistic S for testing (9.3) against (9.4) or (9.5)? We generally use two criteria: (1) we know the pdf of S assuming (9.3) is true; and (2) the value of S should be big if the alternative hypotheses (9.4) or (9.5) are true.

To simplify matters, let's assume that the value of σ_i in (9.1) is known and does not need to be estimated. Consider testing (9.3) against the two-sided alternative (9.4) using a 5% significance level. Let $\{R_t\}_{t=1}^T$ denote a random sample from the CER model. We estimate μ_i using the sample mean:

$$\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^T R_{it}. \quad (9.6)$$

Under the null hypothesis (9.3), it is assumed that $\mu_i = \mu_i^0$. Hence, if the null hypothesis is true then $\hat{\mu}_i$ computed from data should be close to μ_i^0 . If $\hat{\mu}_i$ is far from μ_i^0 (either above or below μ_i^0) then this evidence casts doubt on the validity of the null hypothesis. Now, there is estimation error in (9.6) which is measured by:

$$\text{se}(\hat{\mu}_i) = \frac{\sigma_i}{\sqrt{T}}. \quad (9.7)$$

Because we assume σ_i is known, we also know $\text{se}(\hat{\mu}_i)$. Because of estimation error, we don't expect $\hat{\mu}_i$ to equal μ_i^0 even if the null hypothesis (9.3) is true. How far from μ_i^0 can $\hat{\mu}_i$ be if the null hypothesis is true? To answer this question, recall from Chapter 7 that the exact (finite sample) pdf of $\hat{\mu}_i$ is the normal distribution:

$$\hat{\mu}_i \sim N(\mu_i, \text{se}(\hat{\mu}_i)^2) = N\left(\mu_i, \frac{\sigma_i^2}{T}\right). \quad (9.8)$$

Under the null hypothesis (9.3), the normal distribution for $\hat{\mu}_i$ is centered at μ_i^0 :

$$\hat{\mu}_i \sim N(\mu_i^0, \text{se}(\hat{\mu}_i)^2) = N\left(\mu_i^0, \frac{\sigma_i^2}{T}\right). \quad (9.9)$$

From properties of the normal distribution we have:

$$Pr(\mu_i^0 - 1.96 \times \text{se}(\hat{\mu}_i) \leq \hat{\mu}_i \leq \mu_i^0 + 1.96 \times \text{se}(\hat{\mu}_i)) = 0.95. \quad (9.10)$$

Hence, if the null hypothesis (9.3) is true we don't expect to see $\hat{\mu}_i$ more than 1.96 values of $\text{se}(\hat{\mu}_i)$ away from μ_i^0 . Therefore, it makes intuitive sense to base a test statistic for testing (9.3) on a measure of distance between $\hat{\mu}_i$ and μ_i^0 relative to $\text{se}(\hat{\mu}_i)$. Such a statistic is the z-score:

$$z_{\mu=\mu^0} = \frac{\hat{\mu}_i - \mu_i^0}{\text{se}(\hat{\mu}_i)} = \frac{\hat{\mu}_i - \mu_i^0}{\sigma/\sqrt{T}}. \quad (9.11)$$

Assuming the null hypothesis (9.3) is true, from (9.9) it follows that:

$$z_{\mu=\mu^0} \sim N(0, 1).$$

The intuition for using the z-score (9.11) to test (9.3) is straightforward. If $z_{\mu=\mu^0} \approx 0$ then $\hat{\mu}_i \approx \mu_i^0$ and (9.3) should not be rejected. In contrast, if $z_{\mu=\mu^0} > 1.96$ or $z_{\mu=\mu^0} < -1.96$ then $\hat{\mu}_i$ is more than 1.96 values of $\text{se}(\hat{\mu}_i)$ away from μ_i^0 . From (9.10), this is very unlikely (less than 5% probability) if (9.3) is true. In this case, there is strong data evidence against (9.3) and we should reject it. Notice that the condition $z_{\mu=\mu^0} > 1.96$ or $z_{\mu=\mu^0} < -1.96$ can be simplified as the condition $|z_{\mu=\mu^0}| > 1.96$. Here, our test statistic for testing (9.3) against (9.4) is $S = |z_{\mu=\mu^0}|$ and the 5% critical value is $cv_{.05} = 1.96$.

More formally, the steps for using $S = |z_{\mu=\mu^0}|$ to test the (9.3) against (9.4) are:

1. Set the significance level $\alpha = Pr(\text{Reject } H_0 | H_0 \text{ is true})$ and determine the critical value cv_α such that $Pr(S > cv_\alpha) = \alpha$. Now,

$$Pr(S > cv_\alpha) = Pr(|z_{\mu=\mu^0}| > cv_\alpha) = Pr(z_{\mu=\mu^0} > cv_\alpha) + Pr(z_{\mu=\mu^0} < -cv_\alpha) = \alpha$$

which implies that $cv_\alpha = -q_{\alpha/2}^Z = q_{1-\alpha/2}^Z$, where $q_{\alpha/2}^Z$ denotes the $\frac{\alpha}{2}$ -quantile of $Z \sim N(0, 1)$. For example, if $\alpha = .05$ then $cv_{.05} = -q_{.025}^Z = 1.96$.

2. Reject (9.3) at the $\alpha \times 100\%$ significance level if $S > cv_\alpha$. For example, if $\alpha = .05$ then reject (9.3) at the 5% level if $S > 1.96$.
3. Equivalently, reject (9.3) at the $\alpha \times 100\%$ significance level if the p-value for S is less than α . Here, the p-value is defined as the significance level at which the test is just rejected. Let $Z \sim N(0, 1)$. The p-value is computed as:

$$\begin{aligned} \text{p-value} &= Pr(|Z| > S) = Pr(Z > S) + Pr(Z < -S) \\ &= 2 \times Pr(Z > S) = 2 \times (1 - Pr(Z < S)). \end{aligned}$$

Mention exact test when σ_i is unknown. The principle is the same but the distribution of the test statistic is different.

- We has the Student t distribution with $T - 1$ degrees of freedom

- As sample size goes to infinity the Student's t become the normal distribution. As the sample size gets larger $\hat{\sigma}_i$ gets close to σ_i and so the t-statistics get close to the z-score
- If $T \approx 60$ then $cv_{.05} = 2$ so we have a simple rule of thumb.

Comment on the exact tests based on the z-score and the t-statistic.

- We can derive an exact test for testing hypothesis about the value of μ_i based on the z-score () but we cannot derive similar exact tests for the values of σ_i or for the values of ρ_{ij} . Exact tests for these parameters are much more complicated
- while t-statistics for the values of σ_i or for the values of ρ_{ij} do not have exact t-distributions in finite samples, as the sample size gets large the distributions of the t-statistics get closer and closer to the normal distribution due to the CLT. This motivates the use of so-called asymptotic t-tests.

9.3.2 t-tests under asymptotic normality of estimators

Let $\hat{\theta}$ denote an estimator for θ . For example, in the CER model θ could be μ_i , σ_i , or ρ_{ij} . In many cases the CLT justifies the asymptotic normal distribution:

$$\hat{\theta} \sim N(\theta, \widehat{\text{se}}(\hat{\theta})^2), \quad (9.12)$$

for large enough sample size T , where $\widehat{\text{se}}(\hat{\theta})$ is the estimated standard error for $\hat{\theta}$. Consider testing:

$$H_0 : \theta = \theta_0 \text{ vs. } H_1 : \theta \neq \theta_0. \quad (9.13)$$

Under H_0 , the asymptotic normality result (9.12) implies that the t-statistic for testing (9.13) has a standard normal distribution for large enough sample size T :

$$t_{\theta=\theta_0} = \frac{\hat{\theta} - \theta_0}{\widehat{\text{se}}(\hat{\theta})} \sim N(0, 1) = Z. \quad (9.14)$$

The intuition for using the t-statistic (9.14) is straightforward. If $t_{\theta=\theta_0} \approx 0$ then $\hat{\theta} \approx \theta_0$, and $H_0 : \theta = \theta_0$ should not be rejected. On the other hand, if $|t_{\theta=\theta_0}| > 2$, say, then $\hat{\theta}$ is more than 2 values of $\widehat{\text{se}}(\hat{\theta})$ away from θ_0 . This is very unlikely if $\theta = \theta_0$ because $\hat{\theta} \sim N(\theta_0, \widehat{\text{se}}(\hat{\theta})^2)$, so $H_0 : \theta \neq \theta_0$ should be rejected.

The steps for using the t-statistic (9.14) with its critical value to test the hypotheses (9.13) are:

1. Set the significance level α ($\text{Pr}(\text{Type I error})$) of the test and determine the two-sided critical value $cv_{\alpha/2}$. Using (9.12), the critical value, $cv_{\alpha/2}$, is determined using:

$$\begin{aligned} \text{Pr}(|Z| \geq cv_{\alpha/2}) &= \alpha \\ \Rightarrow cv_{\alpha/2} &= -q_{\alpha/2}^Z = q_{1-\alpha/2}^Z, \end{aligned}$$

where $q_{\alpha/2}^Z$ denotes the $\frac{\alpha}{2}$ -quantile of $N(0, 1)$. A commonly used significance level is $\alpha = 0.05$ and the corresponding critical value is $cv_{.025} = -q_{.025}^Z = q_{.975}^Z = 1.96 \approx 2$.

2. Reject (9.13) at the $100 \times \alpha\%$ significance level if:

$$|t_{\theta=\theta_0}| = \left| \frac{\hat{\theta} - \theta^0}{\widehat{\text{se}}(\hat{\theta})} \right| > cv_{\alpha/2}.$$

If the significance level is $\alpha = 0.05$, then reject (9.13) at the 5% level using the rule-of-thumb:

$$|t_{\theta=\theta_0}| > 2.$$

The steps for using the t-statistic (9.14) with its p-value to test the hypotheses (9.13) are:

- Determine the two-sided p-value. The p-value of the two-sided test is the significance level at which the test is just rejected. From (9.12), the two-sided p-value is defined by

$$\text{p-value} = \Pr(|Z| > |t_{\theta=\theta_0}|) = 2 \times (1 - \Pr(Z \leq |t_{\theta=\theta_0}|)). \quad (9.15)$$

- Reject (9.13) at the $100 \times \alpha\%$ significance level if the p-value (9.15) is less than α .

Example 9.2. Using the t-statistic (9.14) to test hypothesis about μ in the CER model.

Consider using the t-statistic (9.14) to test $H_0 : \mu_i = 0$ vs. $H_1 : \mu_i \neq 0$ ($i = \text{Microsoft}$, Starbucks, S&P 500) using a 5% significance level. First, calculate the CER model estimates for μ_i :

```
n.obs = nrow(cerRetC)
muhat.vals = apply(cerRetC, 2, mean)
muhat.vals

##      MSFT      SBUX      SP500
## 0.00413 0.01466 0.00169
```

Next, calculate the estimated standard errors:

```
sigmahat.vals = apply(cerRetC, 2, sd)
se.muhat = sigmahat.vals/sqrt(n.obs)
se.muhat

##      MSFT      SBUX      SP500
## 0.00764 0.00851 0.00370
```

Then calculate the t-statistics:

```
t.stats = muhat.vals/se.muhat
abs(t.stats)

##  MSFT  SBUX  SP500
## 0.540 1.722 0.457
```

Since the absolute value of all of the t-statistics are less than two, we do not reject $H_0 : \mu_i = 0$ at the 5% level for all assets.

The p-values for all of the t-statistics are computed using:

```

2 * (1 - pnorm(abs(t.stats)))

##   MSFT    SBUX   SP500
## 0.5893 0.0851 0.6480

```

Since all p-values are greater than $\alpha = 0.05$, we reject $H_0 : \mu_i = 0$ at the 5% level for all assets. The p-value for Starbucks is the smallest at 0.0851. Here, we can reject $H_0 : \mu_{SBUX} = 0$ at the 8.51% level. ■

Example 9.3. Using the t-statistic (9.14) to test hypothesis about ρ in the CER model.

Consider using the t-statistic (9.14) to test:

$$H_0 : \rho_{ij} = 0.5 \text{ vs. } H_1 : \rho_{ij} \neq 0.5. \quad (9.16)$$

using a 5% significance level. Here, we use the result from the CER model that for large enough T :

$$\hat{\rho}_{ij} \sim N(\rho_{ij}, \widehat{se}(\hat{\rho}_{ij})), \widehat{se}(\hat{\rho}_{ij}) = \frac{1 - \hat{\rho}_{ij}^2}{\sqrt{T}}.$$

Then the t-statistic for testing (9.16) has the form:

$$t_{\rho_{ij}=0.5} = \frac{\hat{\rho}_{ij} - 0.5}{\widehat{se}(\hat{\rho}_{ij})} = \frac{\hat{\rho}_{ij} - 0.5}{(1 - \hat{\rho}_{ij}^2) / \sqrt{T}}. \quad (9.17)$$

To compute the t-statistics, first, calculate the CER model estimates for ρ_{ij} :

```

corhat.mat = cor(cerRetC)
rhohat.vals = corhat.mat[lower.tri(corhat.mat)]
names(rhohat.vals) = c("MSFT.SBUX", "MSFT.SP500", "SBUX.SP500")
rhohat.vals

##  MSFT.SBUX  MSFT.SP500  SBUX.SP500
##      0.341      0.617      0.457

```

Next, calculate estimated standard errors:

```

se.rhohat = (1 - rhohat.vals^2)/sqrt(n.obs)
se.rhohat

##  MSFT.SBUX  MSFT.SP500  SBUX.SP500
##      0.0674      0.0472      0.0603

```

Then calculate the t-statistics (9.17):

```

t.stats = (rhohat.vals - 0.5)/se.rhohat
abs(t.stats)

##  MSFT.SBUX  MSFT.SP500  SBUX.SP500
##      2.361      2.482      0.706

```

Here, the absolute value of the t-statistics for $\rho_{MSFT,SBUX}$ and $\rho_{MSFT,SP500}$ are greater than 2 whereas the absolute value of the t-statistic for $\rho_{SBUX,SP500}$ is less than 2. Hence,

for the pairs (MSFT, SBUX) and (MSFT, SP500) we cannot reject the null (9.16) at the 5% level but for the pair (SBUX, SP500) we can reject the null at the 5% level. The p-values for the test statistics are:

```
2 * (1 - pnorm(abs(t.stats)))

## MSFT.SBUX MSFT.SP500 SBUX.SP500
##      0.0182     0.0130     0.4804
```

Here, the p-values for the pairs (MSFT,SBUX) and (MSFT,SP500) are less than 0.05 and the p-value for the pair (SBUX, SP500) is much greater than 0.05.

9.3.3 Relationship between hypothesis tests and confidence intervals

Consider testing the hypotheses (9.13) at the 5% significance level. The rule-of-thumb decision rule is to reject $H_0 : \theta = \theta_0$ if $|t_{\theta=\theta_0}| > 2$. This implies that:

$$\frac{\hat{\theta} - \theta_0}{\widehat{\text{se}}(\hat{\theta})} > 2 \text{ or } \frac{\hat{\theta} - \theta_0}{\widehat{\text{se}}(\hat{\theta})} < -2,$$

which further implies that:

$$\theta_0 < \hat{\theta} - 2 \times \widehat{\text{se}}(\hat{\theta}) \text{ or } \theta_0 > \hat{\theta} + 2 \times \widehat{\text{se}}(\hat{\theta}).$$

Recall the definition of an approximate 95% confidence interval for θ :

$$\hat{\theta} \pm 2 \times \widehat{\text{se}}(\hat{\theta}) = [\hat{\theta} - 2 \times \widehat{\text{se}}(\hat{\theta}), \hat{\theta} + 2 \times \widehat{\text{se}}(\hat{\theta})].$$

Notice that if $|t_{\theta=\theta_0}| > 2$ then θ_0 does not lie in the approximate 95% confidence interval for θ . Hence, we can reject $H_0 : \theta = \theta_0$ at the 5% significance level if θ_0 does not lie in the 95% confidence for θ . This result allows us to have a deeper understanding of the 95% confidence interval for θ : it contains all values of θ_0 for which we cannot reject $H_0 : \theta = \theta_0$ at the 5% significance level.

This duality between two-sided hypothesis tests and confidence intervals makes hypothesis testing for individual parameters particularly simple. As part of estimation, we calculate estimated standard errors and form 95% confidence intervals. Then when we look at the 95% confidence interval, it gives us all values of θ_0 for which we cannot reject $H_0 : \theta = \theta_0$ at the 5% significance level.

Example 9.4. Hypothesis testing for μ_i and ρ_{ij} using 95% confidence intervals.

For the example data, the approximate 95% confidence intervals for μ_i for Microsoft, Starbucks and the S&P 500 index are:

```

lower = muhat.vals - 2 * se.muhat
upper = muhat.vals + 2 * se.muhat
cbind(lower, upper)

##           lower   upper
## MSFT    -0.01116 0.01941
## SBUX    -0.00237 0.03168
## SP500   -0.00570 0.00908

```

Consider testing $H_0 : \mu_i = 0$ at the 5% level. Here we see that $\mu_i^0 = 0$ lies in all of the 95% confidence intervals and so we do not reject $H_0 : \mu_i = 0$ at the 5% level for any asset. Each interval gives the values of μ_i^0 for which we cannot reject $H_0 : \mu_i = \mu_i^0$ at the 5% level. For example, for Microsoft we cannot reject the hypothesis (at the 5% level) that μ_{MSFT} is as small as -0.011 or as large as 0.019.

Next, the approximate 95% confidence intervals for ρ_{ij} for the pairs $\rho_{MSFT,SBUX}$, $\rho_{MSFT,SP500}$, and $\rho_{SBUX,SP500}$ are:

```

lower = rhohat.vals - 2*se.rhohat
upper = rhohat.vals + 2*se.rhohat
cbind(lower, upper)

@Consider testing $H_{\{0\}}:\rho_{ij}=0.5$ for all pairs of assets. Here,
we see that $\rho_{ij}^{(0)}=0.5$ is not in the approximate 95\% confidence
intervals for $\rho_{MSFT,SBUX}$ and $\rho_{MSFT,SP500}$ but is in
the approximate 95\% confidence interval for $\rho_{SBUX,SP500}$.
Hence, we can reject $H_{\{0\}}:\rho_{ij}=0.5$ at the 5\% significance
level for $\rho_{MSFT,SBUX}$ and $\rho_{MSFT,SP500}$ but not for
$\rho_{SBUX,SP500}$.

\section{Model Specification Tests}
\begin{itemize}
\item Review CER model assumptions
\item Discuss why it is interesting and useful to test these assumptions
\item What to do if certain assumptions are violated?
\end{itemize}

\subsection{Tests for normality}
\begin{itemize}
\item Why do we care about normality?

\begin{itemize}
\item Return distribution only depends on mean and volatility (and correlation in multivariate case).
\item Justifies volatility as an appropriate measure of risk.
\item Gives formula for Value-at-risk calculations.
\item Justifies mean-variance portfolio theory and Black-Scholes option pricing formula.
\item CER model parameter estimates and standard error formula do not require the data to be normally distributed. These estimates are consistent and asymptotically normally distributed for many non-normal distributions.
\item There may be estimators that are more accurate (i.e., have smaller standard errors)
\end{itemize}



```

```

\end{itemize}
\item What to do if returns are found to not be normally distributed?

\begin{itemize}
\item Return distribution characteristics in addition to mean and volatility may be important to investors. Volatility may not be the best measure of asset return risk. For example, skewness and kurtosis. If many returns are non-normal then this casts doubt on the appropriateness of mean-variance portfolio theory.
\item Standard practice is to compute VaR based on a normal distribution. If returns are not normally distributed this is an incorrect way to compute VaR. Instead of using the normal distribution quantile to compute VaR you can use the empirical quantile to compute VaR. If returns are substantially non-normal in the tails of the distribution then the empirical quantile will be more appropriate than the normal quantile for computing VaR
\item How are returns non-normal? This is important.
\end{itemize}
\end{itemize}
\begin{figure}
\noindent \begin{centering}
\includegraphics[width=0.55\textwidth]{116C__Users_ezivot_Dropbox_FinBook_lyx_svmono_figure_fig_HypothesisFourPanelMSFT}
\par\end{centering}
\end{figure}

\protect\caption{Four panel distribution plot for Microsoft monthly returns.}

\noindent \centering{}\label{fig:HypothesisFourPanelMSFT}
\end{figure}

In chapter \ref{chap:Descriptive-Statistics-for-Financial-Data}, we used histograms, normal QQ-plots, and boxplots to assess whether the normal distribution is a good characterization of the empirical distribution of daily and monthly returns. For example, Figure \ref{fig:HypothesisFourPanelMSFT} shows the four-panel plot for the monthly returns on Microsoft that we used to summarize the empirical distribution of returns. The histogram and smoothed histogram show that the empirical distribution of returns is bell-shaped like the normal distribution, whereas the boxplot and normal QQ-plot shows that returns have fatter tails than the normal distribution. Overall, the graphical descriptive statistics suggest that distribution of monthly Microsoft returns is not normally distributed. However, graphical diagnostics are not formal statistical tests. There are estimation errors in the statistics underlying the graphical diagnostics and it is possible that the observed non-normality of the data is due to sampling uncertainty. To be more definitive, we should use an appropriate test statistic to test the null hypothesis that the random return,  $R_{t}$ , is normally distributed. Formally, the null hypothesis to be tested is:
\begin{equation}
H_0: R_t \text{ is normally distributed} \label{eq:HT_H0normal}
\end{equation}
The alternative hypothesis is that  $R_t$  is not normally distributed:
\begin{equation}
H_1: R_t \text{ is not normally distributed} \label{eq:HT_H1nonnormal}
\end{equation}

```

```
\end{equation}
Here, we do not specify a particular distribution under the alternative
(e.g., a Student's t distribution with a 5 degrees of freedom). Our
goal is to have a test statistic that will reject the null hypothesis
(\ref{eq:HT_H0normal}) for a wide range of non-normal distributions.

## Error: <text>:5:1: unexpected '@'
## 4: cbind(lower, upper)
## 5: @
##   ^
```

JB test based on skewness and kurtosis

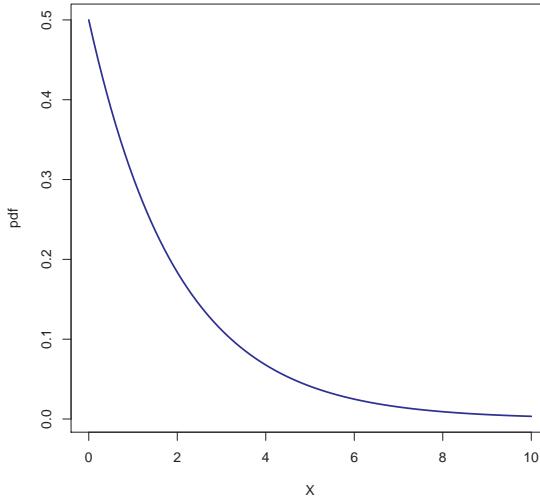


Fig. 9.2 pdf of $\chi^2(2)$

In Chapter 2, we showed that the skewness and kurtosis of a normally distributed random variable are equal to zero and three, respectively. We can use this information to develop a simple test of (??) against (??) based on testing the hypotheses:

$$H_0 : \text{skew}(R_t) = 0 \text{ and } \text{kurt}(R_t) = 3 \text{ vs.} \quad (9.18)$$

$$H_1 : \text{skew}(R_t) \neq 0 \text{ or } \text{kurt}(R_t) \neq 3 \text{ or both.} \quad (9.19)$$

To develop a test statistic for testing (9.18) against (9.19), Jarque and Bera (1981) used the result (see Kendall and Stuart, 1977) that if $\{R_t\} \sim \text{iid } N(\mu, \sigma^2)$ then by the CLT it can be shown that:

$$\begin{pmatrix} \widehat{\text{skew}}_r \\ \widehat{\text{kurt}}_r - 3 \end{pmatrix} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{6}{T} & 0 \\ 0 & \frac{24}{T} \end{pmatrix} \right),$$

for large enough T , where $\widehat{\text{skew}}_r$ and $\widehat{\text{kurt}}_r$ denote the sample skewness and kurtosis of the returns $\{r_t\}_{t=1}^T$, respectively. This motivates Jarque and Bera's *JB statistic*:

$$\text{JB} = T \times \left[\frac{(\widehat{\text{skew}}_r)^2}{6} + \frac{(\widehat{\text{kurt}}_r - 3)^2}{24} \right], \quad (9.20)$$

to test (??) against (??). If returns are normally distributed then $\widehat{\text{skew}}_r \approx 0$ and $\widehat{\text{kurt}}_r - 3 \approx 0$ so that $\text{JB} \approx 0$. In contrast, if returns are not normally distributed then $\widehat{\text{skew}}_r \neq 0$ or $\widehat{\text{kurt}}_r - 3 \neq 0$ or both so that $\text{JB} > 0$. Hence, if JB is big then there is data evidence against the null hypothesis that returns are normally distributed.

To determine the critical value for testing (??) at a given significance level we need the distribution of the JB statistic (9.20) under the null hypothesis (??). To this end, Jarque and Bera show, assuming (??) and large enough T , that:

$$\text{JB} \sim \chi^2(2),$$

where $\chi^2(2)$ denotes a chi-square distribution with two degrees of freedom (see Appendix). Hence, the critical values for the JB statistic are the upper quantiles of $\chi^2(2)$. Figure 9.2 shows the pdf of $\chi^2(2)$. The 90%, 95% and 99% quantiles can be computed using the R function `qchisq()`:

```
qchisq(c(0.9, 0.95, 0.99), 2)
## [1] 4.61 5.99 9.21
```

Since the 95% quantile of $\chi^2(2)$ is $5.99 \approx 6$, a simple rule of thumb is to reject the null hypothesis (??) at the 5% significance level if the JB statistic (9.20) is greater than 6.

Let $X \sim \chi^2(2)$. The p-value of the JB test is computed using:

$$p\text{-value} = Pr(X > \text{JB}).$$

If the p-value () is less than α then the null () is rejected at the

Example 9.5. Using the JB statistic to test the normality assumption for returns on Microsoft, Starbucks and the S&P 500 index.

The sample skewness and excess kurtosis values for the monthly returns on Microsoft, Starbucks, and the S&P 500 index are

```
library(PerformanceAnalytics)
skewhat = skewness(cerRetC)
ekurthat = kurtosis(cerRetC)
rbind(skewhat, ekurthat)
```

```

##          MSFT   SBUX  SP500
## Skewness     -0.0901 -0.987 -0.74
## Excess Kurtosis  2.0802  3.337  1.07

```

The sample skewness values for the assets are not too large but the sample excess kurtosis values are fairly large. The JB statistics for the three assets are easy to compute

```

n.obs = nrow(cerRetC)
JB = n.obs * ((as.numeric(skewwhat)^2)/6 + (as.numeric(ekurthat)^2)/24)
names(JB) = colnames(cerRetC)
JB

##  MSFT   SBUX  SP500
##  31.2 107.7 23.9

```

Since all JB values are greater than 6, we reject the null hypothesis that the monthly returns are normally distributed.

You can also use the **tseries** function **jarque.bera.test()** to compute the JB statistic:

```

library(tseries)
jarque.bera.test(msftRetC)

##
##  Jarque Bera Test
##
##  data: msftRetC
##  X-squared = 30, df = 2, p-value = 2e-07

jarque.bera.test(sbxuRetC)

##
##  Jarque Bera Test
##
##  data: sbxuRetC
##  X-squared = 100, df = 2, p-value <2e-16

jarque.bera.test(sp500RetC)

##
##  Jarque Bera Test
##
##  data: sp500RetC
##  X-squared = 20, df = 2, p-value = 7e-06

```

The test statistics are slightly different from the direct calculation above due to differences in how the sample skewness and sample kurtosis is computed within the function **jarque.bera.test()**.

Shapiro Wilks test

xx

9.3.4 Tests for no autocorrelation

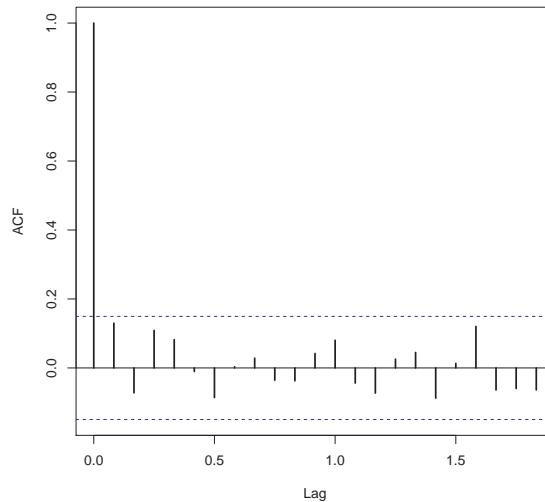


Fig. 9.3 SACF for monthly returns on the S&P 500 index.

A key assumption of the CER model is that returns are uncorrelated over time (not autocorrelated). In efficient financial markets where prices fully reflect true asset value, returns of actively traded assets should not be autocorrelated. In addition, if returns are autocorrelated then they are predictable. That is, future returns can be predicted from the past history of returns. If this is the case then trading strategies can be easily developed to exploit this predictability. But if many market participants act on these trading strategies then prices will adjust to eliminate any predictability in returns. Hence, for actively traded assets in well functioning markets we expect to see that the returns on these assets are uncorrelated over time.

For less frequently traded assets returns may appear to autocorrelated due to stale pricing.

As we saw in Chapter 6, this allows us to aggregate the CER model over time and, in particular, derive the square-root-of-time rule for aggregating return volatility. If returns are substantially correlated over time (i.e., autocorrelated) then the square-root-of-time rule will not be an accurate way to aggregate volatility.

To measure autocorrelation in asset returns $\{R_t\}$ we use the j^{th} lag autocorrelation coefficient

$$\rho_j = \text{cor}(R_t, R_{t-j}) = \frac{\text{cov}(R_t, R_{t-j})}{\text{var}(R_t)},$$

which is estimated using the sample autocorrelation

$$\hat{\rho}_j = \frac{\frac{1}{T-1} \sum_{t=j+1}^T (r_t - \hat{\mu})(r_{t-j} - \hat{\mu})}{\frac{1}{T-1} \sum_{t=1}^T (r_t - \hat{\mu})^2}.$$

In chapter 5, we used the sample autocorrelation function (SACF) to graphically evaluate the data evidence on autocorrelation for daily and monthly returns. For example, Figure 9.3 shows the SACF for the monthly returns on the S&P 500 index. Here we see that none of the $\hat{\rho}_j$ values are very large. We also see a pair of blue dotted lines on the graph. As we shall see, these dotted lines are related to critical values for testing the null hypothesis that $\rho_j = 0$. Hence, if

Individual tests

Individual tests for autocorrelation are similar to coefficient tests discussed earlier and are based on testing the hypotheses

$$H_0 : \rho_j = 0 \text{ vs. } H_1 : \rho_j \neq 0,$$

for a given lag j . The tests rely on the result, justified by the CLT if T is large enough, that if returns are covariance stationary and uncorrelated then

$$\hat{\rho}_j \sim N\left(0, \frac{1}{T}\right) \text{ for all } j \geq 1 \text{ and } \text{se}(\hat{\rho}_j) = \frac{1}{\sqrt{T}}.$$

This motivates the test statistics

$$t_{\rho_j=0} = \frac{\hat{\rho}_j}{\text{se}(\hat{\rho}_j)} = \frac{\hat{\rho}_j}{1/\sqrt{T}} = \sqrt{T}\hat{\rho}_j, j \geq 1$$

and the 95% confidence intervals

$$\hat{\rho}_j \pm 2 \cdot \frac{1}{\sqrt{T}}.$$

If we use a 5% significance level, then we reject the null hypothesis $H_0 : \rho_j = 0$ if

$$|t_{\rho_j=0}| = |\sqrt{T}\hat{\rho}_j| > 2.$$

That is, reject if

$$\hat{\rho}_j > \frac{2}{\sqrt{T}} \text{ or } \hat{\rho}_j < \frac{-2}{\sqrt{T}}.$$

Here we note that the dotted lines on the SACF are at the points $\pm 2 \cdot \frac{1}{\sqrt{T}}$. If the sample autocorrelation $\hat{\rho}_j$ crosses either above or below the dotted lines then we can reject the null hypothesis $H_0 : \rho_j = 0$ at the 5% level.

Joint tests

- Have to be careful how to interpret the individual tests. Cannot simply use 20 individual tests at the 5% level to test a joint hypothesis at the 5% level. With independent individual tests each test has a 5% probability of rejecting the null when it is true. So if you do 20 individual tests it would not be surprising if one tests rejects and this could be purely by chance (a type I error) if the null of no autocorrelation was true.

9.3.5 Tests for constant parameters

A key assumption of the CER model is that returns are generated from a covariance stationary time series process. As a result, all of the parameters of the CER model are constant over time. However, as shown in chapter xxx there is evidence to suggest that the assumption of covariance stationarity may not always be appropriate. For example, the volatility of returns for some assets does not appear to be constant over time.

A natural null hypothesis associated with the covariance stationarity assumption for the CER model for an individual asset return has the form:

$$H_0 : \mu_i, \sigma_i, \rho_{ij} \text{ are constant over time.}$$

To reject this null implied by covariance stationarity, one only needs to find evidence that at least one parameter is not constant over time.

Formal tests (two parameter structural change)

9.3.6 Informal diagnostics based on rolling estimators

A commonly used alternative to formal tests for constant parameters is to compute estimates of parameters over rolling sample windows of fixed length. To illustrate, suppose one is interested in determining if the expected return parameter μ is constant over time. Assume returns are observed over a sample of size T . An informal way to do this is to compute estimates of μ over rolling windows of length $n < T$

$$\hat{\mu}_t(n) = \frac{1}{n} \sum_{k=0}^{n-1} r_{t-k} = \frac{1}{n} (r_t + r_{t-1} + \dots + r_{t-n+1}),$$

for $t = n, \dots, T$. Here $\hat{\mu}_n(n)$ is the sample mean of the returns $\{r_t\}_{t=1}^n$ over the first sample window of size n . Then $\hat{\mu}_{n+1}(n)$ is the sample mean of the returns $\{r_t\}_{t=2}^{n+1}$ over the second sample window of size n . Continuing in this fashion we roll the sample window forward one observation at a time and compute $\hat{\mu}_t(n)$ until we arrive at the last window of size n . The end result is $T - n + 1$ rolling estimates $\{\hat{\mu}_n(n), \hat{\mu}_{n+1}(n), \dots, \hat{\mu}_T(n)\}$. If returns are covariance stationary then μ is constant over time and we would expect that $\hat{\mu}_n(n) \approx$

$\hat{\mu}_{n+1}(n) \approx \dots \approx \hat{\mu}_T(n)$. That is, the estimates of μ over each rolling window should be similar. We would not expect them to be exactly equal due to estimation error but they should not be drastically different.⁵ In contrast, if returns are not covariance stationary due to time variation in μ then we would expect to see meaningful time variation in the rolling estimates $\{\hat{\mu}_n(n), \hat{\mu}_{n+1}(n), \dots, \hat{\mu}_T(n)\}$. A natural way to see this time variation is to make a time series plot of these rolling estimates.

Example 9.6. Rolling estimates of μ for Microsoft, Starbucks and the S&P 500 Index

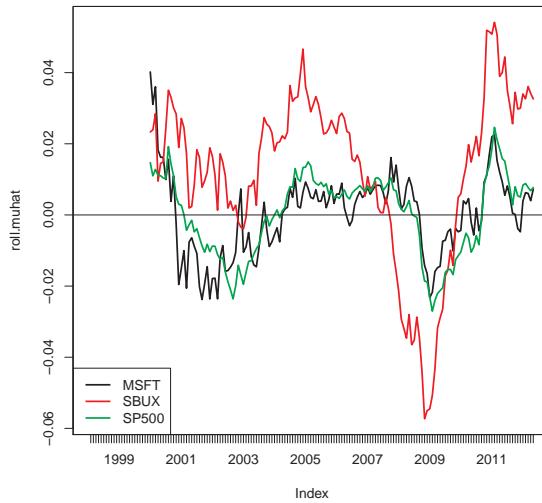


Fig. 9.4 24-month rolling estimates of μ

Consider computing rolling estimates of μ for Microsoft, Starbucks and the S&P 500 index returns. You can easily do this using the **zoo** function **rollapply()**, which applies a user specified function to a time series over rolling windows of fixed size. The arguments to **rollapply()** are

```
args(zoo:::rollapply.zoo)

## function (data, width, FUN, ..., by = 1, by.column = TRUE, fill = if (na.pad) NA,
##          na.pad = FALSE, partial = FALSE, align = c("center", "left",
##          "right"))
## NULL
```

where **data** is either a “**zoo**” or “**xts**” object, **width** is the window width, **FUN** is the R function to be applied over the rolling windows, **...** are any optional arguments to be supplied to **FUN**, **by** specifies how many observations to increment the rolling windows, **by.column** specifies if **FUN** is to be applied to each column of data or to be applied to all columns of data, and **align** specifies the location of the time index in the window. For

⁵ The estimated standard error of $\hat{\mu}_{i,t}(n)$ is $\hat{\sigma}_{i,t}(n)/\sqrt{n}$.

example, to compute 24-month rolling estimates of μ for Microsoft, Starbucks and the S&P 500 index incremented by one month with the time index aligned to the end of the window use

```
roll.muhat = rollapply(cerRetC, width = 24, by = 1, by.column = TRUE, FUN = mean,
  align = "right")
class(roll.muhat)

## [1] "xts" "zoo"

# first 24 values are NA
head(roll.muhat, n = 3)

##           MSFT   SBUX   SP500
## Feb 1998    NA     NA     NA
## Mar 1998    NA     NA     NA
## Apr 1998    NA     NA     NA

# first rolling estimate is for Jan 2000
head(na.omit(roll.muhat), n = 3)

##           MSFT   SBUX   SP500
## Jan 2000  0.0402 0.0233 0.0147
## Feb 2000  0.0311 0.0239 0.0110
## Mar 2000  0.0360 0.0284 0.0128

# total number of rolling estimates
nrow(na.omit(roll.muhat))

## [1] 149
```

The first rolling estimates are for the 24-months just prior to January, 2000 and the last rolling estimates are for the 24-months just prior to May 2012. The rolling means are shown in Figure 9.4 created using

```
plot.zoo(roll.muhat, plot.type = "single", lwd = 2, col = c("black", "red", "green"))
legend("bottomleft", legend = colnames(roll.muhat), lwd = 2, col = c("black", "red",
  "green"))
abline(h = 0)
```

All of the rolling estimates show time variation over the sample which places doubt on the assumption of covariance stationarity. Interestingly, the rolling estimates for the three assets exhibit a similar pattern of time variation. The estimates start positive at the end of the dot-com boom, decrease from 2000 to mid 2002 (Microsoft and the S&P 500 index become negative), increase from 2003 to 2006 during the booming economy, fall sharply during the financial crisis period 2007 to 2009, and increase from 2009 to 2011, and decrease afterward. The Microsoft and S&P 500 estimates follow each other very closely and the Starbucks estimates show the most variability. The largest rolling mean for Starbucks is 0.054 in February, 2011 and the smallest is -0.057 in November, 2008. The time variation exhibited by the rolling estimates places doubt on the covariance stationarity assumption of the CER model.

To investigate time variation in return volatility, we can compute rolling estimates of σ^2 and σ over fixed windows of size $n < T$:

$$\hat{\sigma}_t^2(n) = \frac{1}{n-1} \sum_{k=0}^{n-1} (r_{t-k} - \hat{\mu}_t(n))^2,$$

$$\hat{\sigma}_t(n) = \sqrt{\hat{\sigma}_t^2(n)}.$$

If return volatility is constant over time, then we would expect that $\hat{\sigma}_n(n) \approx \hat{\sigma}_{n+1}(n) \approx \dots \approx \hat{\sigma}_T(n)$. That is, the estimates of σ over each rolling window should be similar. In contrast, if volatility is not constant over time then we would expect to see meaningful time variation in the rolling estimates $\{\hat{\sigma}_n(n), \hat{\sigma}_{n+1}(n), \dots, \hat{\sigma}_T(n)\}$.

Example 9.7. Rolling estimates of σ for Microsoft, Starbucks and the S&P 500 Index

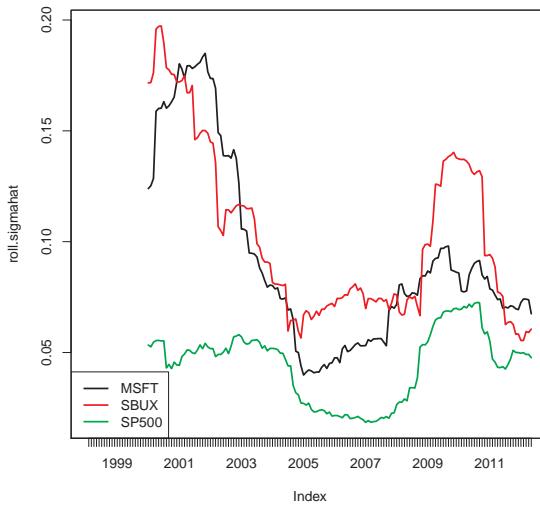


Fig. 9.5 24-month rolling estimates of σ .

To compute and plot 24-month rolling estimates of σ for Microsoft, Starbucks, and the S&P 500 index incremented by one month with the time index aligned to the beginning of the window use

```
roll.sigmahat = rollapply(cerRetC, width = 24, by = 1, by.column = TRUE, FUN = sd,
                           align = "right")
plot.zoo(roll.sigmahat, plot.type = "single", lwd = 2, col = c("black", "red",
                           "green"))
legend("bottomleft", legend = colnames(roll.muhat), lwd = 2, col = c("black", "red",
                           "green"))
```

The rolling estimates are illustrated in Figure 9.5. As with the rolling means, the rolling volatilities for the three assets show substantial and similar time variation over the sample. The rolling volatilities for Starbucks and Microsoft track each other closely. They start out high at the end of the dot-com boom, fall sharply until 2005 and then level off until the beginning of the financial crisis, increase during the financial crisis and fall afterward. This time variation exhibited by the rolling volatilities places additional doubt on the covariance stationarity assumption of the CER model.

■

Example 9.8. Rolling estimates with standard error bands

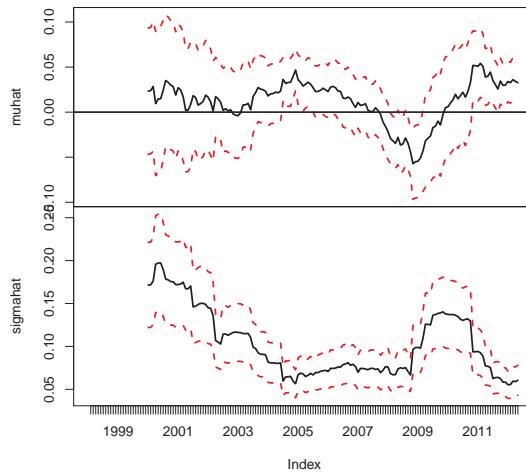


Fig. 9.6 24-month rolling estimates of μ and σ with standard error bands for Starbucks.

The rolling estimates of μ and σ shown in Figures 9.4 and 9.5 appear to show substantial time variation over the sample. However, one must always keep in mind that estimates have estimation error and part of the observed time variation is due to random estimation error. To account for estimation error, rolling estimates are often displayed with estimated standard error bands (i.e., 95% confidence intervals). Using results from chapter xxx, the estimated standard errors for $\hat{\mu}_n(n)$ and $\hat{\sigma}_t(n)$ are

$$\widehat{se}(\hat{\mu}_n(n)) = \frac{\hat{\sigma}_t(n)}{\sqrt{n}}, \quad \widehat{se}(\hat{\sigma}_t(n)) = \frac{\hat{\sigma}_t(n)}{\sqrt{2n}}.$$

The magnitude of the estimation error for the rolling estimates depends on the window width, n , and the estimated volatility in the window, $\hat{\sigma}_t(n)$. Figure 9.6 shows the rolling estimates of μ (top panel) and σ (bottom panel) for Starbucks together with 95% confidence intervals (standard error bands) computed using

```

se.muhat.SBUX = roll.sigmahat[, "SBUX"] / sqrt(24)
se.sigmahat.SBUX = roll.sigmahat[, "SBUX"] / sqrt(2 * 24)
lower.muhat.SBUX = roll.muhat[, "SBUX"] - 2 * se.muhat.SBUX
upper.muhat.SBUX = roll.muhat[, "SBUX"] + 2 * se.muhat.SBUX
lower.sigmahat.SBUX = roll.sigmahat[, "SBUX"] - 2 * se.sigmahat.SBUX
upper.sigmahat.SBUX = roll.sigmahat[, "SBUX"] + 2 * se.sigmahat.SBUX
# plot estimates with standard error bands
dataToPlot = merge(roll.muhat[, "SBUX"], lower.muhat.SBUX, upper.muhat.SBUX, roll.sigmahat[, "SBUX"], lower.sigmahat.SBUX, upper.sigmahat.SBUX)
plot.zoo(dataToPlot, main = "", screens = c(1, 1, 1, 2, 2, 2), lwd = 2, col = c("black", "red", "red", "black", "red", "red"), lty = c("solid", "dashed", "dashed", "solid", "dashed", "dashed"), panel = my.panel, ylab = list("muhat", "sigmahat"))

```

The standard error bands for $\hat{\mu}_n(n)$ are fairly wide, especially during the high volatility periods of the dot-com boom-bust and the financial crisis. During the dot-com period the 95% confidence intervals for μ are wide and contain both positive and negative values reflecting a high degree of uncertainty about μ . However, during the economic boom around 2005, where $\hat{\mu}_n(n)$ is large and $\hat{\sigma}_t(n)$ is small, the 95% confidence interval for μ does not contain zero and one can infer that the mean is significantly positive. Also, at the height of the financial crisis, where $\hat{\mu}_n(n)$ is the largest negative value, the 95% confidence interval for μ contains all negative values and one can infer that the mean is significantly negative. At the end of the sample, the 95% confidence intervals for μ contain all positive values.

The standard error bands for $\hat{\sigma}_t(n)$ are smaller than the bands for $\hat{\mu}_n(n)$ and one can see more clearly that $\hat{\sigma}_t(n)$ significantly larger during the dot-com period and the financial crisis period than during the economic boom period and the period after the financial crisis.

■

To investigate time variation in the correlation between two asset returns, ρ_{ij} , one can compute the rolling covariances and correlations

$$\hat{\sigma}_{ij,t}(n) = \frac{1}{n-1} \sum_{k=0}^{n-1} (r_{i,t-k} - \hat{\mu}_i(n))(r_{j,t-k} - \hat{\mu}_j(n)),$$

$$\hat{\rho}_{ij,t}(n) = \frac{\hat{\sigma}_{ij,t}(n)}{\hat{\sigma}_{i,t}(n)\hat{\sigma}_{j,t}(n)}.$$

Example 9.9. Rolling estimates of ρ_{ij} for Microsoft, Starbucks, and the S&P 500 index

A function to compute the pair-wise sample correlations between Microsoft, Starbucks and the S&P 500 index is

```

rhohat = function(x) {
  corhat = cor(x)
  corhat.vals = corhat[lower.tri(corhat)]
  names(corhat.vals) = c("MSFT.SBUX", "MSFT.SP500", "SBUX.SP500")
  corhat.vals
}

```

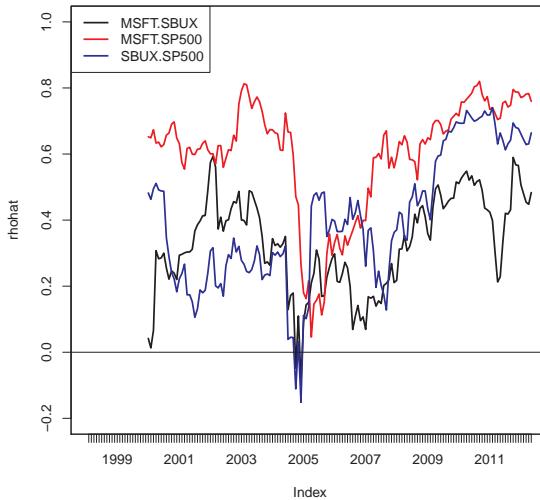


Fig. 9.7 24-month rolling estimates of ρ_{ij} for Microsoft, Starbucks and the S&P 500 index.

Using the function `rhohat()`, 24-month rolling estimates of ρ_{ij} for Microsoft, Starbucks and the S&P 500 index incremented by one month with the time index aligned to the beginning of the window can be computed using

```
roll.rhohat = rollapply(cerRetC, width = 24, FUN = rhohat, by = 1, by.column = FALSE,
                        align = "right")
head(na.omit(roll.rhohat), n = 3)

##          MSFT.SBUX MSFT.SP500 SBUX.SP500
## Jan 2000    0.0418     0.652     0.482
## Feb 2000    0.0130     0.649     0.463
## Mar 2000    0.0660     0.673     0.495
```

Notice that `by.column=FALSE` in the call to `rollapply()`. This is necessary as the function `rhohat()` operates on all columns of the data object `cerRetC`. The rolling estimates are illustrated in Figure 9.7 created using

```
plot.zoo(roll.rhohat, plot.type = "single", ylab = "rhohat", ylim = c(-0.2, 1),
         lwd = 2, col = c("black", "red", "blue"))
abline(h = 0)
legend("topleft", legend = colnames(roll.rhohat), lwd = 2, col = c("black", "red",
                     "blue"))
```

Practical issues associated with rolling estimates

- discuss choice of window width. Small window widths tend to produce choppy estimates with a lot of estimation error. Large window widths produce smooth estimates with less estimation error

- Discuss choice of increment. For a given window width, the roll-by increment determines the number of overlapping returns in each window. Number of overlapping returns is $n - incr$ where n is the window width and $incr$ is the roll-by increment. Incrementing the window by 1 gives the largest number of overlapping returns. If $incr \geq n$ then the rolling windows are non-overlapping.
- A large positive or negative return in a window can greatly influence the estimate and cause a drop-off effect as the window rolls past the large observation.

9.3.7 Informal diagnostics based on exponentially weighted moving averages

- Discuss advantages of EWMA estimates

9.4 Further Reading

- Mention books by Carol Alexander for discussion of rolling and EWMA estimates
- Formal tests for structural change require advanced probability and statistics. See Zivot (2016) for discussion. Many tests are implemented in the strucchange package.

9.5 Problems

TO BE COMPLETED

References

1. Campbell, Lo and MacKinley (1998). *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ.
2. Zivot, E. (2016). *Modeling Financial Time Series with R*. Springer-Verlag, forthcoming.

Chapter 10

Modeling Daily Returns with the GARCH Model

Updated: April 15, 2016

Copyright © Eric Zivot 2016

Outline

1. Motivation

- a. Review stylized facts of daily return data and contrast with monthly returns
- b. Modeling daily returns is most useful for short term risk analysis of assets and portfolios using volatility and VaR
- c. CER model captures most stylized facts of monthly returns but not daily returns
- d. Need a model for time varying volatility
- e. Engle's ARCH model captures stylized facts of daily returns and he was awarded Nobel prize in economics partially for this model.

2. Engle's ARCH Model

- a. Start with ARCH(1) process
- b. State assumptions and derive properties
- c. Show simulated values and do reality check against actual data
- d. Introduce rugarch R package
- e. General ARCH(p) process

3. Bollerslev's GARCH(1,1) process

- a. No need to go beyond GARCH(1,1)
- b. Give assumptions and derive basic properties
- c. Show simulated values and do reality check against actual data
- d. GARCH(1,1) with Student t errors
- e. Show simulated values and do reality check against actual data

4. Maximum Likelihood Estimation

- a. Overview of the technique of MLE

- b. Illustration with CER model and GARCH(1,1)
 - c. Numerical optimization
 - d. Asymptotic properties
 - e. Calculating standard errors
5. Maximum likelihood estimation of ARCH and GARCH models
- a. GARCH(1,1) log-likelihood function
6. Forecasting Conditional Volatility from GARCH(1,1)
- a. Forecasting algorithm
 - b. Multiperiod
7. Conditional VaR
- a. unconditional vs. conditional VaR
 - b. 1-day ahead VaR forecast
 - c. h-day ahead VaR forecast

In chapter xxx (Descriptive statistics) it was shown that daily asset returns have some features in common with monthly asset returns and some not. In particular, daily returns have empirical distributions with much fatter tails than the normal distribution and daily returns are not independent over time. Absolute and squared returns are positively autocorrelated and the correlation dies out very slowly. Daily volatility appears to be autocorrelated and, hence, predictable. In this chapter we present a model of daily asset returns, Robert Engle's *autoregressive conditional heteroskedasticity* (ARCH) model, that can capture these stylized facts that are specific to daily returns. The ARCH model is one of the most important models in the field of financial econometrics, and its creator Robert Engle won the Nobel Prize in Economics in part for his work on the ARCH model and its variants.

The R packages used in this chapter are **IntroCompFinR**, **rugarch** and **xts**. Make sure these package are installed and loaded before running the R examples in this chapter.

- Why do we care about time varying volatility? Volatility is a standard measure of asset risk and if volatility varies over time then asset risk also varies over time. In periods of low volatility, asset risk is low and in periods of high volatility asset risk is high. Time varying volatility also impacts the risk measure value-at-risk (VaR), which gives the dollar loss that could occur over a period of time with a given probability. Volatility is an important driver for the price of call and put options as well as other derivative securities. That volatility is predictable has important implications for accurate pricing models for options and derivatives.
- Why do we care about the predictability of volatility? We can forecast future values of volatility. This allows us to forecast VaR on an asset or portfolio.

10.1 Engle's ARCH Model

- Goal: create a simple time series model that captures the basic stylized facts of daily return data
- Foundation of the field of financial econometrics

10.1.1 The ARCH(1) Model

Let R_t denote the continuously compounded daily return on an asset. The CER model for R_t can be expressed the form:

$$R_t = \mu + \epsilon_t, \quad t = 1, \dots, T, \quad (10.1)$$

$$\epsilon_t = \sigma z_t, \quad (10.2)$$

$$z_t \sim iid N(0, 1). \quad (10.3)$$

Here, the unexpected return ϵ_t is expressed as σz_t where σ is the unconditional volatility of ϵ_t , which is assumed to be constant, and $z_t = \epsilon_t/\sigma$ is the standardized unexpected return. In (10.1)-(10.3), $R_t \sim iid N(\mu, \sigma^2)$ which is not a good model for daily returns given the stylized facts. The ARCH(1) model for R_t has a similar form:

$$R_t = \mu + \epsilon_t, \quad t = 1, \dots, T, \quad (10.4)$$

$$\epsilon_t = \sigma_t z_t, \quad (10.5)$$

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2, \quad (10.6)$$

$$z_t \sim iid N(0, 1), \quad (10.7)$$

$$\omega > 0, \quad 0 \leq \alpha_1 < 1. \quad (10.8)$$

In the ARCH(1), the unexpected return ϵ_t is expressed as $\sigma_t z_t$ where σ_t is the conditional (time t dependent) volatility of ϵ_t . Hence, the ARCH(1) model extends the CER model to allow for time varying conditional volatility σ_t .

In the ARCH(1) model (10.4) - (10.8), the restrictions $\omega > 0$ and $0 \leq \alpha_1 < 1$ are required to ensure that $\sigma_t^2 > 0$ and that $\{R_t\}$ is a covariance stationary time series. Equation (10.6) shows that the return variance at time t , σ_t^2 is a positive linear function of the squared unexpected return at time $t-1$, ϵ_{t-1}^2 . This allows the magnitude of yesterday's return to influence today's return variance and volatility. In particular, a large (small) value of ϵ_{t-1}^2 leads to a large (small) value of σ_t^2 . This feedback from ϵ_{t-1}^2 to σ_t^2 can explain some of the volatility clustering observed in daily returns.

Statistical properties of the ARCH(1) Model

Let $I_t = \{R_t, R_{t-1}, \dots, R_1\}$ denote the information set at time t (conditioning set of random variables) as described in Chapter xxx (Time Series Concepts). As we shall show below, the ARCH(1) model (10.4) - (10.8) has the following statistical properties:

1. $E(\epsilon_t|I_t) = 0, E(\epsilon_t) = 0.$
2. $\text{var}(R_t|I_t) = E(\epsilon_t^2|I_t) = \sigma_t^2.$
3. $\text{var}(R_t) = E(\epsilon_t^2) = E(\sigma_t^2) = \omega/(1 - \alpha_1).$
4. $\{R_t\}$ is an uncorrelated process: $\text{cov}(R_t, R_{t-k}) = E(\epsilon_t \epsilon_{t-j}) = 0$ for $k > 0$.
5. The distribution of R_t conditional on I_{t-1} is normal with mean μ and variance σ_t^2 .
6. The unconditional (marginal) distribution of R_t is not normal and $\text{kurt}(R_t) \geq 3$.
7. $\{R_t^2\}$ and $\{\epsilon_t^2\}$ have a covariance stationary AR(1) model representation. The persistence of the autocorrelations is measured by α_1 .

These properties of the ARCH(1) model match many of the stylized facts of daily asset returns.

It is instructive to derive each of the above properties as the derivations make use of certain results and tricks for manipulating conditional expectations. First, consider the derivation for property 1. Because $\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2$ depends on information dated $t-1$, $\sigma_t^2 \in I_{t-1}$ and so σ_t^2 may be treated as a constant when conditioning on I_{t-1} . Hence,

$$E(\epsilon_t|I_t) = E(\sigma_t z_t|I_{t-1}) = \sigma_t E(z_t|I_{t-1}) = 0.$$

The last equality follows from that fact that $z_t \sim iid N(0, 1)$ which implies that z_t is independent of I_{t-1} and so $E(z_t|I_{t-1}) = E(z_t) = 0$. By iterated expectations, it follows that

$$E(\epsilon_t) = E(E(\epsilon|I_{t-1})) = E(0) = 0.$$

The derivation of the second property uses similar computations:

$$\text{var}(R_t|I_t) = E((R_t - \mu)^2|I_{t-1}) = E(\epsilon_t^2|I_t) = E(\sigma_t^2 z_t^2|I_{t-1}) = \sigma_t^2 E(z_t^2|I_{t-1}) = \sigma_t^2,$$

where the last equality uses the result $E(z_t^2|I_{t-1}) = E(z_t^2) = 1$.

To derive property 3, first note that by iterated expectations and property 2

$$\text{var}(R_t) = E(\epsilon_t^2) = E(E(\epsilon_t^2|I_t)) = E(\sigma_t^2).$$

Next, using (10.6) we have that

$$E(\sigma_t^2) = \omega + \alpha_1 E(\epsilon_{t-1}^2).$$

Now use the fact that $\{R_t\}$, and hence $\{\epsilon_t\}$, is covariance stationary which implies that $E(\epsilon_t^2) = E(\epsilon_{t-1}^2) = E(\sigma_t^2)$ so that

$$E(\sigma_t^2) = \omega + \alpha_1 E(\sigma_t^2).$$

Solving for $E(\sigma_t^2)$ then gives

$$\text{var}(R_t) = E(\sigma_t^2) = \frac{\omega}{1 - \alpha_1}.$$

For the fourth property, first note that

$$\text{cov}(R_t, R_{t-k}) = E((R_t - \mu)(r_{t-j} - \mu)) = E(\epsilon_t \epsilon_{t-k}) = E(\sigma_t z_t \sigma_{t-j} z_{t-j}).$$

Next, by iterated expectations and the fact that $z_t \sim \text{iid } N(0, 1)$ we have

$$E(\sigma_t z_t \sigma_{t-j} z_{t-j}) = E(E(\sigma_t z_t \sigma_{t-j} z_{t-j} | I_{t-1})) = E(\sigma_t \sigma_{t-j} z_{t-j} E(z_t | I_{t-1})) = 0,$$

and so $\text{cov}(R_t, R_{t-k}) = 0$.

To derive the fifth property, write $R_t = \mu + \epsilon_t = \mu + \sigma_t z_t$. Then, conditional on I_{t-1} and the fact that $z_t \sim \text{iid } N(0, 1)$ we have

$$R_t | I_{t-1} \sim N(\mu, \sigma_t^2).$$

The sixth property has two parts. To see that the unconditional distribution of R_t is not a normal distribution, it is useful to express R_t in terms of current and past values of z_t . Start with

$$R_t = \mu + \sigma_t z_t.$$

From (10.6), $\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 = \omega + \alpha_1 \sigma_{t-1}^2 z_{t-1}^2$ which implies that $\sigma_t = (\omega + \alpha_1 \sigma_{t-1}^2 z_{t-1}^2)^{1/2}$. Then

$$R_t = \mu + (\omega + \alpha_1 \sigma_{t-1}^2 z_{t-1}^2)^{1/2} z_t.$$

Even though $z_t \sim \text{iid } N(0, 1)$, R_t is a complicated nonlinear function of z_t and past values of z_t^2 and so R_t cannot be a normally distributed random variable. Next, to see that $\text{kurt}(R_t) \geq 3$ first note that

$$\text{kurt}(R_t) = \frac{E((R_t - \mu)^4)}{\text{var}(R_t)^2} = \frac{E(\epsilon_t^4)}{\omega^2 / (1 - \alpha_1)^2} \quad (10.9)$$

Now by iterated expectations and the fact that $E(z_t^4) = 3$

$$E(\epsilon_t^4) = E(E(\sigma_t^4 z_t^4 | I_{t-1})) = E(\sigma_t^4 E(z_t^4 | I_{t-1})) = 3E(\sigma_t^4).$$

Next, using

$$\begin{aligned} \sigma_t^4 &= (\sigma_t^2)^2 = (\omega + \alpha_1 \epsilon_{t-1}^2)^2 \\ &= \omega^2 + 2\omega\alpha_1 \epsilon_{t-1}^2 + \alpha_1^2 \epsilon_{t-1}^4 \end{aligned}$$

we have

$$\begin{aligned} E(\epsilon_t^4) &= 3E(\omega^2 + 2\omega\alpha_1\epsilon_{t-1}^2 + \alpha_1^2\epsilon_{t-1}^4) \\ &= 3\omega^2 + 6\omega\alpha_1E(\epsilon_{t-1}^2) + 3\alpha_1^2E(\epsilon_{t-1}^4). \end{aligned}$$

Since $\{R_t\}$ is covariance stationary it follows that $E(\epsilon_{t-1}^2) = E(\epsilon_t^2) = E(\sigma_t^2) = \omega/(1 - \alpha_1)$ and $E(\epsilon_{t-1}^4) = E(\epsilon_t^4)$. Hence

$$\begin{aligned} E(\epsilon_t^4) &= 3\omega^2 + 6\omega\alpha_1(\omega/(1 - \alpha_1)) + 3\alpha_1^2E(\epsilon_t^4) \\ &= 3\omega^2 \left(1 + 2\frac{\alpha_1}{1 - \alpha_1}\right) + 3\alpha_1^2E(\epsilon_t^4). \end{aligned}$$

Solving for $E(\epsilon_t^4)$ gives

$$E(\epsilon_t^4) = \frac{3\omega^2(1 + \alpha_1)}{(1 - \alpha_1)(1 - 3\alpha_1^2)}.$$

Because $0 \leq \alpha_1 < 1$, in order for $E(\epsilon_t^4) < \infty$ it must be the case that $1 - 3\alpha_1^2 > 0$ which implies that $0 \leq \alpha_1^2 < \frac{1}{3}$ or $0 \leq \alpha_1 < 0.577$. Substituting the above expression for $E(\epsilon_t^4)$ into (10.9) then gives

$$\begin{aligned} \text{kurt}(R_t) &= \frac{3\omega^2(1 + \alpha_1)}{(1 - \alpha_1)(1 - 3\alpha_1^2)} \times \frac{(1 - \alpha_1)^2}{\omega^2} \\ &= 3\frac{1 - \alpha_1^2}{1 - 3\alpha_1^2} \geq 3. \end{aligned}$$

Hence, if returns follow an ARCH(1) process with $\alpha_1 > 0$ then $\text{kurt}(R_t) > 3$ which implies that the unconditional distribution of returns has fatter tails than the normal distribution.

To show the last property, add ϵ_{t-1}^2 to both sides of (10.6) to give

$$\begin{aligned} \epsilon_t^2 + \sigma_t^2 &= \omega + \alpha_1\epsilon_{t-1}^2 + \epsilon_t^2 \Rightarrow \\ \epsilon_t^2 &= \omega + \alpha_1\epsilon_{t-1}^2 + \epsilon_t^2 - \sigma_t^2 \\ &= \omega + \alpha_1\epsilon_{t-1}^2 + v_t, \end{aligned}$$

where $v_t = \epsilon_t^2 - \sigma_t^2$ is a mean-zero and uncorrelated error term. Hence, ϵ_t^2 and R_t^2 follow an AR(1) process with positive autoregressive coefficient α_1 . The autocorrelations of R_t^2 are

$$\text{cor}(R_t^2, R_{t-j}^2) = \alpha_1^j \text{ for } j > 1,$$

which are positive and decay toward zero exponentially fast as j gets large. Here, the persistence of the autocorrelations is measured by α_1 .

Example 10.1. Simulate an ARCH(1) model using brute force

To be completed

■

The rugarch package

- Give a brief introduction to the rugarch package

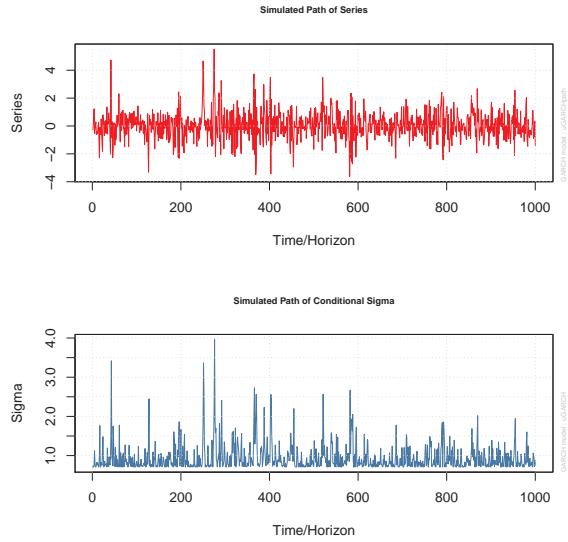


Fig. 10.1 Simulated values from ARCH(1) process. Top panel: simulated values of R_t . Bottom panel: simulated values of σ_t .

Example 10.2. Simulate an ARCH(1) model using **rugarch**

Consider the simple ARCH(1) model for returns

$$\begin{aligned} R_t &= \varepsilon_t = \sigma_t z_t, \quad z_t \sim iid \ N(0, 1) \\ \sigma_t^2 &= 0.5 + 0.5R_{t-1}^2 \end{aligned}$$

Here, $\alpha_1 = 0.5 < 1$ so that R_t is covariance stationary and $\bar{\sigma}^2 = \omega/(1-\alpha_1) = 0.5/(1-0.5) = 1$. Using (10.9),

$$\text{kurt}(R_t) = 3 \frac{1 - \alpha_1^2}{1 - 3\alpha_1^2} = 3 \frac{1 - (0.5)^2}{1 - 3(0.5)^2} = 9,$$

so that the distribution of R_t has much fatter tails than the normal distribution.

The **rugarch** functions **ugarchspec()** and **ugarchpath()** can be used to simulate $T = 1000$ values of R_t and σ_t from this model.¹ The ARCH(1) model is specified using the **ugarchpath()** function as follows

```
library(rugarch)
library(PerformanceAnalytics)
arch1.spec = ugarchspec(variance.model = list(garchOrder = c(1, 0)), mean.model = list(armaOrder = c(0, 0)), fixed.pars = list(mu = 0, omega = 0.5, alpha1 = 0.5))
```

The argument **fixed.pars** is a list whose components give the ARCH(1) model parameters. The names of the list components match the parameters from the ARCH(1) model: **mu**

¹ More details on using the **rugarch** functions are given later in the chapter.

is μ , ω is ω , and alpha1 is α_1 . Simulated values of R_t and σ_t are produced using the `ugarchpath()` function taking as input the “`uGARCHspec`” object `arch1.spec` and the number of simulations, `n.sim=1000`, to produce:

```
set.seed(123)
arch1.sim = ugarchpath(arch1.spec, n.sim = 1000)
class(arch1.sim)

## [1] "uGARCHpath"
## attr(,"package")
## [1] "rugarch"

slotNames(arch1.sim)

## [1] "path"   "model"  "seed"

names(arch1.sim@path)

## [1] "sigmaSim" "seriesSim" "residSim"
```

The object `arch1.sim` is an `Sv4` object of class `uGARCHpath` for which there are `sigma`, `fitted`, `quantile`, `show` and `plot` methods.² The `path` slot is a list which contains the simulated values of σ_t , R_t and $\varepsilon_t = R_t - \mu_t$ as `matrix` objects. The method functions `sigma()` and `fitted()` extract σ_t and μ_t , respectively. Invoking the `plot()` method produces a menu of plot choices

show plot menu here

Individual plots can be produced directly by specifying the plot number in the call to `plot()`. For example, Figure 10.1 shows the plots of the simulated values for R_t and σ_t created with

```
par(mfrow = c(2, 1))
plot(arch1.sim, which = 2)
plot(arch1.sim, which = 1)
par(mfrow = c(1, 1))
```

Figure 10.2 shows the sample autocorrelations of R_t and R_t^2 . As expected returns are uncorrelated whereas R_t^2 has autocorrelations described by an AR(1) process with positive autoregressive coefficient.

The sample variance and excess kurtosis of the simulated returns are

```
as.numeric(var(arch1.sim@path$seriesSim))

## [1] 0.958

kurtosis(arch1.sim@path$seriesSim)

## [1] 2.67
```

² In R there are two main object types: `Sv3` and `Sv4`. The main operational difference between `Sv3` and `Sv4` objects is that `Sv4` components are extracted using the `@` character instead of the `$` character. Also, the names of `Sv4` objects are extracted using the `slotNames()` function instead of the `names()` function.

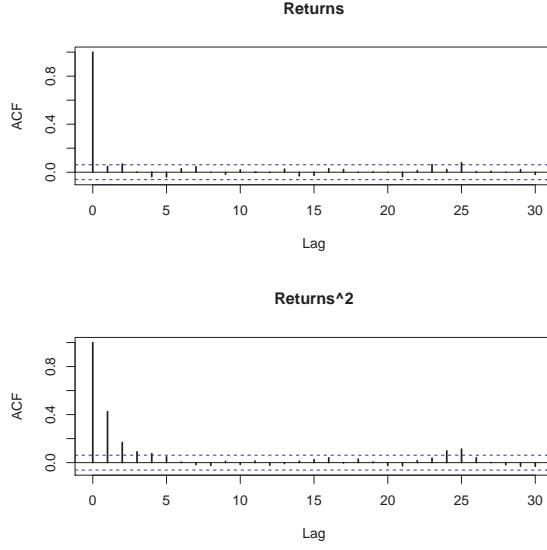


Fig. 10.2 SACFs for R_t (top panel) and R_t^2 (bottom panel) from simulated ARCH(1) model.

The sample variance is very close to the unconditional variance $\bar{\sigma}^2 = 0.2$, and the sample excess kurtosis is very close to the ARCH(1) excess kurtosis of 6.

■

10.1.2 The ARCH(p) Model

The ARCH(p) model extends the autocorrelation structure of R_t^2 and ϵ_t^2 in the ARCH(1) model 10.4() - (10.6) to that of an AR(p) process by adding p lags of ϵ_t^2 to the dynamic equation for σ_t^2 :

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_p \epsilon_{t-p}^2, \quad (10.10)$$

In order for σ_t^2 to be positive we need to impose the restrictions

$$\omega > 0, \alpha_1 \geq 0, \alpha_2 \geq 0, \dots, \alpha_p \geq 0. \quad (10.11)$$

In addition, for $\{R_t\}$ to be a covariance stationary time series we must have the restriction

$$0 \leq \alpha_1 + \alpha_2 + \cdots + \alpha_p < 1. \quad (10.12)$$

The statistical properties of the ARCH(p) model are the same as those for the ARCH(1) model with the following exceptions. The unconditional variance of $\{R_t\}$ is

$$\text{var}(R_t) = E(\epsilon_t^2) = E(\sigma_t^2) = \omega / (1 - \alpha_1 - \alpha_2 - \cdots - \alpha_p), \quad (10.13)$$

and $\{R_t^2\}$ and $\{\epsilon_t^2\}$ have a covariance stationary AR(p) model representation whose autocorrelation persistence is measured by the sum of the ARCH coefficients $\alpha_1 + \alpha_2 + \cdots + \alpha_p$.

The ARCH(p) model is capable of creating a much richer autocorrelation structure for R_t^2 . In the ARCH(1) model, the autocorrelations of R_t^2 decay to zero fairly quickly whereas the sample autocorrelations shown in Figure xxx decay to zero very slowly. In the ARCH(p) model, because of the restrictions (10.11) and (10.12), for large values of p the dynamics of σ_t^2 from () can more closely mimic the observed autocorrelations of actual daily returns.

Example 10.3. Simulate ARCH(p) model using **rugarch**

To be completed. Simulate long order AR(10) to show difference with ARCH(1). Make sum of ARCH coefficients equal to 0.9

■

10.2 Bollerslev's GARCH Model

An important extension of the ARCH(p) model proposed by Bollerslev (1986) replaces the AR(p) representation of σ_t^2 in (10.10) with an ARMA(p, q) formulation giving the GARCH(p, q) model:

$$\sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 \quad (10.14)$$

In order for $\sigma_t^2 > 0$, it is assumed that $\omega > 0$ and the coefficients α_i ($i = 0, \dots, p$) and β_j ($j = 1, \dots, q$) are all non-negative.³ When $q = 0$, the GARCH(p, q) model (10.14) reduces to the ARCH(p) model. In general, the GARCH(p, q) model can be shown to be equivalent to a particular ARCH(∞) model.

Usually the GARCH(1,1) model,

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \quad (10.15)$$

with only three parameters in the conditional variance equation is adequate to obtain a good model fit for daily asset returns. Indeed, Hansen and Lund (2004) provided compelling evidence that is difficult to find a volatility model that outperforms the simple GARCH(1,1). Hence, for many purposes the GARCH(1,1) model is the *de facto* volatility model of choice for daily returns.

³ Positive coefficients are sufficient but not necessary conditions for the positivity of conditional variance. See [?] and [?] for more general conditions.

10.2.1 Statistical Properties of the GARCH(1,1) Model

The statistical properties of the GARCH(1,1) model are derived in the same way as the properties of the ARCH(1) model and are summarized below:

1. $\{R_t\}$ is a covariance stationary and ergodic process provided $\alpha_1 + \beta_1 < 1$.
2. $\text{var}(R_t) = E(\varepsilon_t^2) = E(\sigma_t^2) = \omega/(1 - \alpha_1 - \beta_1) = \bar{\sigma}^2$.
3. The distribution of R_t conditional on I_{t-1} is normal with mean μ and variance σ_t^2 .
4. The unconditional (marginal) distribution of R_t is not normal and

$$\text{kurt}(R_t) = \frac{3(1 + \alpha_1 + \beta_1)(1 - \alpha_1 - \beta_1)}{1 - 2\alpha_1\beta_1 - 3\alpha_1^2 - \beta_1^2} \geq 3.$$

5. $\{R_t^2\}$ and $\{\varepsilon_t^2\}$ have an ARMA(1,1) representation

$$\varepsilon_t^2 = \omega + (\alpha_1 + \beta_1)\varepsilon_{t-1}^2 + u_t - \beta_1 u_{t-1},$$

with $u_t = \varepsilon_t^2 - \sigma_t^2$. The persistence of the autocorrelations is given by $\alpha_1 + \beta_1$.

6. σ_t^2 has an AR(∞) representation

$$\sigma_t^2 = \frac{\omega}{1 - \beta_1} + \alpha_1 \sum_{j=0}^{\infty} \beta_1^j \varepsilon_{t-j-1}^2.$$

The derivations of these properties are left as end-of-chapter exercise.

- Need some comments here: more flexible autocorrelations structure for R_t^2 than ARCH(p) and with fewer parameters.

Example 10.4. Simulate GARCH(1,1) model using **rugarch**

Consider simulating $T = 1000$ observations from the GARCH(1,1) model

$$\begin{aligned} R_t &= \varepsilon_t = \sigma_t z_t, \quad z_t \sim \text{iid } N(0, 1) \\ \sigma_t^2 &= 0.01 + 0.07 R_{t-1}^2 + 0.92 \sigma_{t-1}^2 \end{aligned}$$

This process is covariance stationary since $\alpha_1 + \beta_1 = 0.07 + 0.92 = 0.9 < 1$, and $\bar{\sigma}^2 = \omega/(1 - \alpha_1 - \beta_1) = 0.01/0.01 = 1$. This GARCH(1,1) model has the same unconditional variance as the ARCH(5) model from the previous example but has much higher persistence. This model can be specified using the **rugarch ugarchspec()** function as follows:

```
garch11.spec = ugarchspec(variance.model = list(garchOrder = c(1, 1)), mean.model = list(armaOrder = c(0, 0)), fixed.pars = list(mu = 0, omega = 0.1, alpha1 = 0.1, beta1 = 0.8))
```

This model has the same unconditional variance and persistence as the ARCH(5) model in the previous example. Simulated values for R_t and σ_t , using the same random number seed as the ARCH(5) simulations, are created using

```
set.seed(123)
garch11.sim = ugarchpath(garch11.spec, n.sim = 1000)
```

and are displayed in Figure xxx. The simulated GARCH(1,1) values of R_t and σ_t are quite different from the simulated ARCH(1) values. In particular, the simulated returns R_t show fewer extreme values than the ARCH(1) returns and the simulated volatilities σ_t values appear to show more persistence than the ARCH(1) volatilities. The sample autocorrelations in Figure xxx show the interesting result that R_t^2 exhibits little autocorrelation whereas σ_t^2 exhibits substantial positive autocorrelation.

```
sd(garch11.sim@path$seriesSim)

## [1] 1.03

kurtosis(garch11.sim@path$seriesSim)

## [1] 0.701
```

■

10.3 Maximum Likelihood Estimation

The estimation of the ARCH-GARCH model parameters is more complicated than the estimation of the CER model parameters. There are no simple plug-in principle estimators for the conditional variance parameters. Instead, an alternative estimation method called *maximum likelihood* (ML) is typically used to estimate the ARCH-GARCH parameters. This section reviews the ML estimation method and shows how it can be applied to estimate the ARCH-GARCH model parameters.

10.3.1 The Likelihood Function

Let X_1, \dots, X_T be an iid sample with probability density function (pdf) $f(x_t; \theta)$, where θ is a $(k \times 1)$ vector of parameters that characterize $f(x_t; \theta)$. For example, if $X_t \sim N(\mu, \sigma^2)$ then $f(x_t; \theta) = (2\pi\sigma^2)^{-1/2} \exp(-\frac{1}{2\sigma^2}(x_t - \mu)^2)$ and $\theta = (\mu, \sigma^2)'$. The *joint density* of the sample is, by independence, equal to the product of the marginal densities

$$f(x_1, \dots, x_T; \theta) = f(x_1; \theta) \cdots f(x_T; \theta) = \prod_{t=1}^T f(x_t; \theta). \quad (10.16)$$

The joint density is an T dimensional function of the data x_1, \dots, x_T given the parameter vector θ . The joint density satisfies⁴

⁴ If X_1, \dots, X_T are discrete random variables, then $f(x_1, \dots, x_T; \theta) = \Pr(X_1 = x_1, \dots, X_T = x_T)$ for a fixed value of θ .

$$f(x_1, \dots, x_T; \theta) \geq 0,$$

$$\int \cdots \int f(x_1, \dots, x_T; \theta) dx_1 \cdots dx_T = 1.$$

The *likelihood function* is defined as the joint density treated as a functions of the parameters θ :

$$L(\theta|x_1, \dots, x_T) = f(x_1, \dots, x_T; \theta) = \prod_{t=1}^T f(x_t; \theta). \quad (10.17)$$

Notice that the likelihood function is a k dimensional function of θ given the data x_1, \dots, x_T . It is important to keep in mind that the likelihood function, being a function of θ and not the data, is not a proper pdf. It is always positive but

$$\int \cdots \int L(\theta|x_1, \dots, x_T) d\theta_1 \cdots d\theta_k \neq 1.$$

To simplify notation, let the vector $\mathbf{x} = (x_1, \dots, x_T)'$ denote the observed sample. Then the joint pdf and likelihood function may be expressed as $f(\mathbf{x}; \theta)$ and $L(\theta|\mathbf{x})$.

Example 10.5. Likelihood function for a sample from the CER model for R_t .

Let R_t denote the daily return on an asset and assume that $\{R_t\}_{t=1}^T$ is described by the CER model. Then $\{R_t\}_{t=1}^T$ is an iid sample with $R_t \sim N(\mu, \sigma^2)$. The pdf for R_t is

$$f(r_t; \theta) = (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(r_t - \mu)^2\right), \quad -\infty < \mu < \infty, \quad \sigma^2 > 0, \quad -\infty < r < \infty,$$

so that $\theta = (\mu, \sigma^2)'$. Then the likelihood function is given by

$$L(\theta|\mathbf{r}) = \prod_{t=1}^T (2\pi\sigma^2)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}(r_t - \mu)^2\right) = (2\pi\sigma^2)^{-T/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{t=1}^T (r_t - \mu)^2\right), \quad (10.18)$$

where $\mathbf{r} = (r_1, \dots, r_T)'$ denotes the observed sample. ■

Now suppose $\{X_t\}$ is a covariance stationary time series whose joint pdf for a sample of size T is given by $f(x_1, \dots, x_T; \theta)$. Here, the factorization of the likelihood function given in (10.16) does not work because the random variables $\{X_t\}_{t=1}^T$ generating the sample $\{x_t\}_{t=1}^T$ are not iid. In this case, to factorize the joint density one can use the conditional-marginal factorization of the joint density $f(x_1, \dots, x_T; \theta)$. To see how this works, consider the joint density of two adjacent observations $f(x_1, x_2; \theta)$. This joint density can be factored as the product of the conditional density of X_2 given $X_1 = x_1$ and the marginal density of X_1 :

$$f(x_1, x_2; \theta) = f(x_2|x_1; \theta)f(x_1; \theta).$$

For three observations, the factorization becomes

$$f(x_1, x_2, x_3; \theta) = f(x_3|x_2, x_1; \theta)f(x_2|x_1; \theta)f(x_1; \theta).$$

For a sample of size T , the conditional-marginal factorization of the joint pdf $f(x_1, \dots, x_T; \theta)$ has the form

$$f(x_1, \dots, x_T; \theta) = \left(\prod_{t=p+1}^T f(x_t | I_{t-1}; \theta) \right) \cdot f(x_1, \dots, x_p; \theta), \quad (10.19)$$

where $I_t = \{x_t, \dots, x_1\}$ denotes the conditioning information at time t , $f(x_t | I_{t-1}; \theta)$ is the pdf of x_t conditional on I_t , and $f(x_1, \dots, x_p; \theta)$ denotes the joint density of the p initial values x_1, \dots, x_p . Hence, the conditional-marginal factorization of the likelihood function is

$$L(\theta | x_1, \dots, x_T) = f(x_1, \dots, x_T; \theta) = \left(\prod_{t=p+1}^T f(x_t | I_{t-1}; \theta) \right) \cdot f(x_1, \dots, x_p; \theta). \quad (10.20)$$

For many covariance stationary time series models, the conditional pdf $f(x_t | I_{t-1}; \theta)$ has a simple form whereas the marginal joint pdf $f(x_1, \dots, x_p; \theta)$ is complicated. For these models, the marginal joint pdf is often ignored in (10.20) which gives the simplified *conditional likelihood function*

$$L(\theta | x_1, \dots, x_T) \approx \left(\prod_{t=p+1}^T f(x_t | I_{t-1}; \theta) \right). \quad (10.21)$$

For reasonably large samples (e.g., $T > 100$) the density of the initial values $f(x_1, \dots, x_p; \theta)$ has a negligible influence on the shape of the likelihood function (10.20) and can be safely ignored. In what follows we will only consider the conditional likelihood function (10.21).

Example 10.6. Likelihood function for a sample from GARCH(1,1) model for R_t .

Let R_t denote the daily return on an asset and assume that $\{R_t\}_{t=1}^T$ is described by the GARCH(1,1) model

$$\begin{aligned} R_t &= \mu + \epsilon_t, \\ \epsilon_t &= \sigma_t z_t, \\ \sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2, \\ z_t &\sim iid N(0, 1). \end{aligned}$$

From the properties of the GARCH(1,1) model we know that $R_t | I_{t-1} \sim N(\mu, \sigma_t^2)$ and so

$$f(r_t | I_{t-1}; \theta) = (2\pi\sigma_t^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_t^2}(r_t - \mu)^2\right), \quad (10.22)$$

where $\theta = (\mu, \omega, \alpha_1, \beta_1)'$. The conditional likelihood (10.21) is then

$$L(\theta | \mathbf{r}) = \prod_{t=1}^n (2\pi\sigma_t^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_t^2}(r_t - \mu)^2\right). \quad (10.23)$$

In (10.23), the values for σ_t^2 are determined recursively from (10.15) starting at $t = 1$

$$\sigma_1^2 = \omega + \alpha_1 \epsilon_0^2 + \beta_1 \sigma_0^2. \quad (10.24)$$

Given values for the parameters ω , α_1 , β_1 and initial values ϵ_0^2 and σ_0^2 the value for σ_1^2 is determined from (10.24). For $t = 2$, we have

$$\sigma_2^2 = \omega + \alpha_1 \epsilon_1^2 + \beta_1 \sigma_1^2 = \omega + \alpha_1 (r_1 - \mu)^2 + \beta_1 \sigma_1^2,$$

where σ_1^2 is determined from (10.24). Hence, the values for σ_t^2 in (10.23) are determined recursively using the GARCH(1,1) variance equation (10.15) starting with (10.24). ■

10.3.2 The Maximum Likelihood Estimator

Consider a sample of observations $\{x_t\}_{t=1}^T$ with joint pdf $f(x_1, \dots, x_T; \theta)$ and likelihood function $L(\theta|x_1, \dots, x_T)$. Suppose that X_t is a discrete random variable so that

$$L(\theta|x_1, \dots, x_T) = f(x_1, \dots, x_T; \theta) = \Pr(X_1 = x_1, \dots, X_T = x_T).$$

So for a given value of θ , say $\theta = \theta_0$, $L(\theta_0|x_1, \dots, x_T)$ gives the probability of the observed sample $\{x_t\}_{t=1}^T$ coming from a model in which θ_0 is the true parameter. Now, some values of θ will lead to higher probabilities of observing $\{x_t\}_{t=1}^T$ than others. The value of θ that gives the highest probability for the observed sample $\{x_t\}_{t=1}^T$ is the value of θ that maximizes the likelihood function $L(\theta|x_1, \dots, x_T)$. This value of θ is called the *maximum likelihood estimator* (MLE) for θ . When X_t is a continuous random variable, $f(x_1, \dots, x_T; \theta)$ is not a joint probability but represents the height of the joint pdf as a function of $\{x_t\}_{t=1}^T$ for a given θ . However, the interpretation of the MLE is similar. It is the value of θ that makes the observed data most likely.

Formally, the MLE for θ , denoted $\hat{\theta}_{mle}$, is the value of θ that maximizes $L(\theta|\mathbf{x})$. That is, $\hat{\theta}_{mle}$ solves the optimization problem

$$\max_{\theta} L(\theta|\mathbf{x}).$$

It is often quite difficult to directly maximize $L(\theta|\mathbf{x})$. It is usually much easier to maximize the log-likelihood function $\ln L(\theta|\mathbf{x})$. Since $\ln(\cdot)$ is a monotonically increasing function the value of the θ that maximizes $\ln L(\theta|\mathbf{x})$ will also maximize $L(\theta|\mathbf{x})$. Therefore, we may also define $\hat{\theta}_{mle}$ as the value of θ that solves

$$\max_{\theta} \ln L(\theta|\mathbf{x}).$$

With random sampling, the log-likelihood is additive in the log of the marginal densities:

$$\ln L(\theta|\mathbf{x}) = \ln \left(\prod_{t=1}^T f(x_t; \theta) \right) = \sum_{t=1}^T \ln f(x_t; \theta).$$

For a covariance stationary time series, the conditional log-likelihood is additive in the log of the conditional densities:

$$\ln L(\theta|\mathbf{x}) = \ln \left(\prod_{t=1}^T f(x_t|I_{t-1};\theta) \right) = \sum_{t=1}^T \ln f(x_t|I_{t-1};\theta).$$

Since the MLE is defined as a maximization problem, under suitable regularity conditions we can determine the MLE using simple calculus.⁵ We find the MLE by differentiating $\ln L(\theta|\mathbf{x})$ and solving the *first order conditions*

$$\frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{x})}{\partial \theta} = \mathbf{0}.$$

Since θ is $(k \times 1)$ the first order conditions define k , potentially nonlinear, equations in k unknown values:

$$\frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{x})}{\partial \theta} = \begin{pmatrix} \frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{x})}{\partial \theta_1} \\ \vdots \\ \frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{x})}{\partial \theta_k} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The vector of derivatives of the log-likelihood function is called the *score* vector and is denoted

$$S(\theta|\mathbf{x}) = \frac{\partial \ln L(\theta|\mathbf{x})}{\partial \theta}.$$

By definition, the MLE satisfies $S(\hat{\theta}_{mle}|\mathbf{x}) = \mathbf{0}$.

In some cases, it is possible to find analytic solutions to the set of equations $S(\hat{\theta}_{mle}|\mathbf{x}) = \mathbf{0}$. However, for ARCH and GARCH models the set of equations $S(\hat{\theta}_{mle}|\mathbf{x}) = \mathbf{0}$ are complicated nonlinear functions of the elements of θ and no analytic solutions exist. As a result, numerical optimization methods are required to determine $\hat{\theta}_{mle}$.

Example 10.7. MLE for parameters in CER model

In the CER model with likelihood function (10.18), the log-likelihood function is

$$\ln L(\theta|\mathbf{r}) = -\frac{T}{2} \ln(2\pi) - \frac{T}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{t=1}^T (r_t - \mu)^2, \quad (10.25)$$

and we may determine the MLE for $\theta = (\mu, \sigma^2)'$ by maximizing the log-likelihood (10.25). The sample score is a (2×1) vector given by

$$S(\theta|\mathbf{r}) = \begin{pmatrix} \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \mu} \\ \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \sigma^2} \end{pmatrix},$$

where

⁵ Standard regularity conditions are: (1) the support of the random variables X , $S_X = \{x : f(x;\theta) > 0\}$, does not depend on θ ; (2) $f(x;\theta)$ is at least three times differentiable with respect to θ ; (3) the true value of θ lies in a compact set Θ .

$$\begin{aligned}\frac{\partial \ln L(\theta|\mathbf{r})}{\partial \mu} &= \frac{1}{\sigma^2} \sum_{t=1}^T (r_t - \mu), \\ \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \sigma^2} &= -\frac{T}{2} (\sigma^2)^{-1} + \frac{1}{2} (\sigma^2)^{-2} \sum_{t=1}^T (r_t - \mu)^2.\end{aligned}$$

Solving $S(\hat{\theta}_{mle}|\mathbf{r}) = 0$ gives the *normal equations*

$$\begin{aligned}\frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{r})}{\partial \mu} &= \frac{1}{\hat{\sigma}_{mle}^2} \sum_{t=1}^T (r_t - \hat{\mu}_{mle}) = 0 \\ \frac{\partial \ln L(\hat{\theta}_{mle}|\mathbf{r})}{\partial \sigma^2} &= -\frac{T}{2} (\hat{\sigma}_{mle}^2)^{-1} + \frac{1}{2} (\hat{\sigma}_{mle}^2)^{-2} \sum_{t=1}^T (r_t - \hat{\mu}_{mle})^2 = 0\end{aligned}$$

Solving the first equation for $\hat{\mu}_{mle}$ gives

$$\hat{\mu}_{mle} = \frac{1}{T} \sum_{i=1}^T r_t = \bar{r}. \quad (10.26)$$

Hence, the sample average is the MLE for μ . Using $\hat{\mu}_{mle} = \bar{r}$ and solving the second equation for $\hat{\sigma}_{mle}^2$ gives

$$\hat{\sigma}_{mle}^2 = \frac{1}{T} \sum_{i=1}^T (r_t - \bar{r})^2. \quad (10.27)$$

Notice that $\hat{\sigma}_{mle}^2$ is not equal to the sample variance which uses a divisor of $T-1$ instead of T .

It is instructive to compare the MLEs for μ and σ^2 with the CER model estimates presented in Chapter xxx. Recall, the CER model estimators for μ and σ^2 are motivated by the plug-in principle and are equal to the sample mean and variance, respectively. Here, we see that the MLE for μ is equal to the sample mean and the MLE for σ^2 is $(T-1)/T$ times the sample variance. In fact, it can be shown that the CER model estimates for the other parameters are also equal to or approximately equal to the MLEs.

■

Example 10.8. MLE for parameters in GARCH(1,1) model

In the GARCH(1,1) model with likelihood function (10.23) and parameter vector $\theta = (\mu, \omega, \alpha_1, \beta_1)'$, the log-likelihood function is

$$\ln L(\theta|\mathbf{r}) = -\frac{T}{2} \ln(2\pi) - \sum_{t=1}^T \ln(\sigma_t^2(\theta)) - \frac{1}{2} \sum_{t=1}^T \frac{(r_t - \mu)^2}{\sigma_t^2(\theta)}. \quad (10.28)$$

In (), it is instructive to write $\sigma_t^2 = \sigma_t^2(\theta)$ to emphasize that σ_t^2 is a function of θ . That is, from ()

$$\sigma_t^2 = \sigma_t^2(\theta) = \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2(\theta) = \omega + \alpha_1 (r_{t-1} - \mu)^2 + \beta_1 \sigma_{t-1}^2(\theta).$$

The sample score is a (4×1) vector given by

$$S(\theta|\mathbf{r}) = \begin{pmatrix} \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \mu} \\ \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \omega} \\ \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \alpha_1} \\ \frac{\partial \ln L(\theta|\mathbf{r})}{\partial \beta_1} \end{pmatrix}.$$

The elements of $S(\theta|\mathbf{r})$, unfortunately, do not have simple closed form expressions and no analytic formulas are available for the MLEs of the elements of θ . As a result, numerical methods, as discussed in sub-section xxx below, are required to determine $\hat{\theta}_{mle}$ that maximizes (10.28) and solves $S(\hat{\theta}_{mle}|\mathbf{x}) = 0$.

■

10.3.3 Invariance Property of Maximum Likelihood Estimators

One of the attractive features of the method of maximum likelihood is its invariance to one-to-one transformations of the parameters of the log-likelihood. That is, if $\hat{\theta}_{mle}$ is the MLE of θ and $\alpha = h(\theta)$ is a one-to-one function of θ then $\hat{\alpha}_{mle} = h(\hat{\theta}_{mle})$ is the MLE for α .

Example 10.9. MLE of volatility in the CER model

In the CER model, the log-likelihood is parametrized in terms of μ and σ^2 and we have the MLEs (10.26) and (10.27). Suppose we are interested in the MLE for $\sigma = h(\sigma^2) = (\sigma^2)^{1/2}$, which is a one-to-one function for $\sigma^2 > 0$. The invariance property of MLEs gives the result

$$\hat{\sigma}_{mle} = (\hat{\sigma}_{mle}^2)^{1/2} = \left(\frac{1}{T} \sum_{i=1}^T (r_t - \hat{\mu}_{mle})^2 \right)^{1/2}.$$

10.3.4 The Precision of the Maximum Likelihood Estimator

The likelihood, log-likelihood and score functions for a typical model are illustrated in figure xxx. The likelihood function is always positive (since it is the joint density of the sample) but the log-likelihood function is typically negative (being the log of a number less than 1). Here the log-likelihood is globally concave and has a unique maximum at $\hat{\theta}_{mle}$. Consequently, the score function is positive to the left of the maximum, crosses zero at the maximum and becomes negative to the right of the maximum.

Intuitively, the precision of $\hat{\theta}_{mle}$ depends on the curvature of the log-likelihood function near $\hat{\theta}_{mle}$. If the log-likelihood is very curved or “steep” around $\hat{\theta}_{mle}$, then θ will be precisely estimated. In this case, we say that we have a lot of *information* about θ . On the other hand, if the log-likelihood is not curved or “flat” near $\hat{\theta}_{mle}$, then θ will not be precisely estimated. Accordingly, we say that we do not have much information about θ .

The extreme case of a completely flat likelihood in θ is illustrated in figure xxx. Here, the sample contains no information about the true value of θ because every value of θ produces

the same value of the likelihood function. When this happens we say that θ is not *identified*. Formally, θ is identified if for all $\theta_1 \neq \theta_2$ there exists a sample \mathbf{x} for which $L(\theta_1|\mathbf{x}) \neq L(\theta_2|\mathbf{x})$. The curvature of the log-likelihood is measured by its second derivative (*Hessian*)

$$H(\theta|\mathbf{x}) = \frac{\partial^2 \ln L(\theta|\mathbf{x})}{\partial \theta \partial \theta'}$$

Since the Hessian is negative semi-definite, the *information* in the sample about θ may be measured by $-H(\theta|\mathbf{x})$. If θ is a scalar then $-H(\theta|\mathbf{x})$ is a positive number. The expected amount of information in the sample about the parameter θ is the *information matrix*

$$I(\theta|\mathbf{x}) = -E[H(\theta|\mathbf{x})]$$

As we shall see in the next sub-section, the information matrix is directly related to the precision of the MLE.

10.3.5 Asymptotic Properties of Maximum Likelihood Estimators

- Write this section in a similar way as the asymptotic section in the chapter on Estimating the CER model. Many of the concepts are exactly the same.
- Draw connection between asymptotic results here and those in previous chapter.

Let $\{X_t\}$ be a covariance stationary time series with conditional probability density function (pdf) $f(x_t; I_{t-1}; \theta)$, where θ is a $(k \times 1)$ vector of parameters that characterize $f(x_t; I_{t-1}; \theta)$. Under suitable regularity conditions, the ML estimator of θ has the following asymptotic properties as the sample size

1. $\hat{\theta}_{mle} \xrightarrow{P} \theta$
2. $\sqrt{n}(\hat{\theta}_{mle} - \theta) \xrightarrow{d} N(0, I(\theta|x_t)^{-1})$, where

$$I(\theta|x_t) = -E[H(\theta|x_t)] = -E\left[\frac{\partial^2 \ln f(\theta|x_t)}{\partial \theta \partial \theta'}\right]$$

That is,

$$\text{avar}(\sqrt{n}(\hat{\theta}_{mle} - \theta)) = I(\theta|x_t)^{-1}$$

Alternatively,

$$\hat{\theta}_{mle} \sim N\left(\theta, \frac{1}{n} I(\theta|x_t)^{-1}\right) = N(\theta, I(\theta|\mathbf{x})^{-1})$$

where $I(\theta|\mathbf{x}) = nI(\theta|x_t)$ = information matrix for the sample.

3. $\hat{\theta}_{mle}$ is efficient in the class of consistent and asymptotically normal estimators. That is,

$$\text{avar}(\sqrt{n}(\hat{\theta}_{mle} - \theta)) - \text{avar}(\sqrt{n}(\tilde{\theta} - \theta)) \leq 0$$

for any consistent and asymptotically normal estimator $\tilde{\theta}$.

10.3.6 Computing the MLE Using Numerical Optimization Methods

Computation: Newton-Raphson Iteration

Goal: Use an iterative scheme to compute

$$\hat{\theta} = \arg \max_{\theta} \ln L(\theta | \mathbf{x})$$

Numerical optimization methods generally have the following structure:

1. Determine a set of starting values
2. Update the starting values in a direction that increases $\ln L(\theta | \mathbf{x})$
3. Check to see if the update satisfies the FOCs. If not, update again.
4. Stop updating when the FOCs are satisfied.

The most common method for numerically maximizing () is *Newton-Raphson iteration*. This method can be motivated as follows. Consider a second order Taylor's series expansion of $\ln L(\theta | \mathbf{x})$ about starting value $\hat{\theta}_1$

$$\begin{aligned} \ln L(\theta | \mathbf{x}) &= \ln L(\hat{\theta}_1 | \mathbf{x}) + \frac{\partial \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta'} (\theta - \hat{\theta}_1) \\ &\quad + \frac{1}{2} (\theta - \hat{\theta}_1)' \frac{\partial^2 \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta \partial \theta'} (\theta - \hat{\theta}_1) + \text{error}. \end{aligned} \quad (10.29)$$

Now maximize (10.29) with respect to θ . The FOCs are

$$\mathbf{0}_{p \times 1} = \frac{\partial \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta} + \frac{\partial^2 \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta \partial \theta'} (\hat{\theta}_2 - \hat{\theta}_1),$$

which can be solved for $\hat{\theta}_2$:

$$\begin{aligned} \hat{\theta}_2 &= \hat{\theta}_1 - \left[\frac{\partial^2 \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta \partial \theta'} \right]^{-1} \frac{\partial \ln L(\hat{\theta}_1 | \mathbf{x})}{\partial \theta} \\ &= \hat{\theta}_1 - H(\hat{\theta}_1 | \mathbf{x})^{-1} S(\hat{\theta}_1 | \mathbf{x}) \end{aligned}$$

This suggests the iterative scheme

$$\hat{\theta}_{n+1} = \hat{\theta}_n - H(\hat{\theta}_n | \mathbf{x})^{-1} S(\hat{\theta}_n | \mathbf{x})$$

Iteration stops when $S(\hat{\theta}_n | \mathbf{x}) \approx 0$.

- Practical considerations: Stopping criterion
- Practical considerations: analytic or numerical derivatives
- Practical considerations: NR gives $H(\hat{\theta}_n | \mathbf{x})^{-1}$ as a by-product of the optimization and this is used to compute the asymptotic standard errors of the MLE

10.4 Estimation of ARCH-GARCH Models in R Using rugarch

To be completed.

10.5 Forecasting Conditional Volatility from ARCH Models

An important task of modeling conditional volatility is to generate accurate forecasts for both the future value of a financial time series as well as its conditional volatility. Volatility forecasts are used for risk management, option pricing, portfolio allocation, trading strategies and model evaluation. ARCH and GARCH models can generate accurate forecasts of future daily return volatility, especially over short horizons, and these forecasts will eventually converge to the unconditional volatility of daily returns. This section illustrates how to forecast volatility using the GARCH(1,1) model.

10.5.1 Forecasting daily return volatility from the GARCH(1,1) model

For simplicity, consider the basic GARCH(1,1) model

$$\begin{aligned} R_t &= \mu + \epsilon_t \\ \epsilon_t &= \sigma_t z_t \\ \sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \\ z_t &\sim iid N(0, 1) \end{aligned}$$

Assume the sample period is $t = 1, 2, \dots, T$ and forecasts for volatility are desired for the out-of-sample periods $T+1, T+2, \dots, T+k$. For ease of notation, assume the GARCH(1,1) parameters are known.⁶ The optimal, in terms of mean-squared error, forecast of σ_{T+k}^2 given information at time T is $E[\sigma_{T+k}^2 | I_T]$ and can be computed using a simple recursion known as the chain-rule of forecasting. For $k = 1$,

$$\begin{aligned} E[\sigma_{T+1}^2 | I_T] &= \omega + \alpha_1 E[\epsilon_T^2 | I_T] + \beta_1 E[\sigma_T^2 | I_T] \\ &= \omega + \alpha_1 \varepsilon_T^2 + \beta_1 \sigma_T^2, \end{aligned} \tag{10.30}$$

where it is assumed that ε_T^2 and σ_T^2 are known at time T . Similarly, for $k = 2$

$$E[\sigma_{T+2}^2 | I_T] = \omega + \alpha_1 E[\varepsilon_{T+1}^2 | I_T] + \beta_1 E[\sigma_{T+1}^2 | I_T]. \tag{10.31}$$

Now, in the GARCH(1,1) model, $E[\varepsilon_{T+1}^2 | I_T] = E[z_{T+1}^2 \sigma_{T+1}^2 | I_T] = E[\sigma_{T+1}^2 | I_T]$ so that (10.31) becomes

⁶ In practice, the GARCH forecasting algorithm will use the GARCH parameters estimated over the sample period.

$$\begin{aligned} E[\sigma_{T+2}^2 | I_T] &= \omega + (\alpha_1 + \beta_1) E[\sigma_{T+1}^2 | I_T] \\ &= \omega + (\alpha_1 + \beta_1) (\alpha_1 \varepsilon_T^2 + \beta_1 \sigma_T^2) \end{aligned}$$

In general, for $k \geq 2$ we have

$$E[\sigma_{T+k}^2 | I_T] = \omega + (\alpha_1 + \beta_1) E[\sigma_{T+k-1}^2 | I_T] \quad (10.32)$$

which, by recursive substitution, reduces to

$$E[\sigma_{T+k}^2 | I_T] = \omega \sum_{i=0}^{k-1} (\alpha_1 + \beta_1)^i + (\alpha_1 + \beta_1)^{k-1} (\alpha_1 \varepsilon_T^2 + \beta_1 \sigma_T^2).$$

Now, since $0 < \alpha_1 + \beta_1 < 1$, as $k \rightarrow \infty$

$$\begin{aligned} \sum_{i=0}^{k-1} (\alpha_1 + \beta_1)^i &\rightarrow \frac{1}{1 - (\alpha_1 + \beta_1)}, \\ (\alpha_1 + \beta_1)^{k-1} &\rightarrow 0, \end{aligned}$$

and so

$$E[\sigma_{T+k}^2 | I_T] \rightarrow \frac{\omega}{1 - (\alpha_1 + \beta_1)} = \bar{\sigma}^2 = \text{var}(R_t).$$

Notice that the speed at which $E[\sigma_{T+k}^2 | I_T]$ approaches $\bar{\sigma}^2$ is captured by approaches $\bar{\sigma}^2$ is captured by the GARCH(1,1) persistence $\alpha_1 + \beta_1$.

An alternative representation of the forecasting equation (10.32) starts with the mean-adjusted form

$$\sigma_{T+1}^2 - \bar{\sigma}^2 = \alpha_1 (\varepsilon_T^2 - \bar{\sigma}^2) + \beta_1 (\sigma_T^2 - \bar{\sigma}^2),$$

where $\bar{\sigma}^2 = \omega / (1 - \alpha_1 - \beta_1)$ is the unconditional variance. Then by recursive substitution

$$E[\sigma_{T+k}^2 | I_T] - \bar{\sigma}^2 = (\alpha_1 + \beta_1)^{k-1} (E[\sigma_{T+1}^2 | I_T] - \bar{\sigma}^2).$$

The forecasting algorithm (10.32) produces unbiased forecasts for the conditional variance σ_{T+k}^2 . The forecast for the conditional volatility, σ_{T+k} , is computed as the square root of the forecast for σ_{T+k}^2 which is not unbiased because

$$E[\sigma_{T+k} | I_T] \neq \sqrt{E[\sigma_{T+k}^2 | I_T]}.$$

Example 10.10. Forecasting daily volatility from fitted GARCH(1,1) models using **rugarch**

To be completed

■

10.5.2 Forecasting multi-day return volatility using a GARCH(1,1) model

To be completed

10.6 Forecasting VaR from ARCH Models

To be completed

10.7 Further Reading

To be completed

10.8 Problems

References

1. Ruppert, D.A.. *Statistics and Data Analysis for Financial Engineering*, Springer-Verlag.
2. Martin, R.D., Scherer, B., and Yollin, G. (2016).

Chapter 11

Introduction to Portfolio Theory

Updated: August 5, 2016

Copyright © Eric Zivot 2015, 2016

Chapters 7 through 9 focused on the development and estimation of, and the statistical inference for, the CER model for asset returns. In this chapter, we use the CER model for asset returns as the basis for the quantitative analysis of portfolios and asset allocation. Specifically, we introduce the mean-variance portfolio analysis framework originally developed by Harry Markowitz.¹ This framework is the widely used in academia and industry and is covered in most standard textbooks on investment analysis such as Bodie et al. (2013), Elton et al. (2014), and Sharpe (1999). In this framework, asset returns are assumed to be normally distributed and so investor preferences over portfolios of assets depend only on the expected return and variance of the portfolio returns. Investors are assumed to like portfolio with high mean returns but dislike portfolio with high return variances. Since portfolios that have high (low) expected returns tend to have high (low) variances investors face a mean-variance tradeoff between different portfolios. Markowitz's mean-variance portfolio theory quantifies this mean-variance tradeoff and provides a methodology that investors can use to determine optimal portfolio choice.

This chapter introduces Markowitz's mean-variance portfolio theory in a simplified setting where there are only two risky assets and a single risk-free asset. This allows use to present the main results of mean-variance portfolio theory graphically and with simple algebra. However, as we shall see in the next chapter, many of the analytic results we derive for portfolios of two risky assets, and for portfolios of two risky assets and a single risk-free asset, can be applied to situations in which there are many risky assets and a single risk-free asset. Therefore, it is highly beneficial to study and understand well the portfolio theory results in the simplified framework of this chapter.

This chapter is organized as follows. Section 11.1 reviews expected return, variance and value-at-risk calculations for portfolios of two risky assets. Section 11.2 introduces mean-variance portfolio analysis for portfolios of two risky assets. The portfolio risk-return frontier

¹ The mean-variance frame is described in detail in Markowitz (1987, 1991) . Harry Markowitz was awarded the Nobel Prize in Economics in 1990, along with Merton Miller and William Sharpe, for his work on portfolio theory.

is introduced, efficient portfolios are defined, and the global minimum variance portfolio is defined. Section 11.3 presents the analysis of portfolios with a single risky asset and a risk-free asset. The Sharpe ratio/slope of a risky asset is defined and illustrated graphically. Portfolios of two risky assets and a risk-free asset are discussed in Section 11.4. The tangency portfolio is defined and the Mutual Fund Separation Theorem is presented. Section 11.5 provides a brief illustration of the theory to a real world two asset portfolio of stocks and bonds. The Appendix reviews some basic results on constrained optimization of functions.

The R packages used in this chapter are **IntroCompFinR** and **PerformanceAnalytics**. Make sure these packages are installed and loaded in R before replicating the chapter examples.

11.1 Portfolios of Two Risky Assets

Consider the following investment problem. We can invest in two stocks Amazon (A) and Boeing (B) over the next month. Let R_A denote monthly simple return on Amazon and R_B denote the monthly simple return on Boeing.² These returns are to be treated as random variables because the returns will not be realized until the end of the month. We assume that the returns R_A and R_B are jointly normally distributed, and that we have the following information about the means, variances and covariances of the probability distribution of the two returns:

$$\mu_A = E[R_A], \sigma_A^2 = \text{var}(R_A), \mu_B = E[R_B], \sigma_B^2 = \text{var}(R_B), \quad (11.1)$$

$$\sigma_{AB} = \text{cov}(R_A, R_B), \rho_{AB} = \text{cor}(R_A, R_B) = \frac{\sigma_{AB}}{\sigma_A \sigma_B}. \quad (11.2)$$

We assume that these values are taken as given. Typically, they are estimated from historical return data for the two stocks. However, they can also be subjective guesses by an analyst.

The expected returns, μ_A and μ_B , are our best guesses for the monthly returns on each of the stocks. However, because the investment returns are random variables we must recognize that the realized returns may be different from our expectations. The variances, σ_A^2 and σ_B^2 , provide measures of the uncertainty associated with these monthly returns. We can also think of the variances as measuring the risk associated with the investments. Assets with high return variability (or volatility) are often thought to be risky, and assets with low return volatility are often thought to be safe. The covariance σ_{AB} gives us information about the direction of any linear dependence between returns. If $\sigma_{AB} > 0$ then the two returns tend to move in the same direction; if $\sigma_{AB} < 0$ the returns tend to move in opposite directions; if $\sigma_{AB} = 0$ then the returns tend to move independently. The strength of the dependence between the returns is measured by the correlation coefficient ρ_{AB} . If ρ_{AB} is close to one in absolute value then returns mimic each other extremely closely, whereas if ρ_{AB} is close to zero then the returns may show very little relationship.

² If the stocks pay a dividend then returns are computed as total returns.

Example 11.1. Two risky asset portfolio information.

Table 11.1 gives annual return distribution parameters for two hypothetical assets A and B . Asset A is the high risk asset with an annual return of $\mu_A = 17.5\%$ and annual standard deviation of $\sigma_A = 25.8\%$. Asset B is a lower risk asset with annual return $\mu_B = 5.5\%$ and annual standard deviation of $\sigma_B = 11.5\%$. The assets are assumed to be slightly negatively correlated with correlation coefficient $\rho_{AB} = -0.164$.³ Given the standard deviations and the correlation, the covariance can be determined from $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B = (-0.164)(0.0258)(0.115) = -0.004875$. In R, the example data is:

```
mu.A = 0.175
sig.A = 0.258
sig2.A = sig.A^2
mu.B = 0.055
sig.B = 0.115
sig2.B = sig.B^2
rho.AB = -0.164
sig.AB = rho.AB * sig.A * sig.B
```

μ_A	μ_B	σ_A^2	σ_B^2	σ_A	σ_B	σ_{AB}	ρ_{AB}
0.175	0.055	0.06656	0.01323	0.258	0.115	-0.004866	-0.164

Table 11.1 Example data for two asset portfolio.

The portfolio problem is set-up as follows. We have a given amount of initial wealth W_0 , and it is assumed that we will exhaust all of our wealth between investments in the two stocks. The investment problem is to decide how much wealth to put in asset A and how much to put in asset B . Let x_A denote the share of wealth invested in stock A , and x_B denote the share of wealth invested in stock B . The values of x_A and x_B can be positive or negative. Positive values denote *long* positions (purchases) in the assets. Negative values denote *short* positions (sales)⁴. Since all wealth is put into the two investments it follows that $x_A + x_B = 1$. If asset A is shorted, then it is assumed that the proceeds of the short sale are used to purchase more of asset B . Therefore, to solve the investment problem we must choose the values of x_A and x_B .

Our investment in the two stocks forms a *portfolio*, and the shares x_A and x_B are referred to as *portfolio shares* or weights. The return on the portfolio over the next month is a random variable, and is given by:

³ The small negative correlation between the two stock returns is a bit unusual, as most stock returns are positively correlated. The negative correlation is assumed so that certain figures produced later look nice and are easy to read.

⁴ To short an asset one borrows the asset, usually from a broker, and then sells it. The proceeds from the short sale are usually kept on account with a broker and there often restrictions that prevent the use of these funds for the purchase of other assets. The short position is closed out when the asset is repurchased and then returned to original owner. If the asset drops in value then a gain is made on the short sale and if the asset increases in value a loss is made.

$$R_p = x_A R_A + x_B R_B, \quad (11.3)$$

which is a linear combination or weighted average of the random variables R_A and R_B . Since R_A and R_B are assumed to be normally distributed, R_p is also normally distributed. We use the properties of linear combinations of random variables to determine the mean and variance of this distribution.

11.1.1 Portfolio expected return and variance

The distribution of the return on the portfolio (11.3) is a normal with mean, variance and standard deviation given by:

$$\mu_p = E[R_p] = x_A \mu_A + x_B \mu_B, \quad (11.4)$$

$$\sigma_p^2 = \text{var}(R_p) = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}, \quad (11.5)$$

$$\sigma_p = \text{SD}(R_p) = \sqrt{x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}}. \quad (11.6)$$

That is,

$$R_p \sim N(\mu_p, \sigma_p^2).$$

The results (11.4) and (11.5) are so important to portfolio theory that it is worthwhile to review the derivations.⁵ For the first result (11.4), we have:

$$E[R_p] = E[x_A R_A + x_B R_B] = x_A E[R_A] + x_B E[R_B] = x_A \mu_A + x_B \mu_B,$$

by the linearity of the expectation operator. For the second result (11.5), we have:

$$\begin{aligned} \text{var}(R_p) &= E[(R_p - \mu_p)^2] = E[(x_A(R_A - \mu_A) + x_B(R_B - \mu_B))^2] \\ &= E[x_A^2(R_A - \mu_A)^2 + x_B^2(R_B - \mu_B)^2 + 2x_A x_B(R_A - \mu_A)(R_B - \mu_B)] \\ &= x_A^2 E[(R_A - \mu_A)^2] + x_B^2 E[(R_B - \mu_B)^2] + 2x_A x_B E[(R_A - \mu_A)(R_B - \mu_B)] \\ &= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}. \end{aligned}$$

Notice that the variance of the portfolio is a weighted average of the variances of the individual assets plus two times the product of the portfolio weights times the covariance between the assets. If the portfolio weights are both positive then a positive covariance will tend to increase the portfolio variance, because both returns tend to move in the same direction, and a negative covariance will tend to reduce the portfolio variance. Thus finding assets with negatively correlated returns can be very beneficial when forming portfolios because risk, as measured by portfolio standard deviation, can be reduced. What is perhaps surprising is that forming portfolios with positively correlated assets can also reduce risk as long as the correlation is not too large.

Example 11.2. Two asset portfolios.

⁵ See chapter 2 section 2.1.

Consider creating some portfolios using the asset information in Table 11.1. The first portfolio is an equally weighted portfolio with $x_A = x_B = 0.5$. Using (11.4)-(11.6), we have:

$$\begin{aligned}\mu_p &= (0.5) \cdot (0.175) + (0.5) \cdot (0.055) = 0.115 \\ \sigma_p^2 &= (0.5)^2 \cdot (0.067) + (0.5)^2 \cdot (0.013) \\ &\quad + 2 \cdot (0.5)(0.5)(-0.004866) \\ &= 0.01751 \\ \sigma_p &= \sqrt{0.01751} = 0.1323\end{aligned}$$

This portfolio has expected return half-way between the expected returns on assets A and B , but the portfolio standard deviation is less than half-way between the asset standard deviations. This reflects risk reduction via diversification. In R, the portfolio parameters are computed using:

```
x.A = 0.5
x.B = 0.5
mu.p1 = x.A * mu.A + x.B * mu.B
sig2.p1 = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p1 = sqrt(sig2.p1)
mu.p1

## [1] 0.115

sig2.p1

## [1] 0.01751

sig.p1

## [1] 0.1323
```

Next, consider a long-short portfolio with $x_A = 1.5$ and $x_B = -0.5$. In this portfolio, asset B is sold short and the proceeds of the short sale are used to leverage the investment in asset A . The portfolio characteristics are:

$$\begin{aligned}\mu_p &= (1.5) \cdot (0.175) + (-0.5) \cdot (0.055) = 0.235 \\ \sigma_p^2 &= (1.5)^2 \cdot (0.067) + (-0.5)^2 \cdot (0.013) \\ &\quad + 2 \cdot (1.5)(-0.5)(-0.004866) \\ &= 0.1604 \\ \sigma_p &= \sqrt{0.1604} = 0.4005\end{aligned}$$

This portfolio has both a higher expected return and standard deviation than asset A . In R, the portfolio parameters are computed using:

```
x.A = 1.5
x.B = -0.5
mu.p2 = x.A * mu.A + x.B * mu.B
sig2.p2 = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p2 = sqrt(sig2.p2)
mu.p2
```

```

## [1] 0.235

sig2.p2

## [1] 0.1604

sig.p2

## [1] 0.4005

```

■

In the above example, the equally weighted portfolio has expected return half way between the expected returns on assets A and B, but has standard deviation (volatility) that is less than half way between the standard deviations of the two assets. For long-only portfolios, we can show that this is a general result as long as the correlation between the two assets is not perfectly positive. Consider a portfolio of the two assets A and B with portfolios weights $x_A \geq 0$ and $x_B \geq 0$ such that $x_A + x_B = 1$. Then if $\rho_{AB} \neq 1$

$$\sigma_p = (x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB})^{1/2} < x_A \sigma_A + x_B \sigma_B,$$

and if $\rho_{AB} = 1$ then $\sigma_p = x_A \sigma_A + x_B \sigma_B$. To see this, use $\sigma_{AB} = \rho_{AB} \sigma_A \sigma_B$ and write

$$\sigma_p^2 = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB} = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \rho_{AB} \sigma_A \sigma_B.$$

Now add and subtract $2x_A x_B \sigma_A \sigma_B$ from the right-hand side to give

$$\begin{aligned}\sigma_p^2 &= (x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_A \sigma_B) - 2x_A x_B \sigma_A \sigma_B + 2x_A x_B \rho_{AB} \sigma_A \sigma_B \\ &= (x_A \sigma_A + x_B \sigma_B)^2 - 2x_A x_B \sigma_A \sigma_B (1 - \rho_{AB}).\end{aligned}$$

If $\rho_{AB} \neq 1$ then $2x_A x_B \sigma_A \sigma_B (1 - \rho_{AB}) > 0$ and so

$$\sigma_p^2 < (x_A \sigma_A + x_B \sigma_B)^2 \Rightarrow \sigma_p < x_A \sigma_A + x_B \sigma_B.$$

If $\rho_{AB} = 1$ then $2x_A x_B \sigma_A \sigma_B (1 - \rho_{AB}) = 0$ and

$$\sigma_p^2 = (x_A \sigma_A + x_B \sigma_B) \Rightarrow \sigma_p = x_A \sigma_A + x_B \sigma_B$$

This result shows that the volatility of a two asset portfolio volatility is a weighted average of individual asset volatility only when the two assets have perfectly positively correlated returns. Hence, in this case there is no risk reduction benefit from forming a portfolio. However, if assets are not perfectly correlated then there can be a risk reduction benefit from forming a portfolio. How big the risk reduction benefit depends on the magnitude of the asset return correlation ρ_{AB} . In general, the closer ρ_{AB} is to -1 the larger is the risk reduction benefit.

Figure 11.1 shows the risk-return characteristics of assets A and B as well as the equally weighted and long-short portfolios (labeled P1 and P2, respectively). The dotted line connecting the points A and B represents linear risk-return trade-offs of portfolios of assets A and B, respectively. The point P1, which represents the equally weighted portfolio, has

expected return half-way between the expected returns on assets A and B but has standard deviation less than half-way between the standard deviations of assets A and B. As a result, P1 lies to the left of the dotted line indicating a risk reduction benefit to the portfolio. In contrast, the point P2, which represents the long-short portfolio, lies to the right of the dotted line indicating a risk inflation cost of the portfolio due to leverage.

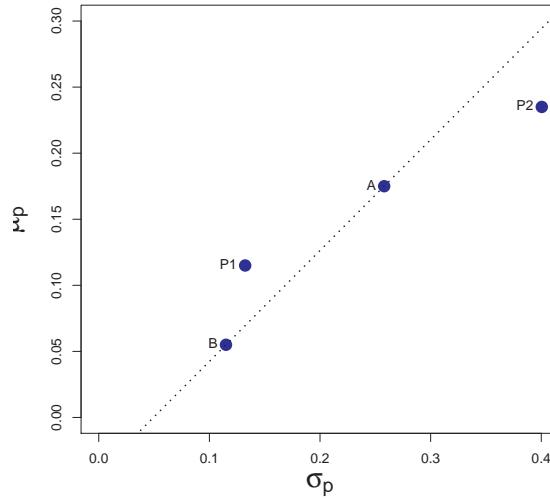


Fig. 11.1 Risk-return characteristics of assets A and B, the equally weighted portfolio P1 and the long-short portfolio P2.

11.1.2 Portfolio Value-at-Risk

Consider an initial investment of $\$W_0$ in the portfolio of assets A and B with return given by (11.3), expected return given by (11.4) and variance given by (11.5). Then $R_p \sim N(\mu_p, \sigma_p^2)$. For $\alpha \in (0, 1)$, the $\alpha \times 100\%$ portfolio value-at-risk is given by:

$$\text{VaR}_{p,\alpha} = q_{p,\alpha}^R W_0, \quad (11.7)$$

where $q_{p,\alpha}^R$ is the α quantile of the distribution of R_p and is given by,

$$q_{p,\alpha}^R = \mu_p + \sigma_p q_\alpha^z, \quad (11.8)$$

where q_α^z is the α quantile of the standard normal distribution⁶.

⁶ If R_p is a continuously compounded return then the implied simple return quantile is $q_{p,\alpha}^R = \exp(\mu_p + \sigma_p q_\alpha^z) - 1$.

What is the relationship between portfolio VaR and the individual asset VaRs? Is portfolio VaR a weighted average of the individual asset VaRs? In general, portfolio VaR is not a weighted average of the asset VaRs. To see this consider the portfolio weighted average of the individual asset return quantiles:

$$\begin{aligned} x_A q_{A,\alpha}^R + x_B q_{B,\alpha}^R &= x_A(\mu_A + \sigma_A q_\alpha^z) + x_B(\mu_B + \sigma_B q_\alpha^z) \\ &= x_A\mu_A + x_B\mu_B + (x_A\sigma_A + x_B\sigma_B)q_\alpha^z \\ &= \mu_p + (x_A\sigma_A + x_B\sigma_B)q_\alpha^z. \end{aligned} \quad (11.9)$$

The weighted asset quantile (11.9) is not equal to the portfolio quantile (11.8) unless $\rho_{AB} = 1$. Hence, weighted asset VaR is in general not equal to portfolio VaR because the quantile (11.9) assumes a perfectly positive correlation between R_A and R_B .

Example 11.3. Portfolio VaR.

Consider an initial investment of $W_0 = \$100,000$. Assuming that returns are simple, the 5% VaRs on assets A and B are:

$$\begin{aligned} \text{VaR}_{A,0.05} &= q_{0.05}^{R_A} W_0 = (\mu_A + \sigma_A q_{0.05}^z) W_0 \\ &= (0.175 + 0.258(-1.645)) \cdot 100,000 = -24,937, \\ \text{VaR}_{B,0.05} &= q_{0.05}^{R_B} W_0 = (\mu_B + \sigma_B q_{0.05}^z) W_0 \\ &= (0.055 + 0.115(-1.645)) \cdot 100,000 = -13,416. \end{aligned}$$

The 5% VaR on the equal weighted portfolio with $x_A = x_B = 0.5$ is:

$$\begin{aligned} \text{VaR}_{p,0.05} &= q_{0.05}^{R_p} W_0 = (\mu_p + \sigma_p q_{0.05}^z) W_0 \\ &= (0.115 + 0.1323(-1.645)) \cdot 100,000 = -10,268, \end{aligned}$$

and the weighted average of the individual asset VaRs is,

$$x_A \text{VaR}_{A,0.05} + x_B \text{VaR}_{B,0.05} = 0.5(-24,937) + 0.5(-13,416) = -19,177.$$

The 5% VaR on the long-short portfolio with $x_A = 1.5$ and $x_B = -0.5$ is:

$$\text{VaR}_{p,0.05} = q_{0.05}^{R_p} W_0 = (0.235 + 0.4005(-1.645)) \cdot 100,000 = -42,371,$$

and the weighted average of the individual asset VaRs is,

$$x_A \text{VaR}_{A,0.05} + x_B \text{VaR}_{B,0.05} = 1.5(-24,937) - 0.5(-13,416) = -30,698.$$

Notice that $\text{VaR}_{p,0.05} \neq x_A \text{VaR}_{A,0.05} + x_B \text{VaR}_{B,0.05}$ because $\rho_{AB} \neq 1$.

Using R, these computations are:

```
w0 = 1e+05
VaR.A = (mu.A + sig.A * qnorm(0.05)) * w0
VaR.A

## [1] -24937
```

```
VaR.B = (mu.B + sig.B * qnorm(0.05)) * w0
VaR.B

## [1] -13416

VaR.p1 = (mu.p1 + sig.p1 * qnorm(0.05)) * w0
VaR.p1

## [1] -10268

VaR.p2 = (mu.p2 + sig.p2 * qnorm(0.05)) * w0
VaR.p2

## [1] -42371

x.A * VaR.A + x.B * VaR.B

## [1] -30698
```

■

Example 11.4. Create R function to compute normal VaR.

The previous example used repetitive R calculations to compute the 5% VaR of an investment. An alternative approach is to first create an R function to compute the VaR given μ , σ , α (VaR probability) and W_0 , and then apply the function using the inputs of the different assets. A simple function to compute VaR based on normally distributed asset returns is:

```
normalVaR <- function(mu, sigma, w0, tail.prob = 0.01, invert = FALSE) {
  ## compute normal VaR for collection of assets given mean and sd vector inputs:
  ## mu n x 1 vector of expected returns sigma n x 1 vector of standard deviations
  ## w0 scalar initial investment in $ tail.prob scalar tail probability invert
  ## logical. If TRUE report VaR as positive number output: VaR n x 1 vector of
  ## left tail return quantiles References: Jorian (2007) pg 111.
  if (length(mu) != length(sigma))
    stop("mu and sigma must have same number of elements")
  if (tail.prob < 0 || tail.prob > 1)
    stop("tail.prob must be between 0 and 1")
  VaR = w0 * (mu + sigma * qnorm(tail.prob))
  if (invert) {
    VaR = -VaR
  }
  return(VaR)
}
```

■

Using the `normalVaR()` function, the 5% VaR values of asset A , B and equally weighted portfolio are:

```
normalVaR(mu = c(mu.A, mu.B, mu.p1), sigma = c(sig.A, sig.B, sig.p1), w0 = 1e+05,
          tail.prob = 0.05)

## [1] -24937 -13416 -10268
```

■

11.2 Efficient portfolios with two risky assets

In this section we describe how mean-variance efficient portfolios are constructed. First we make the following assumptions regarding the probability distribution of asset returns and the behavior of investors:

1. Returns are covariance stationary and ergodic, and jointly normally distributed over the investment horizon. This implies that means, variances and covariances of returns are constant over the investment horizon and completely characterize the joint distribution of returns.
2. Investors know the values of asset return means, variances and covariances.
3. Investors only care about portfolio expected return and portfolio variance. Investors like portfolios with high expected return but dislike portfolios with high return variance.

Given the above assumptions we set out to characterize the set of *efficient portfolios*: those portfolios that have the highest expected return for a given level of risk as measured by portfolio variance. These are the portfolios that investors are most interested in holding.

For illustrative purposes we will show calculations using the data in the Table 11.1. The collection of all *feasible* portfolios, or the *investment possibilities set*, in the case of two assets is simply all possible portfolios that can be formed by varying the portfolio weights x_A and x_B such that the weights sum to one ($x_A + x_B = 1$). We summarize the expected return-risk (mean-variance) properties of the feasible portfolios in a plot with portfolio expected return, μ_p , on the vertical axis and portfolio standard deviation, σ_p , on the horizontal axis. The portfolio standard deviation is used instead of variance because standard deviation is measured in the same units as the expected value (recall, variance is the average squared deviation from the mean).

Example 11.5. Investment possibilities set for example data.

The investment possibilities set or portfolio frontier for the data in Table 11.1 is illustrated in Figure 11.2. Here the portfolio weight on asset A, x_A , is varied from -0.4 to 1.4 in increments of 0.1 and, since $x_B = 1 - x_A$, the weight on asset B then varies from 1.4 to -0.4. This gives us 18 portfolios with weights $(x_A, x_B) = (-0.4, 1.4), (-0.3, 1.3), \dots, (1.3, -0.3), (1.4, -0.4)$. For each of these portfolios we use the formulas (11.4) and (11.6) to compute μ_p and σ_p . We then plot these values. In R, the computations are:

```
x.A = seq(from = -0.4, to = 1.4, by = 0.1)
x.B = 1 - x.A
mu.p = x.A * mu.A + x.B * mu.B
sig2.p = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p = sqrt(sig2.p)
plot(sig.p, mu.p, type = "b", pch = 16, ylim = c(0, max(mu.p)), xlim = c(0, max(sig.p)),
     xlab = expression(sigma[p]), ylab = expression(mu[p]), col = c(rep("red", 6),
     rep("green", 13)))
plot(sig.p, mu.p, type = "b", pch = 16, ylim = c(0, max(mu.p)), xlim = c(0, max(sig.p)),
     xlab = expression(sigma[p]), ylab = expression(mu[p]), col = c(rep("red", 6),
     rep("green", 13)))
text(x = sig.A, y = mu.A, labels = "Asset A", pos = 4)
text(x = sig.B, y = mu.B, labels = "Asset B", pos = 4)
```

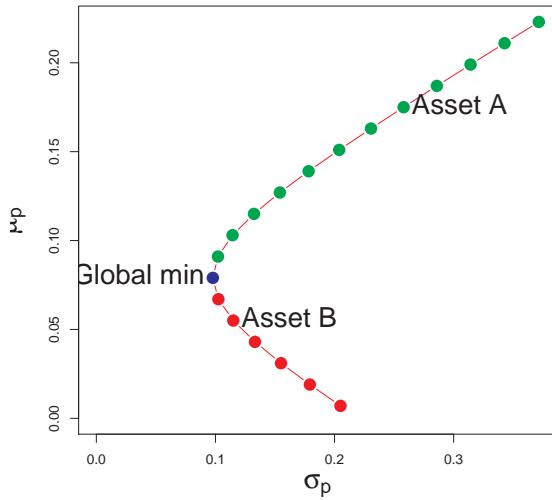


Fig. 11.2 Portfolio frontier of example data.

■

Notice that the plot in (μ_p, σ_p) -space looks like a parabola turned on its side (in fact, it is one side of a hyperbola). Since it is assumed that investors desire portfolios with the highest expected return, μ_p , for a given level of risk, σ_p , combinations that are in the upper left corner are the best portfolios and those in the lower right corner are the worst. Notice that the portfolio at the bottom of the parabola has the property that it has the smallest variance among all feasible portfolios. Accordingly, this portfolio is called the *global minimum variance* portfolio.

Efficient portfolios are those with the highest expected return for a given level of risk. These portfolios are colored green in Figure 11.2. *Inefficient portfolios* are then portfolios such that there is another feasible portfolio that has the same risk (σ_p) but a higher expected return (μ_p). These portfolios are colored red in Figure 11.2. From Figure 11.2 it is clear that the inefficient portfolios are the feasible portfolios that lie below the global minimum variance portfolio, and the efficient portfolios are those that lie above the global minimum variance portfolio.

11.2.1 Computing the Global Minimum Variance Portfolio

It is a simple exercise in calculus to find the global minimum variance portfolio. We solve the constrained optimization problem⁷:

⁷ A review of optimization and constrained optimization is given in the appendix to this chapter.

$$\begin{aligned} \min_{x_A, x_B} \sigma_p^2 &= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB} \\ \text{s.t. } x_A + x_B &= 1. \end{aligned}$$

This constrained optimization problem can be solved using two methods. The first method, called the *method of substitution*, uses the constraint to substitute out one of the variables to transform the constrained optimization problem in two variables into an unconstrained optimization problem in one variable. The second method, called the *method of Lagrange multipliers*, introduces an auxiliary variable called the Lagrange multiplier and transforms the constrained optimization problem in two variables into an unconstrained optimization problem in three variables.

The substitution method is straightforward. Substituting $x_B = 1 - x_A$ into the formula for σ_p^2 reduces the problem to:

$$\min_{x_A} \sigma_p^2 = x_A^2 \sigma_A^2 + (1 - x_A)^2 \sigma_B^2 + 2x_A(1 - x_A) \sigma_{AB}.$$

The first order conditions for a minimum, via the chain rule, are:

$$0 = \frac{d\sigma_p^2}{dx_A} = 2x_A^{\min} \sigma_A^2 - 2(1 - x_A^{\min}) \sigma_B^2 + 2\sigma_{AB}(1 - 2x_A^{\min}),$$

and straightforward calculations yield,

$$x_A^{\min} = \frac{\sigma_B^2 - \sigma_{AB}}{\sigma_A^2 + \sigma_B^2 - 2\sigma_{AB}}, \quad x_B^{\min} = 1 - x_A^{\min}. \quad (11.10)$$

The method of Lagrange multipliers involves two steps. In the first step, the constraint $x_A + x_B = 1$ is put into homogenous form $x_A + x_B - 1 = 0$. In the second step, the Lagrangian function is formed by adding to σ_p^2 the homogenous constraint multiplied by an auxiliary variable λ (the Lagrange multiplier) giving:

$$L(x_A, x_B, \lambda) = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB} + \lambda(x_A + x_B - 1).$$

This function is then minimized with respect to x_A , x_B , and λ . The first order conditions are:

$$\begin{aligned} 0 &= \frac{\partial L(x_A, x_B, \lambda)}{\partial x_A} = 2x_A \sigma_A^2 + 2x_B \sigma_{AB} + \lambda, \\ 0 &= \frac{\partial L(x_A, x_B, \lambda)}{\partial x_B} = 2x_B \sigma_B^2 + 2x_A \sigma_{AB} + \lambda, \\ 0 &= \frac{\partial L(x_A, x_B, \lambda)}{\partial \lambda} = x_A + x_B - 1. \end{aligned}$$

The first two equations can be rearranged to give:

$$x_B = x_A \left(\frac{\sigma_A^2 - \sigma_{AB}}{\sigma_B^2 - \sigma_{AB}} \right).$$

Substituting this value for x_B into the third equation and rearranging gives the solution (11.10).

Example 11.6. Global minimum variance portfolio for example data.

Using the data in Table 11.1 and (11.10) we have:

$$x_A^{\min} = \frac{0.01323 - (-0.004866)}{0.06656 + 0.01323 - 2(-0.004866)} = 0.2021, \quad x_B^{\min} = 0.7979.$$

The expected return, variance and standard deviation of this portfolio are:

$$\begin{aligned}\mu_p &= (0.2021) \cdot (0.175) + (0.7979) \cdot (0.055) = 0.07925 \\ \sigma_p^2 &= (0.2021)^2 \cdot (0.067) + (0.7979)^2 \cdot (0.013) \\ &\quad + 2 \cdot (0.2021)(0.7979)(-0.004875) \\ &= 0.00975 \\ \sigma_p &= \sqrt{0.00975} = 0.09782.\end{aligned}$$

In Figure 11.2, this portfolio is labeled “global min”. In R, the calculations to compute the global minimum variance portfolio are:

```
xA.min = (sig2.B - sig.AB)/(sig2.A + sig2.B - 2 * sig.AB)
xB.min = 1 - xA.min
xA.min

## [1] 0.2021

xB.min

## [1] 0.7979

mu.p.min = xA.min * mu.A + xB.min * mu.B
sig2.p.min = xA.min^2 * sig2.A + xB.min^2 * sig2.B + 2 * xA.min * xB.min * sig.AB
sig.p.min = sqrt(sig2.p.min)
mu.p.min

## [1] 0.07925

sig.p.min

## [1] 0.09782
```

11.2.2 Correlation and the Shape of the Efficient Frontier

The shape of the investment possibilities set is very sensitive to the correlation between assets A and B given the other parameters. If ρ_{AB} is close to 1 then the investment set approaches a straight line connecting the portfolio with all wealth invested in asset B ,

$(x_A, x_B) = (0, 1)$, to the portfolio with all wealth invested in asset A , $(x_A, x_B) = (1, 0)$. This case is illustrated in Figure 11.3. As ρ_{AB} approaches zero the set starts to bow toward the μ_p axis, and the power of diversification starts to kick in. If $\rho_{AB} = -1$ then the set actually touches the μ_p axis. What this means is that if assets A and B are perfectly negatively correlated then there exists a portfolio of A and B that has positive expected return and zero variance! To find the portfolio with $\sigma_p^2 = 0$ when $\rho_{AB} = -1$ we use (11.10) and the fact that $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B$ to give:

$$x_A^{\min} = \frac{\sigma_B}{\sigma_A + \sigma_B}, \quad x_B^{\min} = 1 - x_A.$$

The case with $\rho_{AB} = -1$ is also illustrated in Figure 11.3.

Example 11.7. Portfolio frontier when $\rho = \pm 1$.

Suppose $\rho_{AB} = 1$. Then $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B = \sigma_A\sigma_B$. The portfolio variance is then:

$$\begin{aligned}\sigma_p^2 &= x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_{AB} = x_A^2\sigma_A^2 + x_B^2\sigma_B^2 + 2x_Ax_B\sigma_A\sigma_B \\ &= (x_A\sigma_A + x_B\sigma_B)^2.\end{aligned}$$

Hence, $\sigma_p = x_A\sigma_A + x_B\sigma_B = x_A\sigma_A + (1 - x_A)\sigma_B$ which shows that σ_p lies on a straight line connecting σ_A and σ_B . Next, suppose $\rho_{AB} = -1$. Then $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B = -\sigma_A\sigma_B$ which implies that $\sigma_p^2 = (x_A\sigma_A - x_B\sigma_B)^2$ and $\sigma_p = x_A\sigma_A - x_B\sigma_B = x_A\sigma_A + (1 - x_A)\sigma_B$. In this case we can find a portfolio that has zero volatility. We solve for x_A such that $\sigma_p = 0$:

$$0 = x_A\sigma_A + (1 - x_A)\sigma_B \Rightarrow x_A = \frac{\sigma_B}{\sigma_A + \sigma_B}, \quad x_B = 1 - x_A.$$

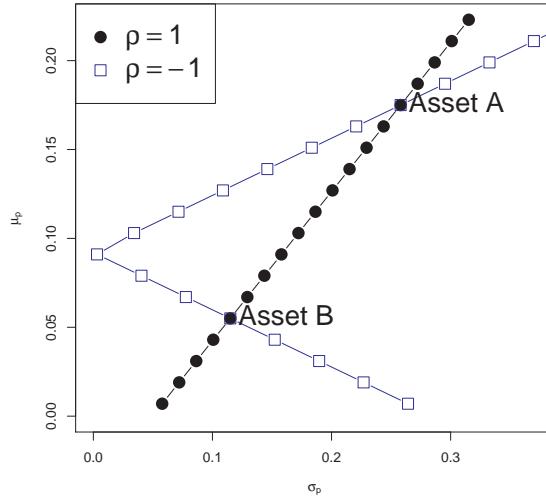


Fig. 11.3 Portfolios with $\rho_{AB} = 1$ and $\rho_{AB} = -1$.

11.2.3 Optimal Portfolios

Given the efficient set of portfolios as described in Figure 11.2, which portfolio will an investor choose? Of the efficient portfolios, investors will choose the one that accords with their risk preferences. Very risk averse investors will want a portfolio that has low volatility (risk) and will choose a portfolio very close to the global minimum variance portfolio. In contrast, very risk tolerant investors will ignore volatility and seek portfolios with high expected returns. Hence, these investors will choose portfolios with large amounts of asset *A* which may involve short-selling asset *B*.

11.3 Efficient portfolios with a risk-free asset

In the preceding section we constructed the efficient set of portfolios in the absence of a risk-free asset. Now we consider what happens when we introduce a risk-free asset. In the present context, a risk-free asset is equivalent to default-free pure discount bond that matures at the end of the assumed investment horizon.⁸ The risk-free rate, r_f , is then the nominal return on the bond. For example, if the investment horizon is one month then the risk-free asset is a 30-day U.S. Treasury bill (T-bill) and the risk-free rate is the nominal rate of return on the T-bill.⁹ If our holdings of the risk-free asset is positive then we are “lending money” at the risk-free rate, and if our holdings are negative then we are “borrowing” at the risk-free rate.

11.3.1 Efficient portfolios with one risky asset and one risk-free asset

Consider an investment in asset *B* and the risk-free asset (henceforth referred to as a T-bill). Since the risk-free rate is fixed (constant) over the investment horizon it has some special properties, namely:

$$\begin{aligned}\mu_f &= E[r_f] = r_f, \\ \text{var}(r_f) &= 0, \\ \text{cov}(R_B, r_f) &= 0.\end{aligned}$$

Let x_B denote the share of wealth in asset *B*, and $x_f = 1 - x_B$ denote the share of wealth in T-bills. The portfolio return is:

$$R_p = (1 - x_B)r_f + x_B R_B = r_f + x_B(R_B - r_f).$$

⁸ In the finance industry the risk-free asset is often referred to as “cash”. In general, “cash” doesn’t refer to holding dollars in your pocket but rather investing dollars in a very low risk asset that pays a stable rate of return.

⁹ The default-free assumption of U.S. debt has recently been questioned due to the inability of the U.S. congress to address the long-term debt problems of the U.S. government.

The quantity $R_B - r_f$ is called the *excess return* (over the return on T-bills) on asset B . The portfolio expected return is then:

$$\mu_p = r_f + x_B(E[R_B] - r_f) = r_f + x_B(\mu_B - r_f), \quad (11.11)$$

where the quantity $(\mu_B - r_f)$ is called the *expected excess return* or *risk premium* on asset B . For risky assets, the risk premium is typically positive indicating that investors expect a higher return on the risky asset than the safe asset (otherwise, why would investors hold the risky asset?). We may express the risk premium on the portfolio in terms of the risk premium on asset B :

$$\mu_p - r_f = x_B(\mu_B - r_f).$$

The more we invest in asset B the higher the risk premium on the portfolio.

Because the risk-free rate is constant, the portfolio variance only depends on the variability of asset B and is given by:

$$\sigma_p^2 = x_B^2 \sigma_B^2.$$

The portfolio standard deviation is therefore proportional to the standard deviation on asset B :¹⁰

$$\sigma_p = x_B \sigma_B, \quad (11.12)$$

which we can use to solve for x_B ,

$$x_B = \frac{\sigma_p}{\sigma_B}.$$

Using the last result, the feasible (and efficient) set of portfolios follows the equation:

$$\mu_p = r_f + \frac{\mu_B - r_f}{\sigma_B} \cdot \sigma_p, \quad (11.13)$$

which is simply straight line in (μ_p, σ_p) -space with intercept r_f and slope $(\mu_B - r_f)/\sigma_B$. This line is often called the *capital allocation line* (CAL). The slope of the CAL is called the *Sharpe ratio* (SR) or *Sharpe's slope* (named after the economist William Sharpe), and it measures the risk premium on the asset per unit of risk (as measured by the standard deviation of the asset). From (11.13) we have

$$\frac{d\mu_p}{d\sigma_p} = \frac{\mu_B - r_f}{\sigma_B} = SR_B.$$

Example 11.8. Portfolios of T-Bills and risky assets.

The portfolios which are combinations of asset A and T-bills and combinations of asset B and T-bills, using the data in Table 11.1 with $r_f = 0.03$, are illustrated in Figure 11.4 which is created using the R code:

```
r.f = 0.03
# T-bills + asset A
x.A = seq(from = 0, to = 1.4, by = 0.1)
mu.p.A = r.f + x.A * (mu.A - r.f)
```

¹⁰ Here, we can allow x_B to be positive or negative by σ_B is always positive.

```

sig.p.A = x.A * sig.A
sharpe.A = (mu.A - r.f)/sig.A
sharpe.A

## [1] 0.562

# T-bills + asset B
x.B = seq(from = 0, to = 1.4, by = 0.1)
mu.p.B = r.f + x.B * (mu.B - r.f)
sig.p.B = x.B * sig.B
sharpe.B = (mu.B - r.f)/sig.B
sharpe.B

## [1] 0.2174

# plot portfolios of T-Bills and assets A and B
plot(sig.p.A, mu.p.A, type = "b", col = "green", ylim = c(0, max(mu.p.A)), xlim = c(0,
  max(sig.p.A, sig.p.B)), pch = 16, cex = cex.val, xlab = expression(sigma[p]),
  ylab = expression(mu[p]), cex.lab = cex.val)
points(sig.p.B, mu.p.B, type = "b", col = "red", pch = 16, cex = cex.val)
text(x = sig.A, y = mu.A, labels = "Asset A", pos = 4, cex = cex.val)
text(x = sig.B, y = mu.B, labels = "Asset B", pos = 1, cex = cex.val)
text(x = 0, y = r.f, labels = expression(r[f]), pos = 2, cex = cex.val)

```

Notice that expected return-risk trade off of these portfolios is linear. In Figure 11.4 the point labeled “rf” represents a portfolio that is 100% invested in T-bills. The points labeled “Asset A” and “Asset B” represent portfolios that are 100% invested in assets A and B, respectively. A point half-way between “rf” and “Asset A” is a portfolio that is 50% invested in T-bills and 50% invested in asset A. A point above and to the right of “Asset A” is a levered portfolio where more than 100% of asset A is financed by borrowing at the T-bill rate. The portfolios which are combinations of asset A and T-bills have expected returns uniformly higher than the portfolios consisting of asset B and T-bills. This occurs because the Sharpe ratio for asset A is higher than the ratio for asset B:

$$\begin{aligned} \text{SR}_A &= \frac{\mu_A - r_f}{\sigma_A} = \frac{0.175 - 0.03}{0.258} = 0.562, \\ \text{SR}_B &= \frac{\mu_B - r_f}{\sigma_B} = \frac{0.055 - 0.03}{0.115} = 0.217. \end{aligned}$$

Hence, portfolios of asset A and T-bills are efficient relative to portfolios of asset B and T-bills.

The previous example shows that the Sharpe ratio can be used to rank the risk return properties of individual assets. Assets with a high Sharpe ratio have a better risk-return tradeoff than assets with a low Sharpe ratio. Accordingly, investment analysts routinely rank assets based on their Sharpe ratios.

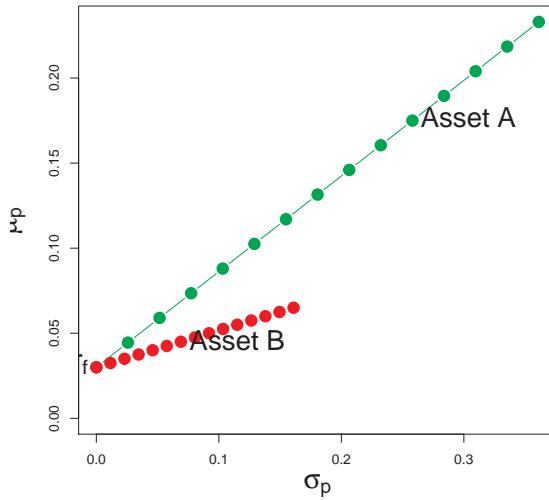


Fig. 11.4 Portfolios of T-Bills and risky assets.

11.4 Efficient portfolios with two risky assets and a risk-free asset

Now we expand on the previous results by allowing our investor to form portfolios of assets A , B and T-bills. The efficient set in this case will still be a straight line in (μ_p, σ_p) -space with intercept r_f . The slope of the efficient set, the maximum Sharpe ratio, is such that it is tangent to the efficient set constructed just using the two risky assets A and B . Figure 11.5 illustrates why this is so.

If we invest in only in asset B and T-bills then the Sharpe ratio is $SR_B = \frac{\mu_B - r_f}{\sigma_B} = 0.217$ and the CAL intersects the set of risky asset portfolios at point B . This is clearly not the efficient set of portfolios. For example, we could do uniformly better if we instead invest only in asset A and T-bills. This gives us a Sharpe ratio of $SR_A = \frac{\mu_A - r_f}{\sigma_A} = 0.562$, and the new CAL intersects the set of risky asset portfolios at point A . However, we could do better still if we invest in T-bills and some combination of assets A and B . Geometrically, it is easy to see that the best we can do is obtained for the combination of assets A and B such that the CAL is just tangent to the set of risky asset portfolios. This point is labeled "Tangency" on the graph and represents the *tangency portfolio* of assets A and B . Portfolios of T-Bills and the tangency portfolio are the set of efficient portfolios consisting of T-Bills, asset A and asset B .

11.4.1 Solving for the Tangency Portfolio

We can determine the proportions of each asset in the tangency portfolio by finding the values of x_A and x_B that maximize the Sharpe ratio of a portfolio that is on the envelope of the parabola. Formally, we solve the constrained maximization problem:

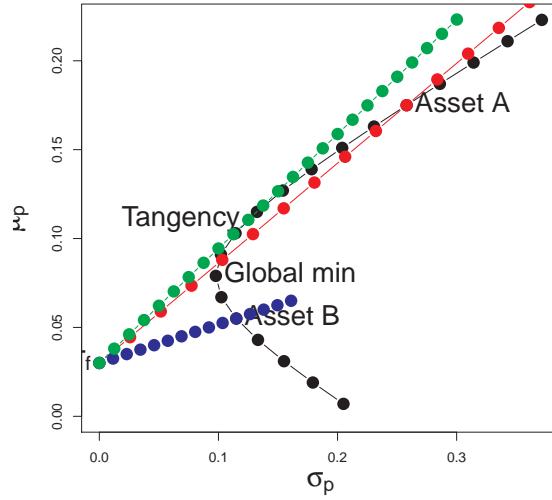


Fig. 11.5 Efficient portfolios of two risky assets and T-bills. Black dots represent portfolios of assets A and B; blue dots represent portfolios of T-bills and asset B; red dots represent portfolios of T-bills and asset A; green dots represent portfolios of the tangency portfolio and T-bills.

$$\begin{aligned} \max_{x_A, x_B} \text{SR}_p &= \frac{\mu_p - r_f}{\sigma_p} \text{ s.t.} \\ \mu_p &= x_A \mu_A + x_B \mu_B, \\ \sigma_p^2 &= x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}, \\ 1 &= x_A + x_B. \end{aligned}$$

After various substitutions, the above problem can be reduced to:

$$\max_{x_A} \frac{x_A(\mu_A - r_f) + (1 - x_A)(\mu_B - r_f)}{(x_A^2 \sigma_A^2 + (1 - x_A)^2 \sigma_B^2 + 2x_A(1 - x_A)\sigma_{AB})^{1/2}}.$$

This is a straightforward, albeit very tedious, calculus problem and the solution can be shown to be:

$$\begin{aligned} x_T^A &= \frac{(\mu_A - r_f)\sigma_B^2 - (\mu_B - r_f)\sigma_{AB}}{(\mu_A - r_f)\sigma_B^2 + (\mu_B - r_f)\sigma_A^2 - (\mu_A - r_f + \mu_B - r_f)\sigma_{AB}}, \\ x_T^B &= 1 - x_T^A. \end{aligned} \quad (11.14)$$

Example 11.9. Tangency portfolio for example data.

For the example data in Table 11.1 using (11.14) with $r_f = 0.03$, we get $x_T^A = 0.4625$ and $x_T^B = 0.5375$. The expected return, variance, standard deviation, and Sharpe ratio of the tangency portfolio are:

$$\begin{aligned}
\mu_T &= x_T^A \mu_A + x_T^B \mu_B \\
&= (0.4625)(0.175) + (0.5375)(0.055) = 0.1105, \\
\sigma_T^2 &= (x_T^A)^2 \sigma_A^2 + (x_T^B)^2 \sigma_B^2 + 2x_T^A x_T^B \sigma_{AB} \\
&= (0.4625)^2(0.06656) + (0.5375)^2(0.01323) + \\
&\quad 2(0.4625)(0.5375)(-0.004866) = 0.01564, \\
\sigma_T &= \sqrt{0.01564} = 0.1251, \\
SR_T &= \frac{\mu_T - r_f}{\sigma_T} = \frac{0.1105 - 0.03}{0.1251} = 0.6434
\end{aligned}$$

In R, the computations to compute the tangency portfolio are:

```

top = (mu.A - r.f) * sig2.B - (mu.B - r.f) * sig.AB
bot = (mu.A - r.f) * sig2.B + (mu.B - r.f) * sig2.A - (mu.A - r.f + mu.B - r.f) *
      sig.AB
x.A.tan = top/bot
x.B.tan = 1 - x.A.tan
x.A.tan

## [1] 0.4625

x.B.tan

## [1] 0.5375

mu.p.tan = x.A.tan * mu.A + x.B.tan * mu.B
sig2.p.tan = x.A.tan^2 * sig2.A + x.B.tan^2 * sig2.B + 2 * x.A.tan * x.B.tan *
              sig.AB
sig.p.tan = sqrt(sig2.p.tan)
SR.tan = (mu.p.tan - r.f)/sig.p.tan
mu.p.tan

## [1] 0.1105

sig.p.tan

## [1] 0.1251

SR.tan

## [1] 0.6437

```

■

11.4.2 Mutual Fund Separation

The efficient portfolios are combinations of the tangency portfolio and the T-bill. Accordingly, using (11.11) and (11.12) the expected return and standard deviation of any efficient portfolio are given by:

$$\mu_p^e = r_f + x_T(\mu_T - r_f), \quad (11.15)$$

$$\sigma_p^e = x_T\sigma_T, \quad (11.16)$$

where x_T represents the fraction of wealth invested in the tangency portfolio (and $1 - x_T$ represents the fraction of wealth invested in T-Bills), and μ_T and σ_T are the expected return and standard deviation of the tangency portfolio, respectively. This important result is known as the *mutual fund separation theorem*. The tangency portfolio can be considered as a mutual fund of the two risky assets, where the shares of the two assets in the mutual fund are determined by the tangency portfolio weights (x_T^A and x_T^B determined from (11.14)), and the T-bill can be considered as a mutual fund of risk-free assets. The expected return-risk trade-off of these portfolios is given by the line connecting the risk-free rate to the tangency point on the efficient frontier of risky asset only portfolios. Which combination of the tangency portfolio and the T-bill an investor will choose depends on the investor's risk preferences. If the investor is very risk averse, then she will choose a portfolio with low volatility which will be a portfolio with very little weight in the tangency portfolio and a lot of weight in the T-bill. This will produce a portfolio with an expected return close to the risk-free rate and a variance that is close to zero. If the investor can tolerate a large amount of risk, then she would prefer a portfolio with highest expected return regardless of the volatility. This portfolio may involve borrowing at the risk-free rate (leveraging) and investing the proceeds in the tangency portfolio to achieve a high expected return.

Example 11.10. Efficient portfolios chosen by risk averse and risk tolerant investors.

A highly risk averse investor may choose to put 10% of her wealth in the tangency portfolio and 90% in the T-bill. Then she will hold $(10\%) \times (46.25\%) = 4.625\%$ of her wealth in asset A , $(10\%) \times (53.75\%) = 5.375\%$ of her wealth in asset B , and 90% of her wealth in the T-bill. The expected return on this portfolio is:

$$\mu_p^e = r_f + 0.10(\mu_T - r_f) = 0.03 + 0.10(0.1105 - 0.03) = 0.03805,$$

and the standard deviation is,

$$\sigma_p^e = 0.10\sigma_T = 0.10(0.1251) = 0.01251.$$

In Figure 11.6, this efficient portfolio is labeled "Safe". A very risk tolerant investor may actually borrow at the risk-free rate and use these funds to leverage her investment in the tangency portfolio. For example, suppose the risk tolerant investor borrows 100% of her wealth at the risk-free rate and uses the proceed to purchase 200% of her wealth in the tangency portfolio. Then she would hold $(200\%) \times (46.25\%) = 92.50\%$ of her wealth in asset A , $(200\%) \times (53.75\%) = 107.5\%$ in asset B , and she would owe 100% of her wealth to her lender. The expected return and standard deviation on this portfolio is:

$$\mu_p^e = 0.03 + 2(0.1105 - 0.03) = 0.1910,$$

$$\sigma_p^e = 2(0.1251) = 0.2501.$$

In Figure 11.6, this efficient portfolio is labeled "Risky".



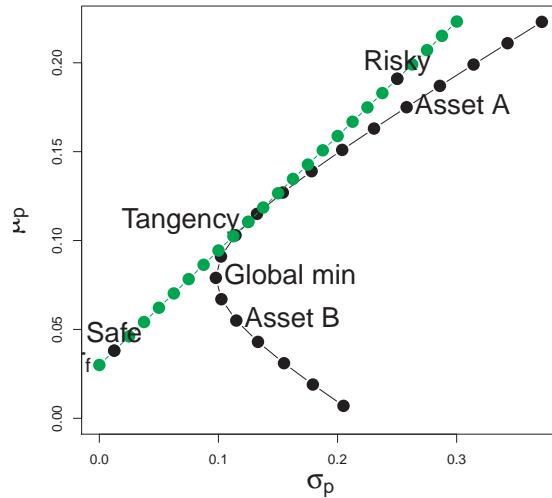


Fig. 11.6 The efficient portfolio labeled “safe” has 10% invested in the tangency portfolio and 90% invested in T-Bills; the efficient portfolio labeled “risky” has 200% invested in the tangency portfolio and -100% invested in T-Bills.

11.4.3 Interpreting Efficient Portfolios

As we have seen, efficient portfolios are those portfolios that have the highest expected return for a given level of risk as measured by portfolio standard deviation. For portfolios with expected returns above the T-bill rate, efficient portfolios can also be characterized as those portfolios that have minimum risk (as measured by portfolio standard deviation) for a given target expected return.

To illustrate, consider Figure 11.7 which shows the portfolio frontier for two risky assets and the efficient frontier for two risky assets plus T-Bills. Suppose an investor initially holds all of his wealth in asset B . The expected return on this portfolio is $\mu_B = 0.055$, and the standard deviation (risk) is $\sigma_B = 0.115$. An efficient portfolio (combinations of the tangency portfolio and T-bills) that has the same standard deviation (risk) as asset B is given by the portfolio on the efficient frontier that is directly above $\sigma_B = 0.115$. To find the shares in the tangency portfolio and T-bills in this portfolio recall from (11.16) that the standard deviation of an efficient portfolio with x_T invested in the tangency portfolio and $1 - x_T$ invested in T-bills is $\sigma_p^e = x_T \sigma_T$. Since we want to find the efficient portfolio with $\sigma_p^e = \sigma_B = 0.115$, we solve:

$$x_T = \frac{\sigma_B}{\sigma_T} = \frac{0.115}{0.1251} = 0.9195, \quad x_f = 1 - x_T = 0.08049.$$

That is, if we invest 91.95% of our wealth in the tangency portfolio and 8.049% in T-bills we will have a portfolio with the same standard deviation as asset B . Since this is an efficient portfolio, the expected return should be higher than the expected return on asset B . Indeed it is since:

$$\mu_p^e = r_f + x_T(\mu_T - r_f) = 0.03 + 0.9195(0.1105 - 0.03) = 0.1040.$$

Notice that by diversifying our holding into assets A , B and T-bills we can obtain a portfolio with the same risk as asset B but with almost twice the expected return! By moving from asset B into an efficient portfolio we raise the Sharpe ratio of the investment from $SR_B = 0.217$ to $SR_T = 0.6434$.

Next, consider finding an efficient portfolio that has the same expected return as asset B . Visually, this involves finding the combination of the tangency portfolio and T-bills that corresponds with the intersection of a horizontal line with intercept $\mu_B = 0.055$ and the line representing efficient combinations of T-bills and the tangency portfolio. To find the shares in the tangency portfolio and T-bills in this portfolio recall from (11.15) that the expected return of an efficient portfolio with x_T invested in the tangency portfolio and $1-x_T$ invested in T-bills has expected return equal to $\mu_p^e = r_f + x_T(\mu_T - r_f)$. Since we want to find the efficient portfolio with $\mu_p^e = \mu_B = 0.055$ we solve:

$$x_T = \frac{\mu_p^e - r_f}{\mu_T - r_f} = \frac{0.055 - 0.03}{0.1105 - 0.03} = 0.3105, \quad x_f = 1 - x_T = 0.6895.$$

That is, if we invest 31.05% of wealth in the tangency portfolio and 68.95% of our wealth in T-bills we have a portfolio with the same expected return as asset B . Since this is an efficient portfolio, the standard deviation (risk) of this portfolio should be lower than the standard deviation on asset B . Indeed it is since:

$$\sigma_p^e = x_T \sigma_T = 0.3105(0.124) = 0.03884.$$

Notice how large the risk reduction is by forming an efficient portfolio. The standard deviation on the efficient portfolio is almost three times smaller than the standard deviation of asset B ! Again, by moving from asset B into an efficient portfolio we raise the Sharpe ratio of the investment from $SR_B = 0.217$ to $SR_T = 0.6434$.

The above example illustrates two ways to interpret the benefits from forming efficient portfolios. Starting from some benchmark portfolio, we can fix standard deviation (risk) at the value for the benchmark and then determine the gain in expected return from forming a diversified portfolio¹¹. The gain in expected return has concrete meaning. Alternatively, we can fix expected return at the value for the benchmark and then determine the reduction in standard deviation (risk) from forming a diversified portfolio. The meaning to an investor of the reduction in standard deviation is not as clear as the meaning to an investor of the increase in expected return. It would be helpful if the risk reduction benefit can be translated into a number that is more interpretable than the standard deviation. The concept of Value-at-Risk (VaR) provides such a translation.

¹¹ The gain in expected return by investing in an efficient portfolio abstracts from the costs associated with selling the benchmark portfolio and buying the efficient portfolio.

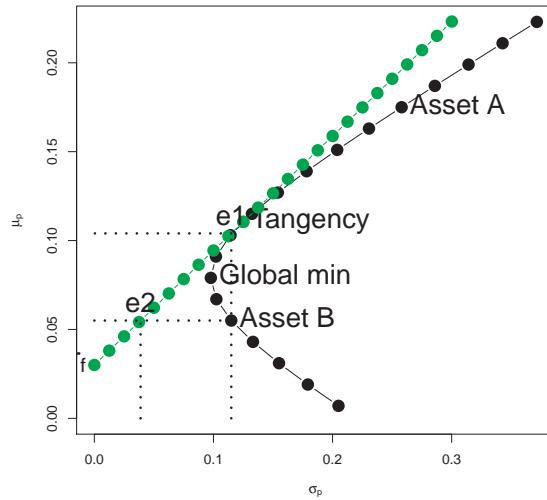


Fig. 11.7 The point “e1” represents an efficient portfolio with the same standard deviation as asset B ; the point “e2” represents an efficient portfolio with the same expected returns as asset B .

11.4.4 Efficient Portfolios and Value-at-Risk

Recall, the VaR of an investment is the (lower bound of) loss in investment value over a given horizon with a stated probability. For example, consider an investor who invests $W_0 = \$100,000$ in asset B over the next year. Assuming that $R_B \sim N(0.055, (0.115)^2)$ represents the annual simple return on asset B , the 5% VaR is:

$$\text{VaR}_{B,0.05} = q_{0.05}^{R_B} W_0 = (0.055 + 0.115(-1.645)) \cdot \$100,000 = -\$13,416.$$

If an investor holds \$100,000 in asset B over the next year, then there is a 5% probability that he will lose \$13,416 or more.

Now suppose the investor chooses to hold an efficient portfolio with the same expected return as asset B . This portfolio consists of 31.05% in the tangency portfolio and 68.95% in T-bills and has a standard deviation equal to 0.03884. Then $R_p \sim N(0.055, 0.03884)$ and the 5% VaR on the portfolio is:

$$\text{VaR}_{p,0.05} = q_{0.05}^{R_p} W_0 = (0.055 + 0.03884(-1.645)) \cdot \$100,000 = -\$884.$$

Notice that the 5% VaR for the efficient portfolio is almost fifteen times smaller than the 5% VaR for the investment in asset B . Since VaR translates risk into a dollar figure, it is more interpretable than standard deviation.

11.5 Application to Real World Portfolios

The portfolio theory for two risky assets and a risk-free asset developed in this chapter is not as limited as you might think if you realize that a single risky asset can also be a portfolio of many assets, such as a mutual fund of stocks or bonds, or an exchange traded fund (ETF) of stocks or bonds. In particular, a very common asset allocation for retirement portfolios is a simple two asset portfolio of in which one risky asset is a mutual fund of stocks and the other (less) risky asset is a mutual fund of U.S. Treasury bonds.

Example 11.11. Application of portfolio theory to simple portfolio of bonds, stocks, and T-Bills.

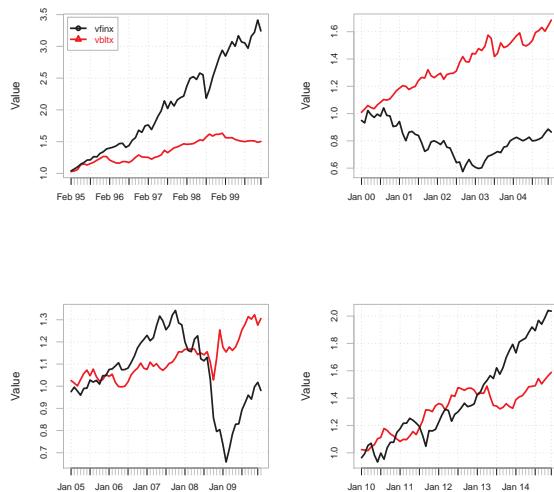


Fig. 11.8 Equity curves for stock (vfinx) and bond (vbltx) portfolios over sub-periods.

In the **IntroCompFinR** package there is a data object called **VanguardPrices** that contains monthly adjusted closing prices on six Vanguard mutual funds over the period January, 1995 through December, 2014.¹² The fund VFINX is a low cost fund designed to track the performance of the S&P 500 index. We will use this as our stock portfolio. The fund VBLTX is a low cost fund constructed to track the performance of a portfolio of long term (maturities greater than 10 years) U.S. Government Bonds. We will use this as our bond portfolio. We will use the U.S. 3-month T-Bill rate as the risk-free asset. The following code creates the monthly simple return data for this example:

```
library(IntroCompFinR)
library(PerformanceAnalytics)
data(VanguardPrices)
```

¹² Details about these funds are available at finance.yahoo.com.

```
VanguardPrices = as.xts(VanguardPrices)
vfinxPrices = VanguardPrices[, "vfinx", drop = FALSE]
vbltxPices = VanguardPrices[, "vbltx", drop = FALSE]
vfinxRetS = na.omit(Return.calculate(vfinxPrices, method = "simple"))
vbltxRetS = na.omit(Return.calculate(vbltxPices, method = "simple"))
stockBondRetS = merge(vfinxRetS, vbltxRetS)
head(stockBondRetS, n = 3)

##          vfinx    vbltx
## Feb 1995 0.03894 0.02797
## Mar 1995 0.02954 0.01020
## Apr 1995 0.02931 0.02020
```

We will construct efficient portfolios using the CER model estimates of expected returns, variances, and covariances over four sub-periods: January, 1995 to January 2000; January, 2000 to January, 2005; January, 2005 to January, 2010; January, 2010 to December, 2014. The approximate annualized T-Bill rates at the end of these periods are 0.05, 0.025, 0.001, and 0.001, respectively.¹³

```
# Define sub-periods
smpl1 = "1995-1::2000-1"
smpl2 = "2000-1::2005-1"
smpl3 = "2005-1::2010-1"
smpl4 = "2010-1::2014-12"
# Monthly risk-free rates for sub-periods
r.f = c(0.05, 0.025, 0.001, 0.001)/12
```

Historical performance of the stock and bond portfolios over the four sub-periods is illustrated in Figure 11.8, created using:

```
par(mfrow = c(2, 2))
chart.CumReturns(stockBondRetS[smpl1], main = "", wealth.index = TRUE, legend.loc = "topleft")
chart.CumReturns(stockBondRetS[smpl2], main = "", wealth.index = TRUE)
chart.CumReturns(stockBondRetS[smpl3], main = "", wealth.index = TRUE)
chart.CumReturns(stockBondRetS[smpl4], main = "", wealth.index = TRUE)
par(mfrow = c(1, 1))
```

The performance of stocks and bonds varies substantially over the four sub-periods. In the first and fourth sub-periods, which cover the dot-com boom and the recovery from the 2009 financial crisis, stocks substantially outperformed bonds. However, in the second and third sub-periods, which cover the dot-com bust and the 2009 financial crisis, bonds outperformed stocks.

The CER model estimates of the monthly expected returns, volatilities, and Sharpe ratios on the stock and bond portfolios are computed and annualized using the square-root-of-time rule as follows:

```
# smpl1 = '1995-1::2000-1'
table.AnnualizedReturns(stockBondRetS[smpl1], Rf = r.f[1], geometric = FALSE)
```

¹³ Historical data on U.S. T-Bill rates is obtained from the Federal Reserve Economic Data (FRED) available at the St. Louis Fed.

```

##                                     vfinx  vbltx
## Annualized Return          0.2479  0.0846
## Annualized Std Dev         0.1433  0.0756
## Annualized Sharpe (Rf=5%) 1.3806  0.4581

# smpl2 = '2000-1::2005-1'


```

The annualized estimates confirm the relative performance of bonds and stocks in the four sub-periods. The sample correlations between the monthly stock and bond returns, by sub-period, are:

```

cor(stockBondRetS[smpl1])[1, 2]

## [1] 0.2004

cor(stockBondRetS[smpl2])[1, 2]

## [1] -0.1846

cor(stockBondRetS[smpl3])[1, 2]

## [1] 0.2403

cor(stockBondRetS[smpl4])[1, 2]

## [1] -0.4354

```

Interestingly, the estimated correlation between stocks and bonds also varies by sub-period. In sub-periods one and three, the estimated correlations are positive, and, in sub-periods two and four, the estimated correlations are negative.

Using the sub-period CER model estimates, we compute risky asset only efficient portfolios and efficient portfolios of T-Bills (risk-free asset) and the tangency portfolios. The resulting portfolios are illustrated in Figure 11.9. In sub-periods one and four, stocks are on the risky asset only efficient frontier and, in sub-periods two and three, bonds are on the risky asset only efficient frontier. The sub-period global minimum variance and tangency portfolios are listed in Table 11.5. The global minimum variance portfolios are similar across sub-periods, with higher weights in bonds than stocks. The tangency portfolios, however, are quite different across sub-periods. In sub-periods one and four, stocks have weights 72% and 42%, respectively. In sub-periods two and three bonds essentially have 100% weights, respectively.

Portfolio:	Global Min	Global Min	Tangency	Tangency
Sub-Period	Stocks	Bonds	Stocks	Bonds
1995-2000	0.165	0.835	0.718	0.282
2000-2005	0.265	0.735	0.0034	0.9966
2005-2010	0.227	0.773	-0.0042	1.0042
2010-2014	0.365	0.635	0.417	0.583

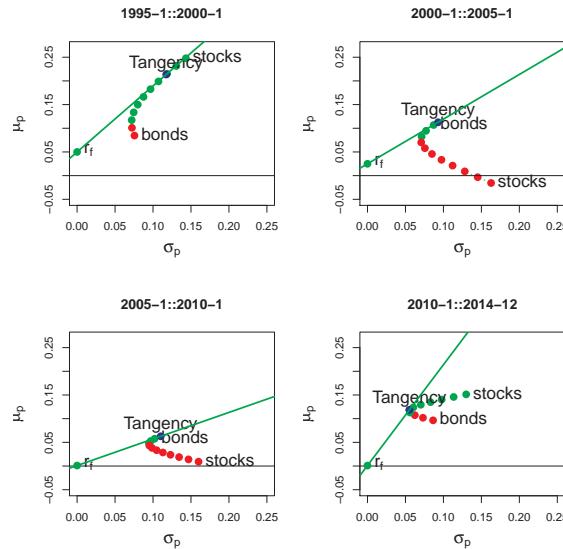


Fig. 11.9 Mean-variance efficient portfolios of stocks and bonds by sub-period. Efficient portfolios are shown in green and inefficient portfolios are shown in red.

11.6 Further Reading

Markowitz's mean-variance portfolio theory is described in detail in Markowitz (1971, 1987, 1991). Most MBA level corporate finance and investments textbooks have chapters on mean-variance portfolio theory. Good intermediate level treatments of portfolio theory from a quantitative perspective are given in Benninga (2000), Bodie et al. (2013) and Elton et al. (2014).

11.7 Appendix: Review of Optimization and Constrained Optimization

Consider the function of a single variable:

$$y = f(x) = x^2.$$

Clearly the minimum of this function occurs at the point $x = 0$. Using calculus, we find the minimum by solving:

$$\min_x y = x^2.$$

The first order (necessary) condition for a minimum is:

$$0 = \frac{d}{dx} f(x) = \frac{d}{dx} x^2 = 2x$$

and solving for x gives $x = 0$. The second order condition for a minimum is,

$$0 < \frac{d^2}{dx^2} f(x),$$

and this condition is clearly satisfied for $f(x) = x^2$.

Next, consider the function of two variables:

$$y = f(x, z) = x^2 + z^2 \tag{11.17}$$

This function looks like a salad bowl whose bottom is at $x = 0$ and $z = 0$. To find the minimum of (11.17), we solve:

$$\min_{x,z} y = x^2 + z^2,$$

and the first order necessary conditions are,

$$\begin{aligned} 0 &= \frac{\partial y}{\partial x} = 2x, \\ 0 &= \frac{\partial y}{\partial z} = 2z. \end{aligned}$$

Solving these two linear equations gives $x = 0$ and $z = 0$.

Now suppose we want to minimize (11.17) subject to the linear constraint:

$$x + z = 1. \quad (11.18)$$

The minimization problem is now a *constrained minimization*:

$$\begin{aligned} \min_{x,z} y &= x^2 + z^2 \text{ subject to (s.t.)} \\ x + z &= 1. \end{aligned}$$

Given the constraint $x + z = 1$, the function (11.17) is no longer minimized at the point $(x, z) = (0, 0)$ because this point does not satisfy $x + z = 1$. One simple way to solve this problem is to substitute the restriction (11.18) into the function (11.17) and reduce the problem to a minimization over one variable. To illustrate, use the restriction (11.18) to solve for z as:

$$z = 1 - x. \quad (11.19)$$

Now substitute (11.19) into (11.17) giving:

$$y = f(x, z) = f(x, 1 - x) = x^2 + (1 - x)^2. \quad (11.20)$$

The function (11.20) satisfies the restriction (11.18) by construction. The constrained minimization problem now becomes:

$$\min_x y = x^2 + (1 - x)^2.$$

The first order conditions for a minimum are:

$$0 = \frac{d}{dx}(x^2 + (1 - x)^2) = 2x - 2(1 - x) = 4x - 2,$$

and solving for x gives $x = 1/2$. To solve for z , use (11.19) to give $z = 1 - (1/2) = 1/2$. Hence, the solution to the constrained minimization problem is $(x, z) = (1/2, 1/2)$.

Another way to solve the constrained minimization is to use the method of *Lagrange multipliers*. This method augments the function to be minimized with a linear function of the constraint in homogeneous form. The constraint (11.18) in homogenous form is:

$$x + z - 1 = 0.$$

The augmented function to be minimized is called the *Lagrangian* and is given by:

$$L(x, z, \lambda) = x^2 + z^2 - \lambda(x + z - 1).$$

The coefficient on the constraint in homogeneous form, λ , is called the *Lagrange multiplier*. It measures the cost, or shadow price, of imposing the constraint relative to the unconstrained problem. The constrained minimization problem to be solved is now:

$$\min_{x,z,\lambda} L(x, z, \lambda) = x^2 + z^2 + \lambda(x + z - 1).$$

The first order conditions for a minimum are:

$$\begin{aligned} 0 &= \frac{\partial L(x, z, \lambda)}{\partial x} = 2x + \lambda, \\ 0 &= \frac{\partial L(x, z, \lambda)}{\partial z} = 2z + \lambda, \\ 0 &= \frac{\partial L(x, z, \lambda)}{\partial \lambda} = x + z - 1. \end{aligned}$$

The first order conditions give three linear equations in three unknowns. Notice that the first order condition with respect to λ imposes the constraint. The first two conditions give:

$$2x = 2z = -\lambda,$$

or,

$$x = z.$$

Substituting $x = z$ into the third condition gives:

$$2z - 1 = 0$$

or,

$$z = 1/2.$$

The final solution is $(x, y, \lambda) = (1/2, 1/2, -1)$.

The Lagrange multiplier, λ , measures the marginal cost, in terms of the value of the objective function, of imposing the constraint. Here, $\lambda = -1$ which indicates that imposing the constraint $x + z = 1$ reduces the objective function. To understand the roll of the Lagrange multiplier better, consider imposing the constraint $x + z = 0$. Notice that the unconstrained minimum achieved at $x = 0, z = 0$ satisfies this constraint. Hence, imposing $x + z = 0$ does not cost anything and so the Lagrange multiplier associated with this constraint should be zero. To confirm this, we solve the problem:

$$\min_{x, z, \lambda} L(x, z, \lambda) = x^2 + z^2 + \lambda(x + z - 0).$$

The first order conditions for a minimum are:

$$\begin{aligned} 0 &= \frac{\partial L(x, z, \lambda)}{\partial x} = 2x - \lambda, \\ 0 &= \frac{\partial L(x, z, \lambda)}{\partial z} = 2z - \lambda, \\ 0 &= \frac{\partial L(x, z, \lambda)}{\partial \lambda} = x + z. \end{aligned}$$

The first two conditions give:

$$2x = 2z = -\lambda$$

or,

$$x = z.$$

Substituting $x = z$ into the third condition gives:

$$2z = 0,$$

or,

$$z = 0.$$

The final solution is $(x, y, \lambda) = (0, 0, 0)$. Notice that the Lagrange multiplier, λ , is equal to zero in this case.

11.8 Problems

Exercise 11.1. Annual estimates of the CER model parameters for Boeing and Microsoft are given in the table below:

	Boeing	Microsoft
μ	0.149	0.331
σ^2	0.069	0.136
ρ	-0.008	-0.008

1. Create the following portfolios.
 - Combinations of Boeing and Microsoft (with $x_{\text{boeing}} = -1, -0.9, \dots, 2$ and $x_{\text{msft}} = 1 - x_{\text{boeing}}$)
 - Combinations of Boeing and T-bills (with $x_{\text{boeing}} = 0, 0.1, \dots, 2$)
 - Combinations of Microsoft and T-bills (with $x_{\text{msft}} = 0, 0.1, \dots, 2$)
 - Use an annual risk-free rate of 3% per year for the T-bill. For each portfolio, compute $E(R_p)$, $\text{var}(R_p)$ and $\text{SD}(R_p)$ using the appropriate formulas. For each portfolio, plot $E(R_p)$ vs. $\text{SD}(R_p)$ and put these values on the same graph. Compute the Sharpe's slope for Boeing and Microsoft. Which asset has the highest slope value?
2. Compute the global minimum variance portfolio using the analytical formula (11.10).
 - Make a bar chart showing the weights of Boeing and Microsoft in global minimum variance portfolio.
 - Compute $E(R_p)$, $\text{var}(R_p)$ and $\text{SD}(R_p)$ for the tangency portfolio.
 - Compute Sharpe's slope for the tangency portfolio.
 - Indicate the location of the tangency portfolio on the graph you created previously in question 1.
3. Using a risk-free rate of 3% per year for the T-bill, compute the tangency portfolio using the analytic formula (11.14).

- Make a bar chart showing the weights of Boeing and Microsoft in the tangency portfolio.
 - Compute $E(R_p)$, $\text{var}(R_p)$ and $\text{SD}(R_p)$ for the tangency portfolio.
 - Compute the Sharpe's slope for the tangency portfolio.
 - Indicate the location of the tangency portfolio on the graph you created previously in question 1.
4. Consider a portfolio that has 10% in the tangency portfolio and 90% in T-bills.
- In this portfolio, what is the percent invested in Boeing and what is the percent invested in Microsoft? Give a bar chart showing the percents invested in T-bills, Boeing and Microsoft.
 - Compute $E(R_p)$, $\text{var}(R_p)$ and $\text{SD}(R_p)$ for this portfolio.
 - Compute Sharpe's slope for this portfolio.
 - Indicate the location of the tangency portfolio on the graph you created previously in question 1.
5. Find the efficient portfolio (combination of T-bills and tangency portfolio) that has the same risk (SD) as Microsoft.
- In this portfolio, what is the percent invested in Boeing and what is the percent invested in Microsoft? Give a bar chart showing the percent invested in T-bills, Boeing and Microsoft.
 - Compute $E(R_p)$, $\text{var}(R_p)$ and $\text{SD}(R_p)$ for this portfolio.
 - Compute Sharpe's slope for this portfolio.
 - Indicate the location of the tangency portfolio on the graph you created previously in question 1.

References

1. Black, F. and Litterman, R. (1992). "Global portfolio optimization", *Financial Analysts Journal*, September-October, 28 - XX.
2. Benninga, S. (2000), *Financial Modeling*, Second Edition. Cambridge, MA: MIT Press.
3. Bodie, Z., Kane, A., and Marcus, A.J. (2013), *Investments*, 10th Edition. McGraw-Hill Education.
4. Elton, E., G. Gruber, Brown, S.J., Goetzmann, W.N. (2014). *Modern Portfolio Theory and Investment Analysis*, 9th Edition. New York: Wiley.
5. Jobson, J.D. and Korkie, B. (1981). "Estimation for Markowitz Efficient Portfolios," *Journal of the American Statistical Association*, 75, 544-554.
6. Jobson, J.D. and Korkie, B. (1981). "Putting Markowitz theory to work," *The Journal of Portfolio Management*, Summer, 70-74.
7. Jobson, J.D. and Korkie, B., and Ratti, B. "Improved estimation of Markowitz Portfolio Using James-Stein Type Estimators," *Proceedings of the American Statistical Association, Business and Economics Statistics Section*, 1979.
8. Jorion, P. (1992). "Portfolio optimization in practice," *Financial Analysts Journal*, January-February, 68-74.

9. Markowitz, H. (1987). *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Cambridge, MA: Basil Blackwell.
10. Markowitz, H. (1991). *Portfolio Selection: Efficient Diversification of Investments*. New York: Wiley, 1959; 2nd ed., Cambridge, MA: Basil Blackwell.
11. Michaud, R.O. (1989). "The Markowitz optimization enigma: Is 'optimized' optimal?," *Financial Analysts Journal*, January-February, 31-42.
12. Michaud, R.O. (1998). *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. Boston, MA:Harvard Business School Press.
13. Sharpe, W. (1999). *Investments*, 6th Edition. Prentice Hall.

Chapter 12

Portfolio Theory with Matrix Algebra

Updated: September 26, 2016

Copyright © Eric Zivot 2015, 2016

When working with large portfolios, the algebra of representing portfolio expected returns and variances becomes extremely cumbersome. As shown in chapter 3, the use of matrix algebra can greatly simplify many of these computations. Matrix algebra formulations are also very useful when it comes time to do actual computations on the computer. The matrix algebra formulas are easy to translate into matrix programming languages like R. Popular spreadsheet programs like Microsoft Excel, which are the workhorse programs of many financial houses, can also handle basic matrix calculations. All of this makes it worthwhile to become familiar with matrix techniques for portfolio calculations.

In this chapter, we extend the mean-variance portfolio theory introduced in chapter 11 to handle more than two risky assets. This extension allows the theory to be applied to the analysis of real world portfolios that consist of many risky assets as well as a risk-free asset. This chapter is laid out as follows. Section

The examples in this chapter use the **IntroCompFinR** and **PerformanceAnalytics** packages. Make sure these packages are installed and loaded in R before replicating the chapter examples.

12.1 Portfolios with N Risky Assets

Consider a portfolio with N risky assets, where N can be a large number (e.g., $N = 1,000$). Let R_i ($i = 1, \dots, N$) denote the simple return on asset i (over some fixed time horizon such as one year) and assume that the constant expected return (CER) model holds for all assets:

$$R_i \sim iid N(\mu_i, \sigma_i^2),$$
$$\text{cov}(R_i, R_j) = \sigma_{ij}.$$

Let x_i denote the share of wealth invested in asset i ($i = 1, \dots, N$), and assume that all wealth is invested in the N assets so that $\sum_{i=1}^N x_i = 1$. Assume that short sales are allowed so that some values of x_i can be negative. The portfolio return, $R_{p,x}$, is the random variable $R_{p,x} = \sum_{i=1}^N x_i R_i$. The subscript "x" indicates that the portfolio is constructed using the "x-weights" x_1, x_2, \dots, x_N . The expected return on the portfolio is:

$$\mu_{p,x} = \sum_{i=1}^N x_i \mu_i,$$

and the variance of the portfolio return is:

$$\sigma_{p,x}^2 = \sum_{i=1}^N x_i^2 \sigma_i^2 + 2 \sum_{i=1}^N \sum_{j \neq i} x_i x_j \sigma_{ij}.$$

Notice that variance of the portfolio return depends on N variance terms and $N(N - 1)$ covariance terms. Hence, with N assets there are many more covariance terms than variance terms contributing to portfolio variance. For example, with $N = 100$ there are 100 variance terms and $100 \times 99 = 9900$ covariance terms. With N assets, the algebra representing the portfolio return and risk characteristics is cumbersome, especially for the variance. We can greatly simplify the portfolio algebra using matrix notation, and this was previewed in Chapter 3.

12.1.1 Portfolio return and risk characteristics using matrix notation

Define the following $N \times 1$ column vectors containing the asset returns and portfolio weights:

$$\mathbf{R} = \begin{pmatrix} R_1 \\ \vdots \\ R_N \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}.$$

In matrix notation we can lump multiple returns in a single vector which we denote by \mathbf{R} . Since each of the elements in \mathbf{R} is a random variable we call \mathbf{R} a *random vector*. The probability distribution of the random vector \mathbf{R} is simply the joint distribution of the elements of \mathbf{R} . In the CER model all returns are jointly normally distributed and this joint distribution is completely characterized by the means, variances and covariances of the returns. We can easily express these values using matrix notation as follows. The $N \times 1$ vector of portfolio expected values is:

$$E[\mathbf{R}] = E \left[\begin{pmatrix} R_1 \\ \vdots \\ R_N \end{pmatrix} \right] = \begin{pmatrix} E[R_1] \\ \vdots \\ E[R_N] \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} = \boldsymbol{\mu},$$

and the $N \times N$ covariance matrix of returns is,

$$\text{var}(\mathbf{R}) = \begin{pmatrix} \text{var}(R_1) & \text{cov}(R_1, R_2) & \cdots & \text{cov}(R_1, R_N) \\ \text{cov}(R_1, R_2) & \text{var}(R_2) & \cdots & \text{cov}(R_2, R_N) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(R_1, R_N) & \text{cov}(R_2, R_N) & \cdots & \text{var}(R_N) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix} = \boldsymbol{\Sigma}.$$

Notice that the covariance matrix is symmetric (elements off the diagonal are equal so that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}'$, where $\boldsymbol{\Sigma}'$ denotes the transpose of $\boldsymbol{\Sigma}$) since $\text{cov}(R_i, R_j) = \text{cov}(R_j, R_i)$ for $i \neq j$. It will be positive definite provided no pair of assets is perfectly correlated ($|\rho_{ij}| \neq 1$ for all $i \neq j$) and no asset has a constant return ($\sigma_i^2 > 0$ for all i).

The return on the portfolio using matrix notation is:

$$R_{p,x} = \mathbf{x}'\mathbf{R} = (x_1, \dots, x_N) \cdot \begin{pmatrix} R_1 \\ \vdots \\ R_N \end{pmatrix} = x_1 R_1 + \cdots + x_N R_N.$$

Similarly, the expected return on the portfolio is:

$$\mu_{p,x} = E[\mathbf{x}'\mathbf{R}] = \mathbf{x}'E[\mathbf{R}] = \mathbf{x}'\boldsymbol{\mu} = (x_1, \dots, x_N) \cdot \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} = x_1 \mu_1 + \cdots + x_N \mu_N.$$

The variance of the portfolio is:

$$\begin{aligned} \sigma_{p,x}^2 &= \text{var}(\mathbf{x}'\mathbf{R}) = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} = (x_1, \dots, x_N) \cdot \begin{pmatrix} \sigma_1^2 & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{1N} & \cdots & \sigma_N^2 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} \\ &= \sum_{i=1}^N x_i^2 \sigma_i^2 + 2 \sum_{i=1}^N \sum_{j \neq i} x_i x_j \sigma_{ij}. \end{aligned}$$

The condition that the portfolio weights sum to one can be expressed as:

$$\mathbf{x}'\mathbf{1} = (x_1, \dots, x_N) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = x_1 + \cdots + x_N = 1,$$

where $\mathbf{1}$ is a $N \times 1$ vector with each element equal to 1.

Consider another portfolio with weights $\mathbf{y} = (y_1, \dots, y_N)' \neq \mathbf{x}$. The return on this portfolio is $R_{p,y} = \mathbf{y}'\mathbf{R}$. Later on we will need to compute the covariance between the return on portfolio \mathbf{x} and the return on portfolio \mathbf{y} , $\text{cov}(R_{p,x}, R_{p,y})$. Using matrix algebra, this covariance can be computed as:

$$\sigma_{xy} = \text{cov}(R_{p,x}, R_{p,y}) = \text{cov}(\mathbf{x}'\mathbf{R}, \mathbf{y}'\mathbf{R}) = \mathbf{x}'\Sigma\mathbf{y} = (x_1, \dots, x_N) \cdot \begin{pmatrix} \sigma_1^2 & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{1N} & \cdots & \sigma_N^2 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

Example 12.1. Three asset example data.

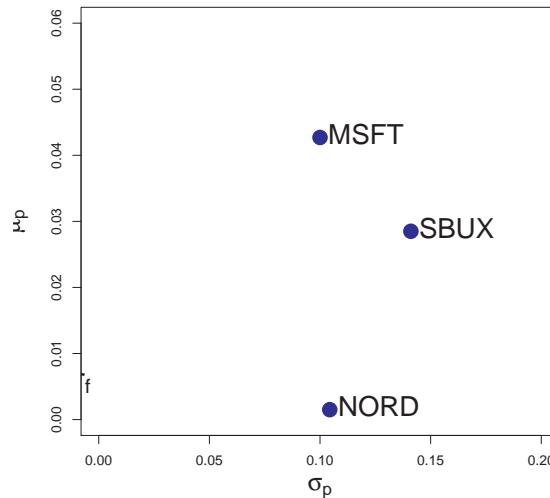


Fig. 12.1 Risk-return characteristics of example data.

To illustrate portfolio calculations in R, table 12.1 gives example values on monthly means, variances and covariances for the simple returns on Microsoft, Nordstrom and Starbucks stock based on sample statistics computed over the five-year period January, 1995 through January, 2000. The risk-free asset is the monthly T-Bill with rate $r_f = 0.005$.

The example data in matrix notation is

Stock i	μ_i	σ_i	Pair (i,j)	σ_{ij}
MSFT	0.0427	0.1000	(MSFT,NORD)	0.0018
NORD	0.0015	0.1044	(MSFT,SBUX)	0.0011
SBUX	0.0285	0.1411	(NORD,SBUX)	0.0026
T-Bill	0.005	0	(T-Bill,MSFT), (T-Bill,NORD), (T-Bill,SBUX)	0

Table 12.1 Three asset example data.

$$\mu = \begin{pmatrix} \mu_{MSFT} \\ \mu_{NORD} \\ \mu_{SBUX} \end{pmatrix} = \begin{pmatrix} 0.0427 \\ 0.0015 \\ 0.0285 \end{pmatrix},$$

$$\Sigma = \begin{pmatrix} \text{var}(R_{MSFT}) & \text{cov}(R_{MSFT}, NORD) & \text{cov}(R_{MSFT}, SBUX) \\ \text{cov}(R_{MSFT}, NORD) & \text{var}(R_{NORD}) & \text{cov}(R_{NORD}, SBUX) \\ \text{cov}(R_{MSFT}, SBUX) & \text{cov}(R_{NORD}, SBUX) & \text{var}(R_{SBUX}) \end{pmatrix} = \begin{pmatrix} 0.0100 & 0.0018 & 0.0011 \\ 0.0018 & 0.0109 & 0.0026 \\ 0.0011 & 0.0026 & 0.0199 \end{pmatrix}.$$

In R, the example data is created using:

```
asset.names <- c("MSFT", "NORD", "SBUX")
mu.vec = c(0.0427, 0.0015, 0.0285)
names(mu.vec) = asset.names
sigma.mat = matrix(c(0.01, 0.0018, 0.0011, 0.0018, 0.0109, 0.0026, 0.0011, 0.0026, 0.0199), nrow = 3, ncol = 3)
dimnames(sigma.mat) = list(asset.names, asset.names)
r.f = 0.005
```

The values of μ_i and σ_i ($i = \text{MSFT}, \text{NORD}, \text{SBUX}$) are shown in Figure 12.1, created with:

```
sd.vec = sqrt(diag(sigma.mat))
cex.val = 2
plot(sd.vec, mu.vec, ylim = c(0, 0.06), xlim = c(0, 0.2), ylab = expression(mu[p]),
      xlab = expression(sigma[p]), pch = 16, col = "blue", cex = 2.5, cex.lab = 1.75)
text(sd.vec, mu.vec, labels = asset.names, pos = 4, cex = cex.val)
text(0, r.f, labels = expression(r[f]), pos = 2, cex = cex.val)
```

Clearly, Microsoft provides the best risk-return trade-off (i.e., has the highest Sharpe ratio) and Nordstrom provides with worst.

Example 12.2. Portfolio computations in R.

Consider an equally weighted portfolio with $x_{MSFT} = x_{NORD} = x_{SBUX} = 1/3$. This portfolio has return $R_{p,x} = \mathbf{x}'\mathbf{R}$ where $\mathbf{x} = (1/3, 1/3, 1/3)'$. Using R, the portfolio mean and variance are:

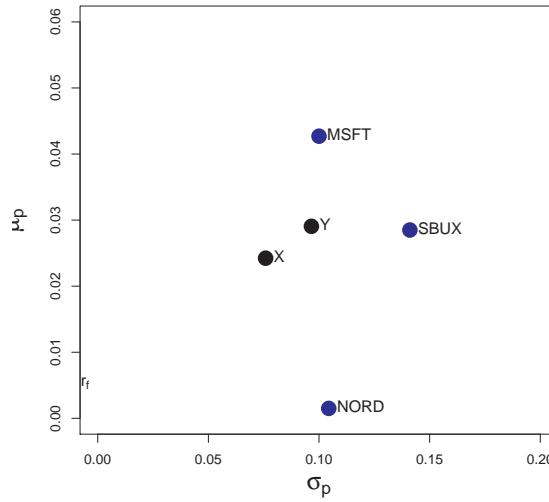


Fig. 12.2 Risk-return characteristics of example data and portfolios with weight vectors $\mathbf{x} = (0.333, 0.333, 0.333)'$ and $\mathbf{y} = (0.8, 0.4, -0.2)'$.

```

x.vec = rep(1, 3)/3
names(x.vec) = asset.names
sum(x.vec)

## [1] 1

mu.p.x = crossprod(x.vec, mu.vec)
sig2.p.x = t(x.vec) %*% sigma.mat %*% x.vec
sig.p.x = sqrt(sig2.p.x)
mu.p.x

##          [,1]
## [1,] 0.0242

sig.p.x

##          [,1]
## [1,] 0.0759

```

Next, consider another portfolio with weight vector $\mathbf{y} = (y_{MSFT}, y_{NORD}, y_{SBUX})' = (0.8, 0.4, -0.2)'$ and return $R_{p,y} = \mathbf{y}'\mathbf{R}$. The mean and volatility of this portfolio are

```

y.vec = c(0.8, 0.4, -0.2)
names(y.vec) = asset.names
mu.p.y = crossprod(y.vec, mu.vec)
sig2.p.y = t(y.vec) %*% sigma.mat %*% y.vec
sig.p.y = sqrt(sig2.p.y)
mu.p.y

##          [,1]

```

```
## [1,] 0.0291
sig.p.y
## [,1]
## [1,] 0.0966
```

The covariance and correlation between $R_{p,x}$ and $R_{p,y}$ are:

```
sig.xy = t(x.vec) %*% sigma.mat %*% y.vec
rho.xy = sig.xy/(sig.p.x * sig.p.y)
sig.xy
## [,1]
## [1,] 0.00391
rho.xy
## [,1]
## [1,] 0.533
```

This return and risk characteristics of the three assets together with portfolio **x** and portfolio **y** are illustrated in Figure 12.2. Notice that the equally weighted portfolio, portfolio **x**, has the smallest volatility.

■

Example 12.3. Risk-return characteristics of random portfolios.

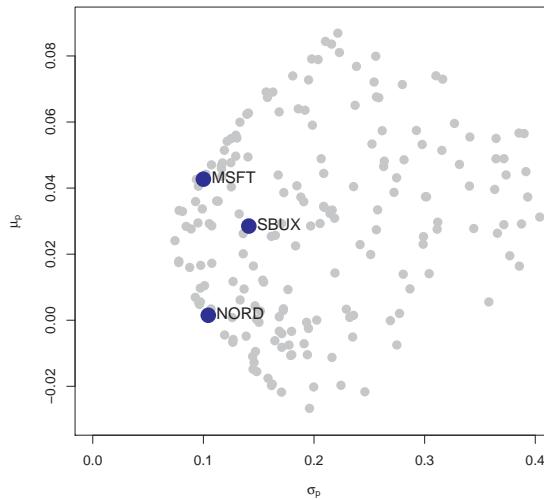


Fig. 12.3 Risk-return characteristics of 200 random portfolios of three assets.

Figure 12.2 shows that the risk-return points associated with portfolios of three assets do not fall nicely on a bullet-shaped line, as they do with two asset portfolios. In fact, the set of risk-return points generated by varying the three portfolio weights is a solid area. To see this, consider creating a set of 200 randomly generated three asset portfolios whose weights sum to one:

```
set.seed(123)
x.msft = runif(200, min = -1.5, max = 1.5)
x.nord = runif(200, min = -1.5, max = 1.5)
x.sbxu = 1 - x.msft - x.nord
head(cbind(x.msft, x.nord, x.sbxu))

##      x.msft  x.nord  x.sbxu
## [1,] -0.637 -0.7838  2.4211
## [2,]  0.865   1.3871 -1.2520
## [3,] -0.273   0.3041  0.9690
## [4,]  1.149   0.0451 -0.1941
## [5,]  1.321  -0.2923 -0.0291
## [6,] -1.363   1.1407  1.2226
```

Figure 12.3 shows the means and volatilities, as grey dots, of these 200 random portfolios of the three assets. Notice that the grey dots outline a bullet-shaped area (Markowitz bullet) that has an outer boundary that looks like the line characterizing the two-asset portfolio frontier. If we let the number of random portfolios get very large then the grey dots will fill in a solid area. At the tip of the Markowitz bullet is the minimum variance portfolio.

Recall the definition of a mean-variance efficient portfolio given in Chapter 11. For a fixed volatility $\sigma_{p,0}$ the efficient portfolio is the one with the highest expected return. From Figure 12.3, we can see that efficient portfolios will lie on the outer boundary (above the minimum variance portfolio at the tip of the Markowitz bullet) of the set of feasible portfolios.

■

12.1.2 Large portfolios and diversification

One of the key benefits of forming large portfolios is risk reduction due to portfolio diversification. Generally, a diversified portfolio is one in which wealth is spread across many assets.

To illustrate the impact of diversification on the risk of large portfolios, consider a equally weighted portfolio of N risky assets where N is a large number (e.g., $N > 100$). Here, $x_i = 1/N$ and the $N \times 1$ portfolio weight vector is $\mathbf{x} = (1/N, 1/N, \dots, 1/N)' = (1/N)(1, 1, \dots, 1)' = (1/N)\mathbf{1}$ where $\mathbf{1}$ is an $N \times 1$ vector of ones. Let the $N \times 1$ return vector \mathbf{R} be described the CER model so that $\mathbf{R} \sim N(\mu, \Sigma)$. The portfolio return is $R_{p,x} = \mathbf{x}'\mathbf{R} = (1/N)\mathbf{1}'\mathbf{R} = (1/N) \sum_{i=1}^N R_i$. The variance of this portfolio is

$$\text{var}(R_{p,x}) = \text{var}(\mathbf{x}'\mathbf{R}) = \text{var}\left(\frac{1}{N}\mathbf{1}'\mathbf{R}\right) = \frac{1}{N^2}\mathbf{1}'\Sigma\mathbf{1}.$$

Now,

$$\mathbf{1}'\boldsymbol{\Sigma}\mathbf{1} = (\mathbf{1} \mathbf{1} \dots \mathbf{1})' \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \dots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \dots & \sigma_N^2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \text{sum of all elements in } \boldsymbol{\Sigma}.$$

As a result, we can write

$$\mathbf{1}'\boldsymbol{\Sigma}\mathbf{1} = \sum_{i=1}^N \sigma_i^2 + \sum_{j \neq i} \sigma_{ij}$$

which is the sum of the N diagonal variance terms plus the $N(N-1)$ off-diagonal covariance terms. Define the average variance as the average of the diagonal terms of $\boldsymbol{\Sigma}$

$$\bar{\text{var}} = \frac{1}{N} \sum_{i=1}^N \sigma_i^2,$$

and the average covariance as the average of the off-diagonal terms of $\boldsymbol{\Sigma}$

$$\bar{\text{cov}} = \frac{1}{N(N-1)} \sum_{j \neq i} \sigma_{ij}.$$

Then $\mathbf{1}'\boldsymbol{\Sigma}\mathbf{1} = N\bar{\text{var}} + N(N-1)\bar{\text{cov}}$ and

$$\begin{aligned} \text{var}(R_{p,x}) &= \frac{1}{N^2} \mathbf{1}'\boldsymbol{\Sigma}\mathbf{1} = \frac{1}{N} \bar{\text{var}} + \frac{N(N-1)}{N^2} \bar{\text{cov}} \\ &= \frac{1}{N} \bar{\text{var}} + \left(1 - \frac{1}{N}\right) \bar{\text{cov}}. \end{aligned} \tag{12.1}$$

If N is reasonably large (e.g. $N > 100$) then

$$\frac{1}{N} \bar{\text{var}} \approx 0 \text{ and } \frac{1}{N} \bar{\text{cov}} \approx 0$$

so that

$$\text{var}(R_{p,x}) \approx \bar{\text{cov}}.$$

Hence, in a large diversified portfolio the portfolio variance is approximately equal to the average of the pairwise covariances. So what matters for portfolio risk is not the individual asset variances but rather the average of the asset covariances. In particular, portfolios with highly positively correlated returns will have higher portfolio variance than portfolios with less correlated returns.

Equation (12.1) also shows that portfolio variance should be a decreasing function of the number of assets in the portfolio and that this function should level off at $\bar{\text{cov}}$ for large N .

Large portfolio with perfectly positively correlated assets

In chapter 11 it was shown that the volatility of a portfolio of two risky assets with perfectly positively correlated returns ($\rho_{12} = 1$) is equal to a share weighted average of the asset volatilities: $\sigma_p = x_1\sigma_1 + x_2\sigma_2$. Here, we show using matrix algebra that this result generalizes to case of N assets. Let \mathbf{R} be an $N \times 1$ vector of asset returns and assume that all returns are perfectly positively correlated. The $N \times N$ return correlation matrix is then

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} = \mathbf{1}\mathbf{1}',$$

and the $N \times N$ return covariance matrix is

$$\Sigma = \mathbf{D}\mathbf{C}\mathbf{D} = \mathbf{D}\mathbf{1}\mathbf{1}'\mathbf{D},$$

where $\mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_N)$. Let $\mathbf{x} = (x_1, x_2, \dots, x_N)'$ be a portfolio weight vector and let $R_{p,x} = \mathbf{x}'\mathbf{R}$ denote the portfolio return. Then

$$\sigma_p^2 = \text{var}(\mathbf{x}'\mathbf{R}) = \mathbf{x}'\Sigma\mathbf{x} = \mathbf{x}'\mathbf{D}\mathbf{1}\mathbf{1}'\mathbf{D}\mathbf{x} = (\mathbf{x}'\mathbf{D}\mathbf{1})^2.$$

Now

$$\begin{aligned} \mathbf{x}'\mathbf{D}\mathbf{1} &= \left(x_1 \ x_2 \ \dots \ x_N \right) \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} x_1\sigma_1 & 0 & \dots & 0 \\ 0 & x_2\sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x_N\sigma_N \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \\ &= \sum_{i=1}^N x_i\sigma_i. \end{aligned}$$

Therefore,

$$\sigma_p^2 = (\mathbf{x}'\mathbf{D}\mathbf{1})^2 = \left(\sum_{i=1}^N x_i\sigma_i \right)^2$$

and so

$$\sigma_p = \sum_{i=1}^N x_i\sigma_i.$$

When all assets have perfectly positively correlated returns, portfolio volatility is a share weighted average of individual asset volatilities. You can think of this as the risk of a (long-only) portfolio when there are no diversification benefits. When assets are not all perfectly positively correlated, there are some diversification benefits and so σ_p will be less than $\sum_{i=1}^N x_i\sigma_i$.

12.2 Determining the Global Minimum Variance Portfolio Using Matrix Algebra

The global minimum variance portfolio $\mathbf{m} = (m_1, \dots, m_N)'$ for the N asset case solves the constrained minimization problem:

$$\min_{\mathbf{m}} \sigma_{p,m}^2 = \mathbf{m}' \Sigma \mathbf{m} \text{ s.t. } \mathbf{m}' \mathbf{1} = 1. \quad (12.2)$$

The Lagrangian for this problem is:

$$L(\mathbf{m}, \lambda) = \mathbf{m}' \Sigma \mathbf{m} + \lambda(\mathbf{m}' \mathbf{1} - 1)$$

,and the first order conditions (FOCs) for a minimum are:

$$\mathbf{0} = \frac{\partial L}{\partial \mathbf{m}} = \frac{\partial \mathbf{m}' \Sigma \mathbf{m}}{\partial \mathbf{m}} + \frac{\partial \lambda(\mathbf{m}' \mathbf{1} - 1)}{\partial \mathbf{m}} = 2\Sigma \mathbf{m} + \lambda \mathbf{1}, \quad (12.3)$$

$$0 = \frac{\partial L}{\partial \lambda} = \frac{\partial \mathbf{m}' \Sigma \mathbf{m}}{\partial \lambda} + \frac{\partial \lambda(\mathbf{m}' \mathbf{1} - 1)}{\partial \lambda} = \mathbf{m}' \mathbf{1} - 1. \quad (12.4)$$

The FOCs (12.3)-(12.4) gives $N+1$ linear equations in $N+1$ unknowns which can be solved to find the global minimum variance portfolio weight vector \mathbf{m} as follows. The $N+1$ linear equations describing the first order conditions has the matrix representation:

$$\begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix} \begin{pmatrix} \mathbf{m} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}. \quad (12.5)$$

The system (12.5) is of the form $\mathbf{A}_m \mathbf{z}_m = \mathbf{b}$, where

$$\mathbf{A}_m = \begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}' & 0 \end{pmatrix}, \quad \mathbf{z}_m = \begin{pmatrix} \mathbf{m} \\ \lambda \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}.$$

Provided \mathbf{A}_m is invertible, the solution for \mathbf{z}_m is:¹

$$\mathbf{z}_m = \mathbf{A}_m^{-1} \mathbf{b}. \quad (12.6)$$

The first N elements of \mathbf{z}_m is the portfolio weight vector \mathbf{m} for the global minimum variance portfolio with expected return $\mu_{p,m} = \mathbf{m}' \mu$ and variance $\sigma_{p,m}^2 = \mathbf{m}' \Sigma \mathbf{m}$.

Example 12.4. Global minimum variance portfolio for example data.

Using the data in Table 12.1, we can use R to compute the global minimum variance portfolio weights from (12.6) as follows:

```
top.mat = cbind(2 * sigma.mat, rep(1, 3))
bot.vec = c(rep(1, 3), 0)
```

¹ For \mathbf{A}_m to be invertible we need Σ to be invertible, which requires $|\rho_{ij}| \neq 1$ for $i \neq j$ and $\sigma_i^2 > 0$ for all i .

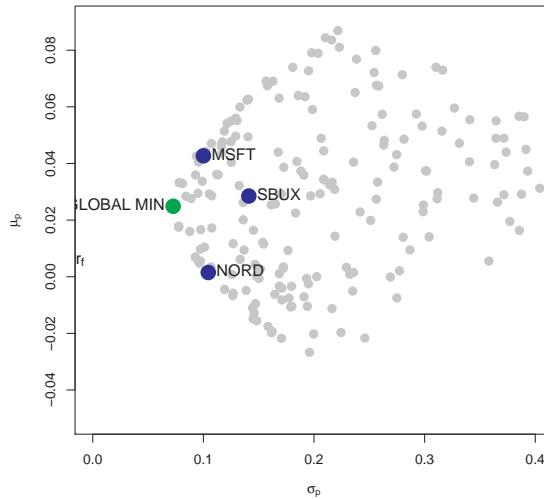


Fig. 12.4 Global minimum variance portfolio from example data.

```

Am.mat = rbind(top.mat, bot.vec)
b.vec = c(rep(0, 3), 1)
z.m.mat = solve(Am.mat) %*% b.vec
m.vec = z.m.mat[1:3, 1]
m.vec

## MSFT  NORD  SBUX
## 0.441 0.366 0.193

```

Hence, the global minimum variance portfolio has portfolio weights $m_{\text{msft}} = 0.441$, $m_{\text{nord}} = 0.366$ and $m_{\text{sbux}} = 0.193$, and is given by the vector $\mathbf{m} = (0.441, 0.366, 0.193)'$. The expected return on this portfolio, $\mu_{p,m} = \mathbf{m}'\boldsymbol{\mu}$, is:

```

mu.gmin = as.numeric(crossprod(m.vec, mu.vec))
mu.gmin

## [1] 0.0249

```

The portfolio variance, $\sigma_{p,m}^2 = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{m}$, and standard deviation, $\sigma_{p,m}$, are:

```

sig2.gmin = as.numeric(t(m.vec) %*% sigma.mat %*% m.vec)
sig.gmin = sqrt(sig2.gmin)
sig2.gmin

## [1] 0.00528

sig.gmin

## [1] 0.0727

```

In Figure 12.4, this portfolio is labeled “GLOBAL MIN”. ■

12.2.1 Alternative derivation of global minimum variance portfolio

We can use the first order conditions (12.3) - (12.4) to give an explicit solution for the global minimum variance portfolio \mathbf{m} as follows. First, use (12.3) to solve for \mathbf{m} as a function of λ :

$$\mathbf{m} = -\frac{1}{2} \cdot \lambda \boldsymbol{\Sigma}^{-1} \mathbf{1}. \quad (12.7)$$

Next, multiply both sides of (12.7) by $\mathbf{1}'$ and use (12.4) to solve for λ :

$$\begin{aligned} \mathbf{1} &= \mathbf{1}' \mathbf{m} = -\frac{1}{2} \cdot \lambda \mathbf{1}' \boldsymbol{\Sigma}^{-1} \mathbf{1} \\ \Rightarrow \lambda &= -2 \cdot \frac{1}{\mathbf{1}' \boldsymbol{\Sigma}^{-1} \mathbf{1}}. \end{aligned}$$

Finally, substitute the value for λ back into (12.7) to solve for \mathbf{m} :

$$\mathbf{m} = -\frac{1}{2} (-2) \frac{1}{\mathbf{1}' \boldsymbol{\Sigma}^{-1} \mathbf{1}} \boldsymbol{\Sigma}^{-1} \mathbf{1} = \frac{\boldsymbol{\Sigma}^{-1} \mathbf{1}}{\mathbf{1}' \boldsymbol{\Sigma}^{-1} \mathbf{1}}. \quad (12.8)$$

Notice that (12.8) shows that a solution for \mathbf{m} exists as long as $\boldsymbol{\Sigma}$ is invertible.

Example 12.5. Finding global minimum variance portfolio for example data.

Using the data in Table 12.1, we can use R to compute the global minimum variance portfolio weights from (12.8) as follows:

```
one.vec = rep(1, 3)
sigma.inv.mat = solve(sigma.mat)
top.mat = sigma.inv.mat %*% one.vec
bot.val = as.numeric((t(one.vec) %*% sigma.inv.mat %*% one.vec))
m.mat = top.mat/bot.val
m.mat[, 1]

## MSFT NORD SBUX
## 0.441 0.366 0.193
```

12.3 Determining Mean-Variance Efficient Portfolios Using Matrix Algebra

The investment opportunity set is the set of portfolio expected return, μ_p , and portfolio standard deviation, σ_p , values for all possible portfolios whose weights sum to one. As in the

two risky asset case, this set can be described in a graph with μ_p on the vertical axis and σ_p on the horizontal axis. With two assets, the investment opportunity set in (μ_p, σ_p) -space lies on a curve (one side of a hyperbola). With three or more assets, the investment opportunity set in (μ_p, σ_p) -space is described by set of values whose general shape is complicated and depends crucially on the covariance terms σ_{ij} .² However, we do not have to fully characterize the entire investment opportunity set. If we assume that investors choose portfolios to maximize expected return subject to a target level of risk, or, equivalently, to minimize risk subject to a target expected return, then we can simplify the asset allocation problem by only concentrating on the set of efficient portfolios. These portfolios lie on the boundary of the investment opportunity set above the global minimum variance portfolio. This is the framework originally developed by Harry Markowitz, the father of portfolio theory and winner of the Nobel Prize in economics.

Following Markowitz, we assume that investors wish to find portfolios that have the best expected return-risk trade-off. We can characterize these efficient portfolios in two equivalent ways. In the first way, investors seek to find portfolios that maximize portfolio expected return for a given level of risk as measured by portfolio variance. Let $\sigma_{p,0}^2$ denote a target level of risk. Then the constrained maximization problem to find an efficient portfolio is:

$$\begin{aligned} \max_{\mathbf{x}} \mu_p &= \mathbf{x}'\boldsymbol{\mu} \text{ s.t.} \\ \sigma_p^2 &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} = \sigma_{p,0}^2 \text{ and } \mathbf{x}'\mathbf{1} = 1. \end{aligned} \tag{12.9}$$

The investor's problem of maximizing portfolio expected return subject to a target level of risk has an equivalent dual representation in which the investor minimizes the risk of the portfolio (as measured by portfolio variance) subject to a target expected return level. Let $\mu_{p,0}$ denote a target expected return level. Then the dual problem is the constrained minimization problem:³

$$\begin{aligned} \min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} \text{ s.t.} \\ \mu_p &= \mathbf{x}'\boldsymbol{\mu} = \mu_{p,0}, \text{ and } \mathbf{x}'\mathbf{1} = 1. \end{aligned} \tag{12.10}$$

To find efficient portfolios of risky assets in practice, the dual problem (12.10) is most often solved. This is partially due to computational conveniences and partly due to investors being more willing to specify target expected returns rather than target risk levels. The efficient portfolio frontier is a graph of μ_p versus σ_p values for the set of efficient portfolios generated by solving (12.10) for all possible target expected return levels $\mu_{p,0}$ above the expected return on the global minimum variance portfolio. Just as in the two asset case, the resulting efficient frontier will resemble one side of an hyperbola and is often called the "Markowitz bullet". This frontier is illustrated in Figure 12.3 as the boundary of the set generated by random portfolios above the global minimum variance portfolio.

To solve the constrained minimization problem (12.10), first form the Lagrangian function:

² This set is partially visualized in Figure 12.3 using random portfolios.

³ This duality is similar to the duality in economics of profit maximization and cost minimization.

$$L(x, \lambda_1, \lambda_2) = \mathbf{x}' \Sigma \mathbf{x} + \lambda_1 (\mathbf{x}' \mu - \mu_{p,0}) + \lambda_2 (\mathbf{x}' \mathbf{1} - 1).$$

Because there are two constraints ($\mathbf{x}' \mu = \mu_{p,0}$ and $\mathbf{x}' \mathbf{1} = 1$) there are two Lagrange multipliers λ_1 and λ_2 . The FOCs for a minimum are the linear equations:

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \mathbf{x}} = 2\Sigma \mathbf{x} + \lambda_1 \mu + \lambda_2 \mathbf{1} = \mathbf{0}, \quad (12.11)$$

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_1} = \mathbf{x}' \mu - \mu_{p,0} = 0, \quad (12.12)$$

$$\frac{\partial L(\mathbf{x}, \lambda_1, \lambda_2)}{\partial \lambda_2} = \mathbf{x}' \mathbf{1} - 1 = 0. \quad (12.13)$$

These FOCs consist of $N + 2$ linear equations in $N + 2$ unknowns $(\mathbf{x}, \lambda_1, \lambda_2)$. We can represent the system of linear equations using matrix algebra as:

$$\begin{pmatrix} 2\Sigma \mu \mathbf{1} \\ \mu' \mathbf{0} \mathbf{0} \\ \mathbf{1}' \mathbf{0} \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mu_{p,0} \\ 1 \end{pmatrix},$$

which is of the form $\mathbf{A}\mathbf{z}_x = \mathbf{b}_0$, where

$$\mathbf{A} = \begin{pmatrix} 2\Sigma \mu \mathbf{1} \\ \mu' \mathbf{0} \mathbf{0} \\ \mathbf{1}' \mathbf{0} \mathbf{0} \end{pmatrix}, \quad \mathbf{z}_x = \begin{pmatrix} \mathbf{x} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \text{ and } \mathbf{b}_0 = \begin{pmatrix} \mathbf{0} \\ \mu_{p,0} \\ 1 \end{pmatrix}.$$

The solution for \mathbf{z}_x is then:

$$\mathbf{z}_x = \mathbf{A}^{-1} \mathbf{b}_0. \quad (12.14)$$

The first N elements of \mathbf{z}_x are the portfolio weights \mathbf{x} for the minimum variance portfolio with expected return $\mu_{p,x} = \mu_{p,0}$. If $\mu_{p,0}$ is greater than or equal to the expected return on the global minimum variance portfolio, then \mathbf{x} is an efficient (frontier) portfolio. Otherwise, it is an inefficient (frontier) portfolio.

Example 12.6. Efficient portfolio with the same expected return as Microsoft.

Using the data in Table 12.1, consider finding a minimum variance portfolio with the same expected return as Microsoft. This will be an efficient portfolio because $\mu_{msft} = 0.043 > \mu_{p,m} = 0.025$. Call this portfolio $\mathbf{x} = (x_{msft}, x_{nord}, x_{sbux})'$. That is, consider solving (12.10) with target expected return $\mu_{p,0} = \mu_{msft} = 0.043$ using (12.14). The R calculations to create the matrix \mathbf{A}_x and the vectors \mathbf{z}_x and \mathbf{b}_{msft} are:

```
top.mat = cbind(2 * sigma.mat, mu.vec, rep(1, 3))
mid.vec = c(mu.vec, 0, 0)
bot.vec = c(rep(1, 3), 0, 0)
Ax.mat = rbind(top.mat, mid.vec, bot.vec)
bmsft.vec = c(rep(0, 3), mu.vec["MSFT"], 1)
```

and the R code to solve for \mathbf{x} using (12.14) is:

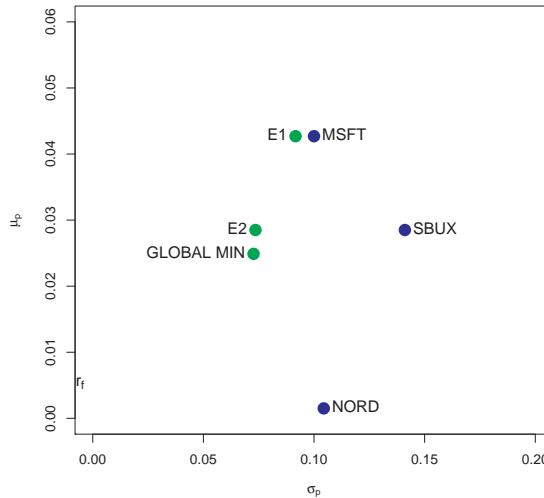


Fig. 12.5 Minimum variance efficient portfolios from example data. Portfolio “E1” has the same expected return as Microsoft, and portfolio “E2” has the same expected returns as Starbucks.

```

z.mat = solve(Ax.mat) %*% bmsft.vec
x.vec = z.mat[1:3, ]
x.vec

##      MSFT      NORD      SBUX
## 0.8275 -0.0907  0.2633

```

The efficient portfolio with the same expected return as Microsoft has portfolio weights $x_{\text{msft}} = 0.8275$, $x_{\text{nord}} = -0.0907$ and $x_{\text{sbux}} = 0.2633$, and is given by the vector $\mathbf{x} = (0.8275, -0.0907, 0.2633)'$. The expected return on this portfolio, $\mu_{p,x} = \mathbf{x}'\boldsymbol{\mu}$, is equal to the target return μ_{msft} :

```

mu.px = as.numeric(crossprod(x.vec, mu.vec))
mu.px

## [1] 0.0427

```

The portfolio variance, $\sigma_{p,x}^2 = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}$, and standard deviation, $\sigma_{p,x}$, are:

```

sig2.px = as.numeric(t(x.vec) %*% sigma.mat %*% x.vec)
sig.px = sqrt(sig2.px)
sig2.px

## [1] 0.0084

sig.px

## [1] 0.0917

```

and are smaller than the corresponding values for Microsoft (see Table 12.1). This efficient portfolio is labeled “E1” in Figure 12.5.

■

Example 12.7. Efficient portfolio with the same expected return as Starbucks.

To find a minimum variance portfolio $\mathbf{y} = (y_{\text{msft}}, y_{\text{nord}}, y_{\text{sbux}})'$ with the same expected return as Starbucks we use (12.14) with $\mathbf{b}_{\text{sbux}} = (\mathbf{0}, \mu_{\text{sbux}}, 1)'$:

```
bsbux.vec = c(rep(0, 3), mu.vec["SBUX"], 1)
z.mat = solve(Ax.mat) %*% bsbux.vec
y.vec = z.mat[1:3, ]
y.vec

## MSFT NORD SBUX
## 0.519 0.273 0.207
```

The portfolio $\mathbf{y} = (0.519, 0.273, 0.207)'$ is an efficient portfolio on the outer boundary because $\mu_{\text{sbux}} = 0.0285 > \mu_{p,m} = 0.0249$. The portfolio expected return and standard deviation are:

```
mu.py = as.numeric(crossprod(y.vec, mu.vec))
sig2.py = as.numeric(t(y.vec) %*% sigma.mat %*% y.vec)
sig.py = sqrt(sig2.py)
mu.py

## [1] 0.0285

sig.py

## [1] 0.0736
```

This efficient portfolio is labeled “E2” in Figure 12.5. It has the same expected return as SBUX but a smaller standard deviation.

The covariance and correlation values between the portfolio returns $R_{p,x} = \mathbf{x}'\mathbf{R}$ and $R_{p,y} = \mathbf{y}'\mathbf{R}$ are given by:

```
sigma.xy = as.numeric(t(x.vec) %*% sigma.mat %*% y.vec)
rho.xy = sigma.xy/(sig.py * sig.py)
sigma.xy

## [1] 0.00591

rho.xy

## [1] 0.877
```

This covariance will be used later on when constructing the entire frontier of efficient portfolios.

■

12.3.1 Alternative derivation of an efficient portfolio

The equation (12.14) does not give an explicit solution for the minimum variance portfolio \mathbf{x} . As with the global minimum variance portfolio (12.8), an explicit solution for \mathbf{x} can also be found. Consider the first order conditions (12.11)-(12.13) from the optimization problem (12.10). First, use (12.11) to solve for the $N \times 1$ vector \mathbf{x} :

$$\mathbf{x} = -\frac{1}{2}\lambda_1\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2}\lambda_2\boldsymbol{\Sigma}^{-1}\mathbf{1}. \quad (12.15)$$

Define the $N \times 2$ matrix $\mathbf{M} = [\boldsymbol{\mu} : \mathbf{1}]$ and the 2×1 vector $\lambda = (\lambda_1, \lambda_2)'$. Then we can rewrite (12.15) in matrix form as:

$$\mathbf{x} = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\mathbf{M}\lambda. \quad (12.16)$$

Next, to find the values for λ_1 and λ_2 , pre-multiply (12.15) by $\boldsymbol{\mu}'$ and use (12.12) to give:

$$\mu_0 = \boldsymbol{\mu}'\mathbf{x} = -\frac{1}{2}\lambda_1\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2}\lambda_2\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1}. \quad (12.17)$$

Similarly, pre-multiply (12.15) by $\mathbf{1}'$ and use (12.13) to give:

$$1 = \mathbf{1}'\mathbf{x} = -\frac{1}{2}\lambda_1\mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - \frac{1}{2}\lambda_2\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}. \quad (12.18)$$

Now, we have two linear equations (12.17) and (12.18) involving λ_1 and λ_2 which we can write in matrix notation as:

$$-\frac{1}{2} \begin{pmatrix} \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} & \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1} \\ \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1} & \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mu_0 \\ 1 \end{pmatrix}. \quad (12.19)$$

Define,

$$\begin{aligned} \begin{pmatrix} \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} & \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1} \\ \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\mathbf{1} & \mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1} \end{pmatrix} &= \mathbf{M}'\boldsymbol{\Sigma}^{-1}\mathbf{M} = \mathbf{B}, \\ \tilde{\boldsymbol{\mu}}_0 &= \begin{pmatrix} \mu_0 \\ 1 \end{pmatrix}, \end{aligned}$$

so that we can rewrite (12.19) as,

$$-\frac{1}{2}\mathbf{B}\lambda = \tilde{\boldsymbol{\mu}}_0.$$

Provided \mathbf{B} is invertible, the solution for $\lambda = (\lambda_1, \lambda_2)'$ is

$$\lambda = -2\mathbf{B}^{-1}\tilde{\boldsymbol{\mu}}_0. \quad (12.20)$$

Substituting (12.20) back into (12.16) gives an explicit expression for the efficient portfolio weight vector \mathbf{x} :

$$\mathbf{x} = -\frac{1}{2}\Sigma^{-1}\mathbf{M}\lambda = -\frac{1}{2}\Sigma^{-1}\mathbf{M}(-2\mathbf{B}^{-1}\tilde{\mu}_0) = \Sigma^{-1}\mathbf{M}\mathbf{B}^{-1}\tilde{\mu}_0. \quad (12.21)$$

Example 12.8. Alternative solution for efficient portfolio with the same expected return as Microsoft.

The R code to compute the efficient portfolio with the same expected return as Microsoft using (12.21) is:

```
M.mat = cbind(mu.vec, one.vec)
B.mat = t(M.mat) %*% solve(sigma.mat) %*% M.mat
mu.tilde.msft = c(mu.vec["MSFT"], 1)
x.vec.2 = solve(sigma.mat) %*% M.mat %*% solve(B.mat) %*% mu.tilde.msft
x.vec.2

##          [,1]
## MSFT   0.8275
## NORD  -0.0907
## SBUX   0.2633
```

■

12.4 Computing the Mean-Variance Efficient Frontier

The analytic expression for a minimum variance portfolio (12.21) can be used to show that any minimum variance portfolio can be created as an affine combination of any two minimum variance portfolios with different target expected returns. If the expected return on the resulting portfolio is greater than the expected on the global minimum variance portfolio, then the portfolio is an efficient frontier portfolio. Otherwise, the portfolio is an inefficient frontier portfolio. As a result, to compute the portfolio frontier in (μ_p, σ_p) space (Markowitz bullet) we only need to find two efficient portfolios. The remaining frontier portfolios can then be expressed as affine combinations of these two portfolios. The following proposition describes the process for the three risky asset case using matrix algebra.

Proposition 12.1. *Creating a frontier portfolio from two minimum variance portfolios.*

Let \mathbf{x} and \mathbf{y} be any two minimum variance portfolios with different target expected returns $\mathbf{x}'\mu = \mu_{p,0} \neq \mathbf{y}'\mu = \mu_{p,1}$. That is, portfolio \mathbf{x} solves:

$$\min_{\mathbf{x}} \sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x} \text{ s.t. } \mathbf{x}'\mu = \mu_{p,0} \text{ and } \mathbf{x}'\mathbf{1} = 1,$$

and portfolio \mathbf{y} solves,

$$\min_{\mathbf{y}} \sigma_{p,y}^2 = \mathbf{y}'\Sigma\mathbf{y} \text{ s.t. } \mathbf{y}'\mu = \mu_{p,1} \text{ and } \mathbf{y}'\mathbf{1} = 1.$$

Let α be any constant and define the portfolio \mathbf{z} as an affine combination of portfolios \mathbf{x} and \mathbf{y} :

$$\begin{aligned}\mathbf{z} &= \alpha \cdot \mathbf{x} + (1 - \alpha) \cdot \mathbf{y} \\ &= \begin{pmatrix} \alpha x_1 + (1 - \alpha)y_1 \\ \vdots \\ \alpha x_N + (1 - \alpha)y_N \end{pmatrix}.\end{aligned}\tag{12.22}$$

Then the following results hold:

1. The portfolio \mathbf{z} is a minimum variance portfolio with expected return and variance given by:

$$\mu_{p,z} = \mathbf{z}'\boldsymbol{\mu} = \alpha \cdot \mu_{p,x} + (1 - \alpha) \cdot \mu_{p,y},\tag{12.23}$$

$$\sigma_{p,z}^2 = \mathbf{z}'\boldsymbol{\Sigma}\mathbf{z} = \alpha^2\sigma_{p,x}^2 + (1 - \alpha)^2\sigma_{p,y}^2 + 2\alpha(1 - \alpha)\sigma_{xy},\tag{12.24}$$

where $\sigma_{p,x}^2 = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}$, $\sigma_{p,y}^2 = \mathbf{y}'\boldsymbol{\Sigma}\mathbf{y}$, $\sigma_{xy} = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{y}$.

2. If $\mu_{p,z} \geq \mu_{p,m}$, where $\mu_{p,m}$ is the expected return on the global minimum variance portfolio, then portfolio \mathbf{z} is an efficient frontier portfolio. Otherwise, \mathbf{z} is an inefficient frontier portfolio.

The proof of 1. follows directly from applying (12.21) to portfolios \mathbf{x} and \mathbf{y} :

$$\begin{aligned}\mathbf{x} &= \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}\tilde{\boldsymbol{\mu}}_x, \\ \mathbf{y} &= \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}\tilde{\boldsymbol{\mu}}_y,\end{aligned}$$

where $\tilde{\boldsymbol{\mu}}_x = (\mu_{p,x}, 1)'$ and $\tilde{\boldsymbol{\mu}}_y = (\mu_{p,y}, 1)'$. Then for portfolio \mathbf{z} :

$$\begin{aligned}\mathbf{z} &= \alpha \cdot \mathbf{x} + (1 - \alpha) \cdot \mathbf{y} \\ &= \alpha \cdot \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}\tilde{\boldsymbol{\mu}}_x + (1 - \alpha) \cdot \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}\tilde{\boldsymbol{\mu}}_y \\ &= \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}(\alpha \cdot \tilde{\boldsymbol{\mu}}_x + (1 - \alpha) \cdot \tilde{\boldsymbol{\mu}}_y) \\ &= \boldsymbol{\Sigma}^{-1}\mathbf{MB}^{-1}\tilde{\boldsymbol{\mu}}_z,\end{aligned}$$

where $\tilde{\boldsymbol{\mu}}_z = \alpha \cdot \tilde{\boldsymbol{\mu}}_x + (1 - \alpha) \cdot \tilde{\boldsymbol{\mu}}_y = (\mu_{p,z}, 1)'$. Result 2. follows from the definition of an efficient portfolio.

Example 12.9. Create an arbitrary efficient frontier portfolio from two efficient minimum variance portfolios.

Consider the data in Table 1 and the previously computed minimum variance portfolios that have the same expected return as Microsoft and Starbucks, respectively, and let $\alpha = 0.5$. From (12.22), the frontier portfolio \mathbf{z} is constructed using:

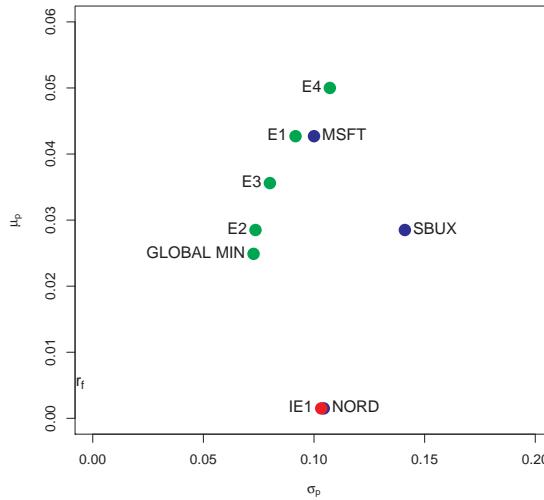


Fig. 12.6 Minimum variance portfolios created as convex combinations of two minimum variance portfolios. Portfolios E3, E4 and IE1 are created from portfolios E1 and E2.

$$\begin{aligned}
 \mathbf{z} &= \alpha \cdot \mathbf{x} + (1 - \alpha) \cdot \mathbf{y} \\
 &= 0.5 \cdot \begin{pmatrix} 0.8275 \\ -0.0907 \\ 0.2633 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 0.519 \\ 0.273 \\ 0.207 \end{pmatrix} \\
 &= \begin{pmatrix} (0.5)(0.8275) \\ (0.5)(-0.0907) \\ (0.5)(0.2633) \end{pmatrix} + \begin{pmatrix} (0.5)(0.519) \\ (0.5)(0.273) \\ (0.5)(0.207) \end{pmatrix} \\
 &= \begin{pmatrix} 0.6734 \\ 0.0912 \\ 0.2354 \end{pmatrix} = \begin{pmatrix} z_A \\ z_B \\ z_C \end{pmatrix}.
 \end{aligned}$$

In R, the new frontier portfolio is computed using:

```
a = 0.5
z.vec = a * x.vec + (1 - a) * y.vec
z.vec

##   MSFT     NORD     SBUX
## 0.6734  0.0912  0.2354
```

Using $\mu_{p,z} = \mathbf{z}'\boldsymbol{\mu}$ and $\sigma_{p,z}^2 = \mathbf{z}'\boldsymbol{\Sigma}\mathbf{z}$, the expected return, variance and standard deviation of this portfolio are:

```

mu.pz = as.numeric(crossprod(z.vec, mu.vec))
sig2.pz = as.numeric(t(z.vec) %*% sigma.mat %*% z.vec)
sig.pz = sqrt(sig2.pz)
mu.pz

## [1] 0.0356

sig.pz

## [1] 0.0801

```

Equivalently, using $\mu_{p,z} = \alpha\mu_{p,x} + (1-\alpha)\mu_{p,y}$ and $\sigma_{p,z}^2 = \alpha^2\sigma_{p,x}^2 + (1-\alpha)^2\sigma_{p,y}^2 + 2\alpha(1-\alpha)\sigma_{xy}$ the expected return, variance and standard deviation of this portfolio are:

```

mu.pz = a * mu.px + (1 - a) * mu.py
sig.xy = as.numeric(t(x.vec) %*% sigma.mat %*% y.vec)
sig2.pz = a^2 * sig2.px + (1 - a)^2 * sig2.py + 2 * a * (1 - a) * sig.xy
sig.pz = sqrt(sig2.pz)
mu.pz

## [1] 0.0356

sig.pz

## [1] 0.0801

```

Because $\mu_{p,z} = 0.0356 > \mu_{p,m} = 0.0249$ the frontier portfolio **z** is an efficient frontier portfolio. The three efficient frontier portfolios **x**, **y** and **z** are illustrated in Figure 12.6 and are labeled "E1", "E2" and "E3", respectively.

■

Example 12.10. Create an efficient frontier portfolio with a given expected return from two efficient minimum variance portfolios

Given the two minimum variance portfolios (??) and (??) with target expected returns equal to the expected returns on Microsoft and Starbucks, respectively, consider creating a frontier portfolio with target expected return equal to 0.05. To determine the value of α that corresponds with this portfolio we use the equation

$$\mu_{p,z} = \alpha_{.05}\mu_{p,x} + (1 - \alpha_{.05})\mu_{p,y} = 0.05.$$

We can then solve for $\alpha_{.05}$:

$$\alpha_{.05} = \frac{0.05 - \mu_{p,y}}{\mu_{p,x} - \mu_{p,y}} = \frac{0.05 - 0.0285}{0.0427 - 0.0285} = 1.51.$$

Given $\alpha_{.05}$ we can solve for the portfolio weights using

$$\begin{aligned}
 \mathbf{z}_{.05} &= \alpha_{.05}\mathbf{x} + (1 - \alpha_{.05})\mathbf{y} \\
 &= 1.51 \times \begin{pmatrix} 0.8275 \\ -0.0907 \\ 0.2633 \end{pmatrix} - 0.514 \times \begin{pmatrix} 0.519 \\ 0.273 \\ 0.207 \end{pmatrix} \\
 &= \begin{pmatrix} 0.986 \\ -0.278 \\ 0.292 \end{pmatrix}.
 \end{aligned}$$

We can then compute the mean and variance of this portfolio using $\mu_{p,z_{.05}} = \mathbf{z}'_{.05}\boldsymbol{\mu}$ and $\sigma^2_{p,z_{.05}} = \mathbf{z}'_{.05}\boldsymbol{\Sigma}\mathbf{z}_{.05}$. Using R, the calculations are:

```

# compute alpha
a.05 = (0.05 - mu.py)/(mu.px - mu.py)
a.05

## [1] 1.51

# compute weights
z.05 = a.05 * x.vec + (1 - a.05) * y.vec
z.05

##    MSFT    NORD    SBUX
##  0.986 -0.278  0.292

# compute mean and volatility
mu.pz.05 = as.numeric(crossprod(z.05, mu.vec))
sig.pz.05 = as.numeric(sqrt(t(z.05) %*% sigma.mat %*% z.05))
mu.pz.05

## [1] 0.05

sig.pz.05

## [1] 0.107

```

This portfolio is labeled "E4" in Figure 12.6.

Example 12.11. Create an inefficient frontier portfolio with a given expected return from two minimum variance portfolios

Given the two minimum variance portfolios with target expected returns equal to the expected returns on Microsoft and Starbucks, respectively, consider creating a frontier portfolio with target expected return equal to the expected return on Nordstrom. Then,

$$\mu_{p,z} = \alpha_{nord}\mu_{p,x} + (1 - \alpha_{nord})\mu_{p,y} = \mu_{nord} = 0.0015,$$

and we can solve for α_{nord} using:

$$\alpha_{nord} = \frac{\mu_{nord} - \mu_{p,y}}{\mu_{p,x} - \mu_{p,y}} = \frac{0.0015 - 0.0285}{0.0427 - 0.0285} = -1.901.$$

The portfolio weights are:

$$\begin{aligned}\mathbf{z}_{nord} &= \alpha_{nord}\mathbf{x} + (1 - \alpha_{nord})\mathbf{y} \\ &= -1.901 \times \begin{pmatrix} 0.8275 \\ -0.0907 \\ 0.2633 \end{pmatrix} + 2.9 \times \begin{pmatrix} 0.519 \\ 0.273 \\ 0.207 \end{pmatrix} \\ &= \begin{pmatrix} -0.0064 \\ 0.9651 \\ 0.1013 \end{pmatrix}.\end{aligned}$$

Using R, the calculations are:

```
# compute alpha
a.nord = (mu.vec["NORD"] - mu.py)/(mu.px - mu.py)
a.nord

## NORD
## -1.9

# compute weights
z.nord = a.nord * x.vec + (1 - a.nord) * y.vec
z.nord

## MSFT      NORD      SBUX
## -0.0664  0.9651  0.1013

# compute mean and volatility
mu.pz.nord = as.numeric(crossprod(z.nord, mu.vec))
sig.pz.nord = as.numeric(sqrt(t(z.nord) %*% sigma.mat %*% z.nord))
mu.pz.nord

## [1] 0.0015

sig.pz.nord

## [1] 0.103
```

Because $\mu_{p,z} = 0.0015 < \mu_{p,m} = 0.02489$ the frontier portfolio \mathbf{z} is an inefficient frontier portfolio. This portfolio is labeled "IE4" in Figure 12.6.

■

12.4.1 Algorithm for computing efficient frontier

The efficient frontier of portfolios, i.e., those frontier portfolios with expected return greater than the expected return on the global minimum variance portfolio, can be conveniently created using (12.22) with two specific efficient portfolios. The first efficient portfolio is

the global minimum variance portfolio (12.2). The second efficient portfolio is the efficient portfolio whose target expected return is equal to the highest expected return among all of the assets under consideration. The choice of these two efficient portfolios makes it easy to produce a nice plot of the efficient frontier. The steps for constructing the efficient frontier are:

1. Compute the global minimum variance portfolio \mathbf{m} by solving (12.2), and compute $\mu_{p,m} = \mathbf{m}'\boldsymbol{\mu}$ and $\sigma_{p,m}^2 = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{m}$.
2. Compute the efficient portfolio \mathbf{x} with target expected return equal to the maximum expected return of the assets under consideration. That is, solve (12.10) with $\mu_0 = \max\{\mu_1, \dots, \mu_N\}$, and compute $\mu_{p,x} = \mathbf{x}'\boldsymbol{\mu}$ and $\sigma_{p,m}^2 = \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x}$.
3. Compute $\text{cov}(R_{p,m}, R_{p,x}) = \sigma_{mx} = \mathbf{m}'\boldsymbol{\Sigma}\mathbf{x}$.
4. Create an initial grid of α values $\{0, 0.1, \dots, 0.9, 1\}$, and compute the frontier portfolios \mathbf{z} using

$$\mathbf{z} = \alpha \times \mathbf{x} + (1 - \alpha) \times \mathbf{m},$$

and compute their expected returns and variances using (12.22), (12.23) and (12.24), respectively.

5. Plot $\mu_{p,z}$ against $\sigma_{p,z}$ and adjust the grid of α values appropriately to create a nice plot. Negative values of α will give inefficient frontier portfolios. Values of α greater than one will give efficient frontier portfolios with expected returns greater than μ_0 .

Example 12.12. Compute and plot the efficient frontier of risky assets (Markowitz bullet).

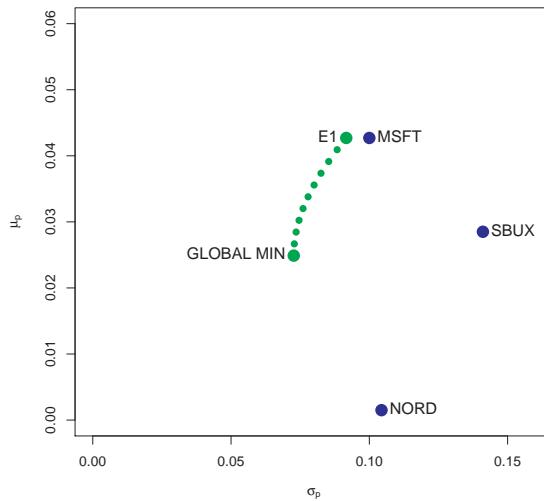


Fig. 12.7 Efficient frontier computed from example data.

To compute the efficient frontier from the three risky assets in Table 12.1 in R use:

```

a = seq(from = 0, to = 1, by = 0.1)
n.a = length(a)
z.mat = matrix(0, n.a, 3)
colnames(z.mat) = names(mu.vec)
mu.z = rep(0, n.a)
sig2.z = rep(0, n.a)
sig.mx = t(m.vec) %*% sigma.mat %*% x.vec
for (i in 1:n.a) {
  z.mat[i, ] = a[i] * x.vec + (1 - a[i]) * m.vec
  mu.z[i] = a[i] * mu.px + (1 - a[i]) * mu.gmin
  sig2.z[i] = a[i]^2 * sig2.px + (1 - a[i])^2 * sig2.gmin + 2 * a[i] * (1 - a[i]) *
    sig.mx
}

```

The variables `z.mat`, `mu.z` and `sig2.z` contain the weights, expected returns and variances, respectively, of the efficient frontier portfolios for a grid of α values between 0 and 1. The resulting efficient frontier is illustrated in Figure 12.7 created with:

```

plot(sqrt(sig2.z), mu.z, type = "b", ylim = c(0, 0.06), xlim = c(0, 0.16), pch = 16,
  col = "green", cex = cex.val, ylab = expression(mu[p]), xlab = expression(sigma[p]))
points(sd.vec, mu.vec, pch = 16, cex = 2, lwd = 2, col = "blue")
points(sig.gmin, mu.gmin, pch = 16, col = "green", cex = 2)
points(sig.px, mu.px, pch = 16, col = "green", cex = 2)
text(sig.gmin, mu.gmin, labels = "GLOBAL MIN", pos = 2, cex = cex.val)
text(sd.vec, mu.vec, labels = asset.names, pos = 4, cex = cex.val)
text(sig.px, mu.px, labels = "E1", pos = 2, cex = cex.val)

```

Each point on the efficient frontier is a portfolio of Microsoft, Nordstrom and Starbucks. It is instructive to visualize the weights in these portfolios as we move along the frontier from the global minimum variance portfolio to the efficient portfolio with expected return equal to the expected return on Microsoft. We can do this easily using the **PerformanceAnalytics** function `chart.StackedBar()`:

```

library(PerformanceAnalytics)
chart.StackedBar(z.mat, xaxis.labels = round(sqrt(sig2.z), digits = 3), xlab = "Portfolio SD",
  ylab = "Weights")

```

The resulting plot is shown in Figure 12.8. As we move along the frontier, the allocation to Microsoft increases and the allocation to Nordstrom decreases whereas the allocation to Starbucks stays about the same.

12.5 Computing Efficient Portfolios of N risky Assets and a Risk-Free Asset Using Matrix Algebra

In Chapter 11, we showed that efficient portfolios of two risky assets and a single risk-free (T-Bill) asset are portfolios consisting of the highest Sharpe ratio portfolio (tangency portfolio) and the T-Bill. With three or more risky assets and a T-Bill the same result holds.

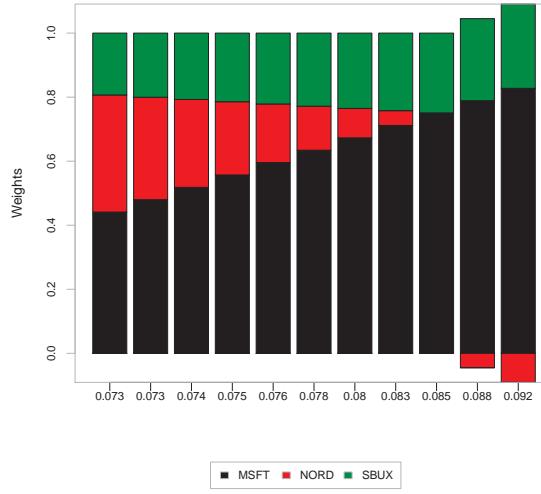


Fig. 12.8 Portfolio weights in efficient frontier portfolios.

12.5.1 Computing the tangency portfolio using matrix algebra

The tangency portfolio is the portfolio of risky assets that has the highest Sharpe ratio. The tangency portfolio, denoted $\mathbf{t} = (t_1, \dots, t_N)'$, solves the constrained maximization problem:

$$\max_{\mathbf{t}} \frac{\mathbf{t}'\mu - r_f}{(\mathbf{t}'\Sigma\mathbf{t})^{\frac{1}{2}}} = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}} \text{ s.t. } \mathbf{t}'\mathbf{1} = 1, \quad (12.25)$$

where $\mu_{p,t} = \mathbf{t}'\mu$ and $\sigma_{p,t} = (\mathbf{t}'\Sigma\mathbf{t})^{\frac{1}{2}}$. The Lagrangian for this problem is:

$$L(\mathbf{t}, \lambda) = (\mathbf{t}'\mu - r_f)(\mathbf{t}'\Sigma\mathbf{t})^{-\frac{1}{2}} + \lambda(\mathbf{t}'\mathbf{1} - 1).$$

Using the chain rule, the first order conditions are:

$$\begin{aligned} \frac{\partial L(\mathbf{t}, \lambda)}{\partial \mathbf{t}} &= \mu(\mathbf{t}'\Sigma\mathbf{t})^{-\frac{1}{2}} - (\mathbf{t}'\mu - r_f)(\mathbf{t}'\Sigma\mathbf{t})^{-3/2}\Sigma\mathbf{t} + \lambda\mathbf{1} = \mathbf{0}, \\ \frac{\partial L(\mathbf{t}, \lambda)}{\partial \lambda} &= \mathbf{t}'\mathbf{1} - 1 = 0. \end{aligned}$$

After much tedious algebra, it can be shown that the solution for \mathbf{t} has a nice simple expression:

$$\mathbf{t} = \frac{\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}{\mathbf{1}'\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}. \quad (12.26)$$

The formula for the tangency portfolio (12.26) looks similar to the formula for the global minimum variance portfolio (12.8). Both formulas have Σ^{-1} in the numerator and $\mathbf{1}'\Sigma^{-1}$ in the denominator.

The location of the tangency portfolio, and the sign of the Sharpe ratio, depends on the relationship between the risk-free rate r_f and the expected return on the global minimum variance portfolio $\mu_{p,m}$. If $\mu_{p,m} > r_f$, which is the usual case, then the tangency portfolio will have a positive Sharpe ratio. If $\mu_{p,m} < r_f$, which could occur when stock prices are falling and the economy is in a recession, then the tangency portfolio will have a negative Sharpe slope. In this case, efficient portfolios involve shorting the tangency portfolio and investing the proceeds in T-Bills.⁴

Example 12.13. Computing the tangency portfolio.

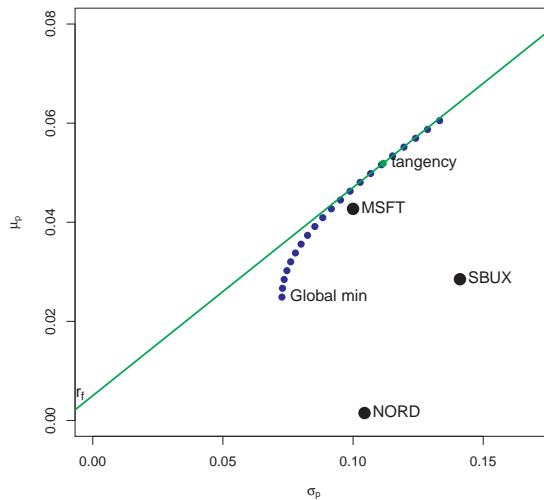


Fig. 12.9 Tangency portfolio from example data.

Suppose $r_f = 0.005$. To compute the tangency portfolio (12.26) in R for the three risky assets in Table 12.1 use:

```
##   MSFT    NORD    SBUX
## 1.027 -0.326  0.299
```

The tangency portfolio has weights $t_{\text{msft}} = 1.027$, $t_{\text{nord}} = -0.326$ and $t_{\text{sbux}} = 0.299$, and is given by the vector $\mathbf{t} = (1.027, -0.326, 0.299)'$. Notice that Nordstrom, which has the lowest mean return, is sold short in the tangency portfolio. The expected return on the tangency portfolio, $\mu_{p,t} = \mathbf{t}'\boldsymbol{\mu}$, is:

```
mu.t = as.numeric(crossprod(t.vec, mu.vec))
mu.t
## [1] 0.0519
```

⁴ For a mathematical proof of these results, see Ingersoll (1987).

The portfolio variance, $\sigma_{p,t}^2 = \mathbf{t}'\Sigma\mathbf{t}$, and standard deviation, $\sigma_{p,t}$, are:

```
sig2.t = as.numeric(t(t.vec) %*% sigma.mat %*% t.vec)
sig.t = sqrt(sig2.t)
sig2.t

## [1] 0.0125

sig.t

## [1] 0.112
```

Because $r_f = 0.005 < \mu_{p,m} = 0.0249$ the tangency portfolio has a positive Sharpe's ratio/slope given by:

```
SR.t = (mu.t - r.f)/sig.t
SR.t

## [1] 0.42
```

The tangency portfolio is illustrated in Figure 12.9. It is the portfolio on the efficient frontier of risky assets in which a straight line drawn from the risk-free rate to the tangency portfolio (green line) is just tangent to the efficient frontier (blue dots). ■

12.5.2 Alternative derivation of the tangency portfolio

The derivation of tangency portfolio formula (12.26) from the optimization problem (12.25) is a very tedious problem. It can be derived in a different way as follows. Consider forming portfolios of N risky assets with return vector \mathbf{R} and T-bills (risk-free asset) with constant return r_f . Let \mathbf{x} denote the vector of risky asset weights and let x_f denote the safe asset weight and assume that $\mathbf{x}'\mathbf{1} + x_f = 1$ so that all wealth is allocated to these assets. The portfolio return is:

$$R_{p,x} = \mathbf{x}'\mathbf{R} + x_f r_f = \mathbf{x}'\mathbf{R} + (1 - \mathbf{x}'\mathbf{1})r_f = r_f + \mathbf{x}'(\mathbf{R} - r_f \cdot \mathbf{1}).$$

The portfolio excess return is:

$$R_{p,x} - r_f = \mathbf{x}'(\mathbf{R} - r_f \cdot \mathbf{1}). \quad (12.27)$$

The expected portfolio return excess return (risk premium) and portfolio variance are:

$$\mu_{p,x} - r_f = \mathbf{x}'(\mu - r_f \cdot \mathbf{1}), \quad (12.28)$$

$$\sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x}. \quad (12.29)$$

For notational simplicity, define $\tilde{\mathbf{R}} = \mathbf{R} - r_f \cdot \mathbf{1}$, $\tilde{\mu} = \mu - r_f \cdot \mathbf{1}$, $\tilde{R}_{p,x} = R_{p,x} - r_f$, and $\tilde{\mu}_{p,x} = \mu_{p,x} - r_f$. Then (12.27) and (12.28) can be re-expressed as:

$$\tilde{R}_{p,x} = \mathbf{x}'\tilde{\mathbf{R}}, \quad (12.30)$$

$$\tilde{\mu}_{p,x} = \mathbf{x}'\tilde{\mu}. \quad (12.31)$$

To find the minimum variance portfolio of risky assets and a risk free asset that achieves the target excess return $\tilde{\mu}_{p,0} = \mu_{p,0} - r_f$ we solve the minimization problem:

$$\min_{\mathbf{x}} \sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x} \text{ s.t. } \tilde{\mu}_{p,x} = \tilde{\mu}_{p,0}.$$

Note that $\mathbf{x}'\mathbf{1} = 1$ is not a constraint because wealth need not all be allocated to the risky assets; some wealth may be held in the riskless asset. The Lagrangian is:

$$L(\mathbf{x}, \lambda) = \mathbf{x}'\Sigma\mathbf{x} + \lambda(\mathbf{x}'\tilde{\mu} - \tilde{\mu}_{p,0}).$$

The first order conditions for a minimum are:

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = 2\Sigma\mathbf{x} + \lambda\tilde{\mu} = \mathbf{0}, \quad (12.32)$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda} = \mathbf{x}'\tilde{\mu} - \tilde{\mu}_{p,0} = 0. \quad (12.33)$$

Using the first equation (12.32), we can solve for \mathbf{x} in terms of λ :

$$\mathbf{x} = -\frac{1}{2}\lambda\Sigma^{-1}\tilde{\mu}. \quad (12.34)$$

The second equation (12.33) implies that $\mathbf{x}'\tilde{\mu} = \tilde{\mu}'\mathbf{x} = \tilde{\mu}_{p,0}$. Then pre-multiplying (12.34) by $\tilde{\mu}'$ gives:

$$\tilde{\mu}'\mathbf{x} = -\frac{1}{2}\lambda\tilde{\mu}'\Sigma^{-1}\tilde{\mu} = \tilde{\mu}_{p,0},$$

which we can use to solve for λ :

$$\lambda = -\frac{2\tilde{\mu}_{p,0}}{\tilde{\mu}'\Sigma^{-1}\tilde{\mu}}. \quad (12.35)$$

Plugging (12.35) into (12.34) then gives the solution for \mathbf{x} :

$$\mathbf{x} = -\frac{1}{2}\lambda\Sigma^{-1}\tilde{\mu} = -\frac{1}{2}\left(-\frac{2\tilde{\mu}_{p,0}}{\tilde{\mu}'\Sigma^{-1}\tilde{\mu}}\right)\Sigma^{-1}\tilde{\mu} = \tilde{\mu}_{p,0} \cdot \frac{\Sigma^{-1}\tilde{\mu}}{\tilde{\mu}'\Sigma^{-1}\tilde{\mu}}. \quad (12.36)$$

The solution for x_f is then $1 - \mathbf{x}'\mathbf{1}$.

Now, the tangency portfolio \mathbf{t} is 100% invested in risky assets so that $\mathbf{t}'\mathbf{1} = \mathbf{1}'\mathbf{t} = 1$. Using (12.36), the tangency portfolio satisfies:

$$\mathbf{1}'\mathbf{t} = \tilde{\mu}_{p,t} \cdot \frac{\mathbf{1}'\Sigma^{-1}\tilde{\mu}}{\tilde{\mu}'\Sigma^{-1}\tilde{\mu}} = 1,$$

which implies that,

$$\tilde{\mu}_{p,t} = \frac{\tilde{\mu}'\Sigma^{-1}\tilde{\mu}}{\mathbf{1}'\Sigma^{-1}\tilde{\mu}}. \quad (12.37)$$

Plugging (12.37) back into (12.36) then gives an explicit solution for \mathbf{t} :

$$\begin{aligned}\mathbf{t} &= \left(\frac{\tilde{\mu}' \Sigma^{-1} \tilde{\mu}}{\mathbf{1}' \Sigma^{-1} \tilde{\mu}} \right) \frac{\Sigma^{-1} \tilde{\mu}}{\tilde{\mu}' \Sigma^{-1} \tilde{\mu}} = \frac{\Sigma^{-1} \tilde{\mu}}{\mathbf{1}' \Sigma^{-1} \tilde{\mu}} \\ &= \frac{\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}{\mathbf{1}' \Sigma^{-1}(\mu - r_f \cdot \mathbf{1})},\end{aligned}$$

which is the result (12.26) we got from finding the portfolio of risky assets that has the maximum Sharpe ratio.

12.5.3 Mutual fund separation theorem again

When there is a risk-free asset (T-bill) available, the efficient frontier of T-bills and risky assets consists of portfolios of T-bills and the tangency portfolio. The expected return and standard deviation values of any such efficient portfolio are given by:

$$\mu_p^e = r_f + x_t(\mu_{p,t} - r_f), \quad (12.38)$$

$$\sigma_p^e = x_t \sigma_{p,t}, \quad (12.39)$$

where x_t represents the fraction of wealth invested in the tangency portfolio ($1 - x_t$ represents the fraction of wealth invested in T-Bills), and $\mu_{p,t} = \mathbf{t}'\mu$ and $\sigma_{p,t} = (\mathbf{t}'\Sigma\mathbf{t})^{1/2}$ are the expected return and standard deviation on the tangency portfolio, respectively. Recall, this result is known as the *mutual fund separation theorem*. The tangency portfolio can be considered as a mutual fund of the risky assets, where the shares of the assets in the mutual fund are determined by the tangency portfolio weights, and the T-bill can be considered as a mutual fund of risk-free assets. The expected return-risk trade-off of these portfolios is given by the line connecting the risk-free rate to the tangency point on the efficient frontier of risky asset only portfolios. Which combination of the tangency portfolio and the T-bill an investor will choose depends on the investor's risk preferences. If the investor is very risk averse and prefers portfolios with very low volatility, then she will choose a combination with very little weight in the tangency portfolio and a lot of weight in the T-bill. This will produce a portfolio with an expected return close to the risk-free rate and a variance that is close to zero. If the investor can tolerate a large amount of volatility, then she will prefer a portfolio with a high expected return regardless of volatility. This portfolio may involve borrowing at the risk-free rate (leveraging) and investing the proceeds in the tangency portfolio to achieve a high expected return.

Example 12.14. Efficient portfolios of three risky assets and T-bills chosen by risk averse and risk tolerant investors.

Consider the tangency portfolio computed from the example data in Table 12.1 with $r_f = 0.005$. This portfolio is:

```
t.vec
##   MSFT    NORD    SBUX
## 1.027 -0.326  0.299

mu.t
```

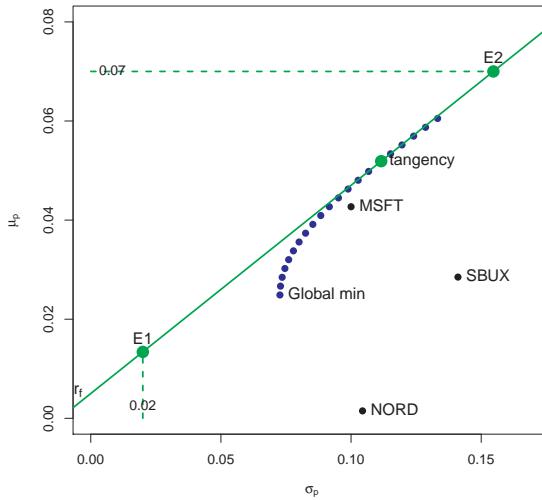


Fig. 12.10

```

## [1] 0.0519
sig.t
## [1] 0.112

```

The efficient portfolios of T-Bills and the tangency portfolio is illustrated in Figure 12.10.

We want to compute an efficient portfolio that would be preferred by a highly risk averse investor, and a portfolio that would be preferred by a highly risk tolerant investor. A highly risk averse investor might have a low volatility (risk) target for his efficient portfolio. For example, suppose the volatility target is $\sigma_p^e = 0.02$ or 2%. Using (12.39) and solving for x_t , the weights in the tangency portfolio and the T-Bill are:

```

x.t.02 = 0.02/sig.t
x.t.02

## [1] 0.179
1 - x.t.02
## [1] 0.821

```

In this efficient portfolio, the weights in the risky assets are proportional to the weights in the tangency portfolio:

```

x.t.02 * t.vec
##      MSFT      NORD      SBUX
##  0.1840 -0.0585  0.0537

```

The expected return and volatility values of this portfolio are:

```
mu.t.02 = x.t.02 * mu.t + (1 - x.t.02) * r.f
sig.t.02 = x.t.02 * sig.t
mu.t.02

## [1] 0.0134

sig.t.02

## [1] 0.02
```

These values are illustrated in Figure 12.10 as the portfolio labeled “E1”.

A highly risk tolerant investor might have a high expected return target for his efficient portfolio. For example, suppose the expected return target is $\mu_p^e = 0.07$ or 7%. Using (12.38) and solving for the x_t , the weights in the tangency portfolio and the T-Bill are:

```
x.t.07 = (0.07 - r.f)/(mu.t - r.f)
x.t.07

## [1] 1.39

1 - x.t.07

## [1] -0.386
```

Notice that this portfolio involves borrowing at the T-Bill rate (leveraging) and investing the proceeds in the tangency portfolio. In this efficient portfolio, the weights in the risky assets are:

```
x.t.07 * t.vec

##   MSFT    NORD    SBUX
## 1.423 -0.452  0.415
```

The expected return and volatility values of this portfolio are:

```
mu.t.07 = x.t.07 * mu.t + (1 - x.t.07) * r.f
sig.t.07 = x.t.07 * sig.t
mu.t.07

## [1] 0.07

sig.t.07

## [1] 0.155
```

In order to achieve the target expected return of 7%, the investor must tolerate a 15.47% volatility. These values are illustrated in Figure 12.10 as the portfolio labeled “E2”. ■

12.6 Computational Problems with Very Large Portfolios

In principle, mean-variance portfolio analysis can be applied in situations in which there is a very large number of risky assets (e.g., $N = 5,000$). However, there are a number of practical problems that can arise. First, the computation of efficient portfolios requires inverting the $N \times N$ asset return covariance matrix Σ . When N is very large, inverting Σ can be computationally burdensome. Second, the practical application of the theory requires the estimation of Σ . Recall, there are N variance terms and $N(N - 1)/2$ unique covariance terms in Σ . When $N = 5,000$ there are 12,502,500 unique elements of Σ to estimate. And since each estimated element of Σ has estimation error, there is a tremendous amount of estimation error in the estimate of Σ . There is an additional problem with the estimation of Σ using the sample covariance matrix of asset returns when N is very large. If the number of assets, N , is greater than the number of sample observations, T , then the $N \times N$ sample covariance matrix

$$\hat{\Sigma} = \frac{1}{T-1} \sum_{t=1}^T (\mathbf{R}_t - \hat{\mu})(\mathbf{R}_t - \hat{\mu})',$$

$$\hat{\mu} = \frac{1}{T} \sum_{t=1}^T \mathbf{R}_t,$$

is only positive semi-definite and less than full rank N . This means that $\hat{\Sigma}$ is not invertible and so mean-variance efficient portfolios cannot be uniquely computed. This problem can happen often. For example, suppose $N = 5,000$. For the sample covariance matrix to be full rank, you need at least $T = 5,000$ sample observations. For daily data, this mean you would need $5,000/250 = 20$ years of daily data.⁵ For weekly data, you would need $5000/52 = 96.2$ years of weekly data. For monthly data, you would need $5,000/12 = 417$ years of monthly data.

Example 12.15. Nonsingular sample return covariance matrix.

To be completed. ■

Due to these practical problems of using the sample covariance matrix $\hat{\Sigma}$ to compute mean-variance efficient portfolios when N is large, there is a need for alternative methods for estimating Σ when N is large. One such method based on the Single Index Model for returns is presented in Chapter 16.

12.7 Portfolio Analysis Functions in R

The package **IntroCompFinR** contains a few R functions for computing Markowitz mean-variance efficient portfolios allowing for short sales using matrix algebra computations. These

⁵ Recall, there are approximately 250 trading days per year.

functions allow for the easy computation of the global minimum variance portfolio, an efficient portfolio with a given target expected return, the tangency portfolio, and the efficient frontier. These functions are summarized in Table 12.2.

Function	Description
<code>getPortfolio</code>	create portfolio object
<code>globalMin.portfolio</code>	compute global minimum variance portfolio
<code>efficient.portfolio</code>	compute minimum variance portfolio subject to target return
<code>tangency.portfolio</code>	compute tangency portfolio
<code>efficient.frontier</code>	compute efficient frontier of risky assets

Table 12.2 `IntroCompFinR` functions for computing mean-variance efficient portfolios

The following examples illustrate the use of the functions in Table 12.2 using the example data in Table 12.1:

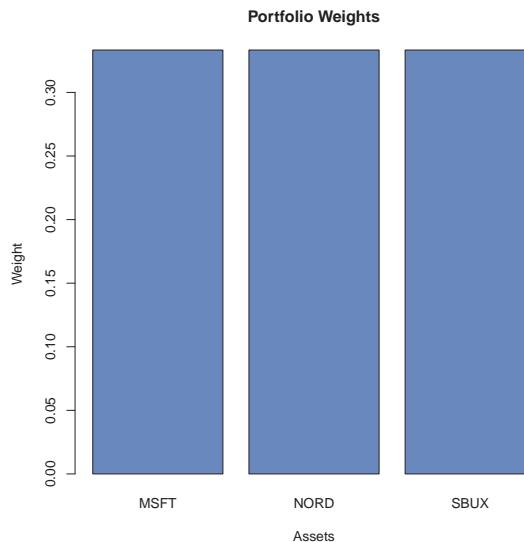


Fig. 12.11 plot method for objects of class “portfolio”

```
mu.vec

##    MSFT    NORD    SBUX
## 0.0427 0.0015 0.0285

sigma.mat

##          MSFT    NORD    SBUX
## MSFT 0.0100 0.0018 0.0011
```

```

## NORD 0.0018 0.0109 0.0026
## SBUX 0.0011 0.0026 0.0199

r.f

## [1] 0.005

```

To specify a “portfolio” object, you need an expected return vector and covariance matrix for the assets under consideration as well as a vector of portfolio weights. To create an equally weighted portfolio use:

```

library(IntroCompFinR)
ew = rep(1, 3)/3
equalWeight.portfolio = getPortfolio(er = mu.vec, cov.mat = sigma.mat, weights = ew)
class(equalWeight.portfolio)

## [1] "portfolio"

```

Portfolio objects have the following components:

```

names(equalWeight.portfolio)

## [1] "call"      "er"        "sd"        "weights"

```

There are `print()`, `summary()` and `plot()` methods for portfolio objects. The `print()` method gives:

```

equalWeight.portfolio

## Call:
## getPortfolio(er = mu.vec, cov.mat = sigma.mat, weights = ew)
##
## Portfolio expected return: 0.0242
## Portfolio standard deviation: 0.0759
## Portfolio weights:
## MSFT NORD SBUX
## 0.333 0.333 0.333

```

The `plot()` method shows a bar chart of the portfolio weights:

```
plot(equalWeight.portfolio, col = "cornflowerblue")
```

The resulting plot is shown in Figure 12.11. The global minimum variance portfolio (allowing for short sales) `m` solves the optimization problem (12.2) and is computed using (12.6). To compute this portfolio use the function `globalMin.portfolio()`:

```

gmin.port = globalMin.portfolio(mu.vec, sigma.mat)
class(gmin.port)

## [1] "portfolio"

gmin.port

```

```
## Call:
## globalMin.portfolio(er = mu.vec, cov.mat = sigma.mat)
##
## Portfolio expected return: 0.0249
## Portfolio standard deviation: 0.0727
## Portfolio weights:
## MSFT NORD SBUX
## 0.441 0.366 0.193
```

A mean-variance efficient portfolio \mathbf{x} that achieves the target expected return μ_0 solves the optimization problem (12.10) and is computed using (12.14). To compute this portfolio for the target expected return $\mu_0 = E[R_{\text{msft}}] = 0.0427$ use the `efficient.portfolio()` function:

```
target.return = mu.vec[1]
e.port.msft = efficient.portfolio(mu.vec, sigma.mat, target.return)
class(e.port.msft)

## [1] "portfolio"

e.port.msft

## Call:
## efficient.portfolio(er = mu.vec, cov.mat = sigma.mat, target.return = target.return)
##
## Portfolio expected return: 0.0427
## Portfolio standard deviation: 0.0917
## Portfolio weights:
## MSFT NORD SBUX
## 0.8275 -0.0907 0.2633
```

The tangency portfolio \mathbf{t} is the portfolio of risky assets with the highest Sharpe's slope and has solutions given by (12.26). To compute this portfolio with $r_f = 0.005$ use the `tangency.portfolio()` function:

```
tan.port = tangency.portfolio(mu.vec, sigma.mat, r.f)
class(tan.port)

## [1] "portfolio"

tan.port

## Call:
## tangency.portfolio(er = mu.vec, cov.mat = sigma.mat, risk.free = r.f)
##
## Portfolio expected return: 0.0519
## Portfolio standard deviation: 0.112
## Portfolio weights:
## MSFT NORD SBUX
## 1.027 -0.326 0.299
```

The the set of efficient portfolios of risky assets can be computed as a convex combination of any two efficient portfolios. It is convenient to use the global minimum variance portfolio as one portfolio and an efficient portfolio with target expected return equal to the maximum

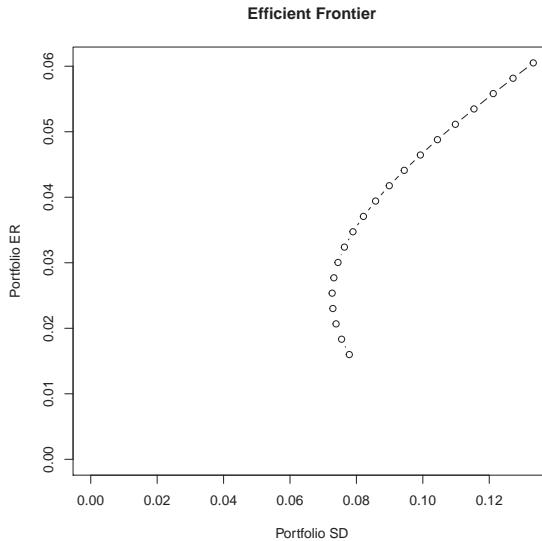


Fig. 12.12 Plot method for objects of class “Markowitz”

expected return of the assets under consideration as the other portfolio. Call these portfolios \mathbf{m} and \mathbf{x} , respectively. For any number α , another efficient portfolio can be computed as:

$$\mathbf{z} = \alpha\mathbf{x} + (1 - \alpha)\mathbf{m}$$

The function `efficient.frontier()` constructs the set of efficient portfolios using this method for a collection of α values on an equally spaced grid between α_{min} and α_{max} . For example, to compute 20 efficient portfolios for values of α between -2 and 1.5 use:

```
ef = efficient.frontier(mu.vec, sigma.mat, alpha.min = -0.5, alpha.max = 2, nport = 20)
attributes(ef)

## $names
## [1] "call"      "er"        "sd"        "weights"
##
## $class
## [1] "Markowitz"

ef

## Call:
## efficient.frontier(er = mu.vec, cov.mat = sigma.mat, nport = 20,
## alpha.min = -0.5, alpha.max = 2)
##
## Frontier portfolios' expected returns and standard deviations
##   port 1 port 2 port 3 port 4 port 5 port 6 port 7 port 8 port 9 port 10
##   ER 0.0160 0.0183 0.0207 0.0230 0.0254 0.0277 0.0300 0.0324 0.0347 0.0371
##   SD 0.0779 0.0755 0.0739 0.0729 0.0727 0.0732 0.0745 0.0764 0.0790 0.0821
##   port 11 port 12 port 13 port 14 port 15 port 16 port 17 port 18 port 19
##   ER 0.0394 0.0418 0.0441 0.0464 0.0488 0.0511 0.0535 0.0558 0.0582
```

```
## SD  0.0858  0.0899  0.0944  0.0993  0.1044  0.1098  0.1154  0.1212  0.1272
##      port 20
## ER   0.0605
## SD   0.1333
```

Use the `summary()` method to show the weights of these portfolios. Use the `plot()` method to create a simple plot the efficient frontier:

```
plot(ef)
```

The resulting plot is shown in Figure 12.12. To create a more elaborate plot of the efficient frontier showing the original assets and the tangency portfolio use:

```
plot(ef, plot.assets = T, col = "blue", pch = 16)
points(gmin.port$sd, gmin.port$er, col = "green", pch = 16, cex = 2)
text(gmin.port$sd, gmin.port$er, labels = "Global min", pos = 4)
points(tan.port$sd, tan.port$er, col = "red", pch = 16, cex = 2)
text(tan.port$sd, tan.port$er, labels = "Tangency", pos = 3)
sr.tan = (tan.port$er - r.f)/tan.port$sd
abline(a = r.f, b = sr.tan, col = "green", lwd = 2)
```

The resulting plot is shown in Figure 12.13.

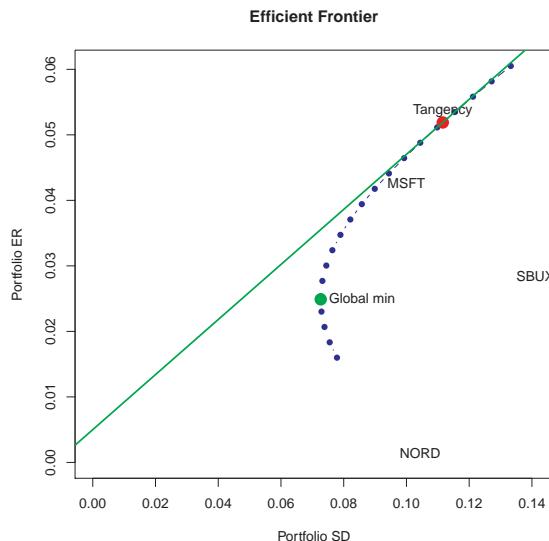


Fig. 12.13 Efficient frontier plot.

12.8 Further Reading

Portfolio theory using matrix algebra is discussed in advanced textbooks on asset pricing and portfolio theory. Good treatments are given in Ingersoll (1986), Hwang and Litzenberger (1988), Campbell, Lo and MacKinlay (1996), Cochrane (2008) and Martin, Scherer and Yollin (2016). A nice review of portfolio theory with matrix algebra is (chapter in Finance Handbook). In R, mean-variance optimized portfolios can be computed using the packages **fPortfolio**, **PARMA**, and **PortfolioAnalytics**.

12.9 Exercises

Exercise 12.1. The global minimum variance portfolio $\mathbf{m} = (m_1, m_2, m_3)'$ for the 3 asset case solves the constrained minimization problem:

$$\begin{aligned} \min_{m_1, m_2, m_3} \sigma_{p,m}^2 &= m_1^2 \sigma_1^2 + m_2^2 \sigma_2^2 + m_3^2 \sigma_3^2 \\ &+ 2m_1 m_2 \sigma_{12} + 2m_1 m_3 \sigma_{13} + 2m_2 m_3 \sigma_{23} \\ \text{s.t. } m_1 + m_2 + m_3 &= 1. \end{aligned}$$

The Lagrangian for this problem is:

$$\begin{aligned} L(m_1, m_2, m_3, \lambda) &= m_A^2 \sigma_A^2 + m_B^2 \sigma_B^2 + m_C^2 \sigma_C^2 \\ &+ 2m_A m_B \sigma_{AB} + 2m_A m_C \sigma_{AC} + 2m_B m_C \sigma_{BC} \\ &+ \lambda(m_A + m_B + m_C - 1) \end{aligned}$$

(a) Write out the first order conditions

Exercise: use 2 x 2 matrix inversion rule to explicitly solve for \mathbf{m}

12.10 References

Cochrane, J. (2008). *Asset Pricing*, Second Edition.

Constantinides, G.M., and Malliaris, A.G. (1995). "Portfolio Theory", Chapter 1 in R. Jarow et al., Eds., *Handbooks in OR & MS*, Vol. 9, Elsevier.

Ingersoll, Jr., J.E. (1987). *Theory of Financial Decision Making*, Rowman and Littlefield, Totowa, NJ.

Markowitz, H. (1959). Portfolio Selection: *Efficient Diversification of Investments*, Wiley, New York.

Chapter 13

Portfolio Theory with Short Sales Constraints

Updated: September 26, 2016

Copyright © Eric Zivot 2015, 2016

Restrictions on Short Selling

- Review short selling
- Exchanges prevent short sales in some assets
- Some institutions are prevented from short selling
- Certain accounts do not allow short sales
- Short selling requires substantial credit qualifications

Most investments involve purchases of assets that you believe will increase in value over time. When you buy an asset you establish a “long” position in the asset. You profit from the long position by selling the asset in the future at a price higher than the original purchase price. However, sometimes you may believe that an asset will decrease in value over time. To profit on an investment in which the asset price is lower in the future requires short selling the asset. In general, short selling involves selling an asset you borrow from someone else today and repurchasing the asset in the future and returning it to its owner. You profit on the short sale if you repurchase the asset for a lower price than you initially sold it. Because short selling involves borrowing it is a type of leveraged investment and so risk increases with the size of the short position.

In practice, a short sale of an asset works as follows. Suppose you have an investment account at some brokerage firm and you are interested in short selling one share of an individual stock, such as Microsoft, that you do not currently own. Because the brokerage firm has so many clients, one of the clients will own Microsoft. The brokerage firm will allow you to borrow Microsoft, sell it and add the proceeds of the short sale to your investment account. Doing so establishes a “short” position in Microsoft which is a liability to return the borrowed share sometime in the future. You close the short position in the future by repurchasing one share of Microsoft through the brokerage firm who returns the share to the original owner. However, borrowing is not free. You will need sufficient capital in your account to cover any potential losses from the short sale. In addition, if the price of the short sold asset rises substantially during the short sale then the brokerage firm may require you

to add additional margin capital to your account. If you cannot do this then the brokerage firm will force you to close the short position immediately.

This chapter is organized as follows. Section 13.1 describes the impact of short sales constraints, on risky assets, on efficient portfolios in the simplified two risk asset setting presented in Chapter 11. Section 13.2 discusses the computation of mean-variance efficient portfolios when there are short sales constraints in the general case of n risky assets as described in Chapter 12.

The examples in this chapter use the **IntroCompFinR**, **PerformanceAnalytics**, and **quadprog** packages. Make sure these packages are installed and loaded in R before replicating the chapter examples.

13.1 Portfolio Theory with Short Sales Constraints in a Simplified Setting

In this section, we illustrate the impact of short sales constraints on risky assets on portfolios in the simplified setting described in Chapter 11. We first look at simple portfolios of two risky assets, then consider portfolios of a risky asset plus a risk-free asset, and finish with portfolios of two risky assets plus a risk-free asset.

13.1.1 Two Risky Assets

Consider the two risky asset example data from Chapter 11:

```
mu.A = 0.175
sig.A = 0.258
sig2.A = sig.A^2
mu.B = 0.055
sig.B = 0.115
sig2.B = sig.B^2
rho.AB = -0.164
sig.AB = rho.AB * sig.A * sig.B
assetNames = c("Asset A", "Asset B")
mu.vals = c(mu.A, mu.B)
names(mu.vals) = assetNames
Sigma = matrix(c(sig2.A, sig.AB, sig.AB, sig2.B), 2, 2)
colnames(Sigma) = rownames(Sigma) = assetNames
```

Asset A is the high average return and high risk asset, and asset B is the low average return and low risk asset. Consider the following set of 19 portfolios:

```
x.A = seq(from = -0.4, to = 1.4, by = 0.1)
x.B = 1 - x.A
n.port = length(x.A)
```

```

names(x.A) = names(x.B) = paste("p", 1:n.port, sep = ".")
mu.p = x.A * mu.A + x.B * mu.B
sig2.p = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p = sqrt(sig2.p)

```

Figure 13.1 shows the risky asset only portfolio frontier created using:

```

cex.val = 2
plot(sig.p, mu.p, type = "b", pch = 16, cex = cex.val, ylim = c(0, max(mu.p)),
     xlim = c(0, max(sig.p)), xlab = expression(sigma[p]), ylab = expression(mu[p]),
     cex.lab = cex.val, col = c(rep("lightblue", 4), rep("black", 11), rep("lightblue",
     4)))
text(x = sig.A, y = mu.A, labels = "Asset A", pos = 4, cex = cex.val)
text(x = sig.B, y = mu.B, labels = "Asset B", pos = 4, cex = cex.val)

```

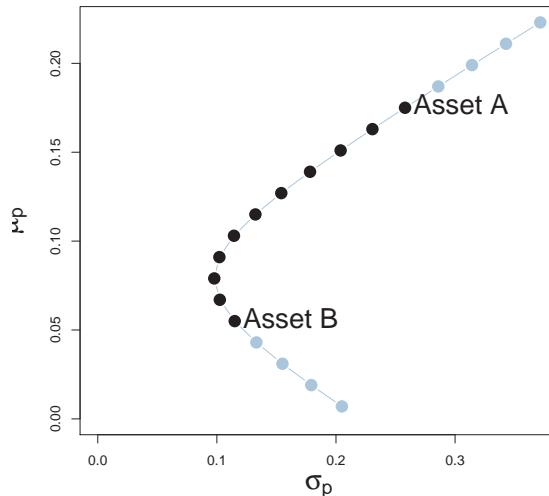


Fig. 13.1 Two-risky asset portfolio frontier. Long-only portfolios are in black (between “Asset B” and “Asset A”), and long-short portfolios are in blue (below “Asset B” and above “Asset A”).

The long only (no short sale) portfolios, shown in black dots between the points labeled “Asset B” and “Asset A”, are portfolios 5 to 15. The asset weights on these portfolios are:

```

longOnly = rbind(x.A[5:15], x.B[5:15])
rownames(longOnly) = assetNames
longOnly

##          p.5 p.6 p.7 p.8 p.9 p.10 p.11 p.12 p.13 p.14 p.15
## Asset A  0 0.1 0.2 0.3 0.4  0.5  0.6  0.7  0.8  0.9   1
## Asset B  1 0.9 0.8 0.7 0.6  0.5  0.4  0.3  0.2  0.1   0

```

The long-short portfolios, shown in blue dots below the point labeled “Asset B” and above the point labeled “Asset A”, are portfolios 1 to 4 and portfolios 16 to 19:

```

longShort = rbind(x.A[c(1:4, 16:19)], x.B[c(1:4, 16:19)])
rownames(longShort) = assetNames
longShort

##          p.1  p.2  p.3  p.4 p.16 p.17 p.18 p.19
## Asset A -0.4 -0.3 -0.2 -0.1  1.1  1.2  1.3  1.4
## Asset B  1.4  1.3  1.2  1.1 -0.1 -0.2 -0.3 -0.4

```

Not allowing short sales for asset A eliminates the inefficient portfolios 1 to 4 (blue dots below the label “Asset B”), and not allowing short sales for asset B eliminates the efficient portfolios 16 to 19 (blue dots above the label “Asset A”). We make the following remarks:

- Not allowing shorts sales in both assets limits the set of feasible portfolios.
- It is not possible to invest in a portfolio with a higher expected return than asset A.
- It is not possible to invest in a portfolio with a lower expected return than asset B.

Short sales constraints and the global minimum variance portfolio

Recall, the two risky asset global minimum variance portfolio solves the unconstrained optimization problem:

$$\begin{aligned} \min_{m_A, m_B} \sigma_p^2 &= m_A^2 \sigma_A^2 + m_B^2 \sigma_B^2 + 2m_A m_B \sigma_{AB} \\ s.t. \quad m_A + m_B &= 1. \end{aligned}$$

From Chapter 11, the two risky asset unconstrained global minimum variance portfolio weights are:

$$m_A = \frac{\sigma_B^2 - \sigma_{AB}}{\sigma_A^2 + \sigma_B^2 - 2\sigma_{AB}}, \quad m_B = 1 - m_A. \quad (13.1)$$

For the example data, the unconstrained global minimum variance portfolio weights are both positive:

```

library(IntroCompFinR)
gmin.port = globalMin.portfolio(mu.vals, Sigma)
gmin.port

## Call:
## globalMin.portfolio(er = mu.vals, cov.mat = Sigma)
##
## Portfolio expected return:    0.0793
## Portfolio standard deviation: 0.0978
## Portfolio weights:
## Asset A Asset B
##    0.202    0.798

```

This means that if we disallow short-sales in both assets we still get the same global minimum variance portfolio weights. In this case, we say that the no short sales constraints are *not binding* (have no effect) on the solution for the unconstrained optimization problem (13.1).

However, it is possible for one of the weights in the global minimum variance portfolio to be negative if correlation between the returns on assets A and B is sufficiently positive. To see this, the numerator in the formula for the global minimum variance weight for asset A can be re-written as:¹

$$\sigma_B^2 - \sigma_{AB} = \sigma_B^2 - \rho_{AB}\sigma_A\sigma_B = \sigma_B^2 \left(1 - \rho_{AB} \frac{\sigma_A}{\sigma_B}\right).$$

Hence, the weight in asset A in the global minimum variance portfolio will be negative if:

$$1 - \rho_{AB} \frac{\sigma_A}{\sigma_B} < 0 \Rightarrow \rho_{AB} > \frac{\sigma_B}{\sigma_A}.$$

For the example data, this cut-off value for the correlation is:

```
sig.B/sig.A
```

```
## [1] 0.446
```

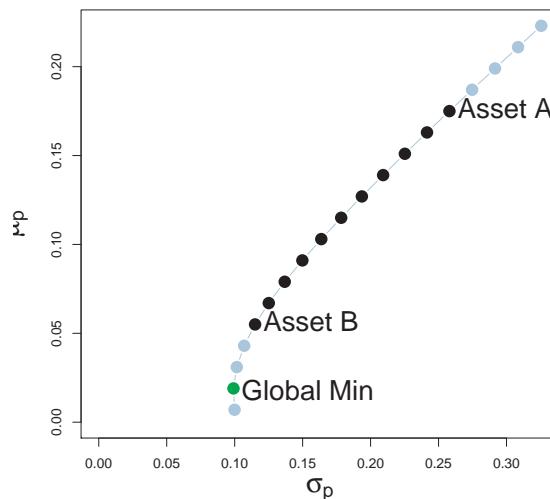


Fig. 13.2 Two risky asset portfolio frontier with short sale in asset A in the global minimum variance portfolio.

Figure 13.2 shows the two risky asset portfolio frontier computed with $\rho_{AB} = 0.8$. Here, the global minimum variance portfolio is:

```
rho.AB = 0.8
sig.AB = rho.AB * sig.A * sig.B
Sigma = matrix(c(sig2.A, sig.AB, sig.AB, sig2.B), 2, 2)
```

¹ By the Cauchy-Schwarz inequality the denominator for m_A in (13.1) is always positive.

```

gmin.port = globalMin.portfolio(mu.vals, Sigma)
gmin.port

## Call:
## globalMin.portfolio(er = mu.vals, cov.mat = Sigma)
##
## Portfolio expected return: 0.016
## Portfolio standard deviation: 0.099
## Portfolio weights:
## Asset A Asset B
## -0.325 1.325

```

and contains a negative weight in asset A. This portfolio is shown as the green dot in Figure 13.2, and plots below the point labeled “Asset B”. Here, the no-short sales constraint on asset A is a *binding constraint*: it is not possible to invest in the global minimum variance portfolio without shorting asset A. Visually, we can see that the feasible long-only portfolio with the smallest variance is 100% invested in asset B. This is the *no short sales constrained* global minimum variance portfolio. Notice that the variance of this portfolio is larger than the variance of the global minimum variance portfolio. This is the cost, in terms of variance, of imposing the no-short sales constraint.

13.1.2 One risky asset and a risk-free asset

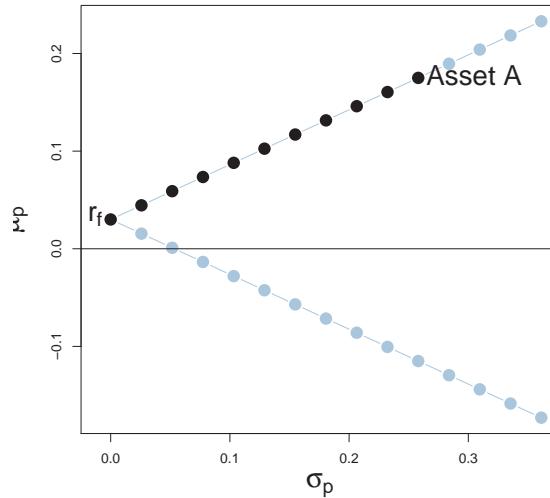


Fig. 13.3 Return-risk characteristics of portfolios of asset A and a risk-free asset with $r_f = 0.03$. Long-only portfolios are in black (between “ r_f ” and “Asset A”), and long-short portfolios are in blue (below “ r_f ” and above “Asset A”).

Now consider portfolios of a risky asset (e.g., asset A) and a risk-free asset (e.g., U.S. T-Bill) with risk-free rate $r_f > 0$. Let x denote the portfolio weight in the risky asset. Then $1 - x$ gives the weight in the risk-free asset. From Chapter 11, the expected return and volatility of portfolios of one risky asset and one risk-free asset are:

$$\mu_p = r_f + x(\mu - r_f), \\ \sigma_p = |x|\sigma,$$

where μ and σ are the expected return and volatility on the risky asset, respectively. For example, Figure 13.3 shows 29 portfolios of asset A and the risk-free asset with $r_f = 0.03$ created with:

```
r.f = 0.03
x.A = seq(from = -1.4, to = 1.4, by = 0.1)
mu.p.A = r.f + x.A * (mu.A - r.f)
sig.p.A = abs(x.A) * sig.A
plot(sig.p.A, mu.p.A, type = "b", ylim = c(min(mu.p.A), max(mu.p.A)), xlim = c(-0.01,
  max(sig.p.A)), pch = 16, cex = cex.val, xlab = expression(sigma[p]), ylab = expression(mu[p]),
  cex.lab = cex.val, col = c(rep("lightblue", 14), rep("black", 11), rep("lightblue",
  4)))
abline(h = 0)
text(x = sig.A, y = mu.A, labels = "Asset A", pos = 4, cex = cex.val)
text(x = 0, y = r.f, labels = expression(r[f]), pos = 2, cex = cex.val)
```

The portfolios (blue dots) above the point labeled “Asset A” are short the risk-free asset and long more than 100% of asset A. The portfolios (blue dots) below $r_f = 0.03$ are short the risky asset and long more than 100% in the risk-free asset.

Here, we assume that the short sales constraint only applies to the risky asset and not the risk-free asset. That is, we assume that we can borrow and lend at risk-free rate r_f without constraints. In this situation, the no short sales constraint on the risky asset eliminates the inefficient portfolios (blue dots) below the risk-free rate r_f .

13.1.3 Two risky assets and risk-free asset

As discussed in Chapter 11, in the case of two risky assets (with no short sales restrictions) plus a risk-free asset the set of efficient portfolios is a combination of the tangency portfolio (i.e., the maximum Sharpe ratio portfolio of two risky assets) and the risk-free asset. For the example data with $\rho_{AB} = -0.164$ and $r_f = 0.03$, the tangency portfolio is:

```
rho.AB = -0.164
sig.AB = rho.AB * sig.A * sig.B
Sigma = matrix(c(sig2.A, sig.AB, sig.AB, sig2.B), 2, 2)
r.f = 0.03
tan.port = tangency.portfolio(mu.vals, Sigma, r.f)
summary(tan.port, r.f)

## Call:
```

```
## tangency.portfolio(er = mu.vals, cov.mat = Sigma, risk.free = r.f)
##
## Portfolio expected return:      0.111
## Portfolio standard deviation: 0.125
## Portfolio Sharpe Ratio:       0.644
## Portfolio weights:
## Asset A Asset B
##   0.463   0.537
```

No asset is sold short in the unconstrained tangency portfolio, and so this is also the short-sales constrained tangency portfolio. The set of efficient portfolios is illustrated in Figure 13.4, created with:

```
# risky asset only portfolios
x.A = seq(from = 0, to = 1, by = 0.1)
x.B = 1 - x.A
mu.p = x.A * mu.A + x.B * mu.B
sig2.p = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p = sqrt(sig2.p)
# T-bills plus tangency
x.tan = seq(from = 0, to = 2.4, by = 0.1)
mu.p.tan.tbill = r.f + x.tan * (tan.port$er - r.f)
sig.p.tan.tbill = x.tan * tan.port$sd
# global minimum variance portfolio
gmin.port = globalMin.portfolio(mu.vals, Sigma)
# plot efficient portfolios
plot(sig.p, mu.p, type = "b", pch = 16, cex = cex.val, ylim = c(0, 0.2), xlim = c(-0.01,
  0.35), xlab = expression(sigma[p]), ylab = expression(mu[p]), cex.lab = cex.val)
text(x = sig.A, y = mu.A, labels = "Asset A", pos = 4, cex = cex.val)
text(x = sig.B, y = mu.B, labels = "Asset B", pos = 4, cex = cex.val)
text(x = 0, y = r.f, labels = expression(r[f]), pos = 2, cex = cex.val)
text(x = tan.port$sd, y = tan.port$er, labels = "Tangency", pos = 2, cex = cex.val)
text(gmin.port$sd, gmin.port$er, labels = "Global Min", pos = 4, cex = cex.val)
points(sig.p.tan.tbill, mu.p.tan.tbill, type = "l", col = "green", lwd = 2, cex = cex.val)
```

Figure 13.4 shows that when the tangency portfolio is located between the global minimum variance portfolio (tip of the Markowitz bullet) and asset A, there will be no short sales in the tangency portfolio. However, if the correlation between assets A and B is sufficiently positive (and the risk-free rate is adjusted so that it is greater than the expected return on the global minimum variance portfolio) then asset B can be sold short in the unconstrained tangency portfolio. For example, if $\rho_{AB} = 0.7$ and $r_f = 0.01$ the unconstrained tangency portfolio becomes:

```
rho.AB = 0.7
sig.AB = rho.AB * sig.A * sig.B
Sigma = matrix(c(sig2.A, sig.AB, sig.AB, sig2.B), 2, 2)
r.f = 0.01
tan.port = tangency.portfolio(mu.vals, Sigma, r.f)
summary(tan.port, r.f)

## Call:
## tangency.portfolio(er = mu.vals, cov.mat = Sigma, risk.free = r.f)
##
## Portfolio expected return:      0.238
```

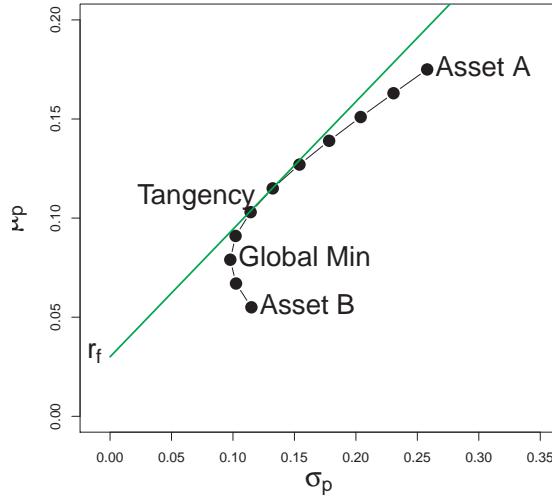


Fig. 13.4 Efficient portfolios of two risky assets plus a risk-free asset: $\rho_{AB} = -0.164$ and $r_f = 0.03$. No short sales in unconstrained tangency portfolio.

```
## Portfolio standard deviation: 0.355
## Portfolio Sharpe Ratio:      0.644
## Portfolio weights:
## Asset A Asset B
##    1.529  -0.529
```

Now, asset B is sold short in the tangency portfolio. The set of efficient portfolios in this case is illustrated in Figure 13.5. Notice that the tangency portfolio is located on the set of risky asset only portfolios above the point labeled “Asset A”, which indicates that asset B is sold short and more than 100% is invested in asset A.²

When shorts sales of risky assets are not allowed, the unconstrained tangency portfolio is infeasible. In this case, the constrained tangency portfolio is not the tangency portfolio but is 100% invested in asset A. This portfolio is located at the tangency point of a straight line drawn from the risk-free rate to the long-only risky asset frontier. Here, the cost of the no-short sales constraint is the reduction in the Sharpe ratio of the tangency portfolio. The Sharpe ratio of the constrained tangency portfolio is the Sharpe ratio of asset A:

```
(mu.A - r.f)/sig.A
## [1] 0.64
```

Hence, the cost of the no short sales constraint is very small.

² Also notice that when $\rho_{AB} = 0.7$ the expected return on the global minimum variance drops (frontier of risky assets is shifted down) below 0.03 so we need to reduce r_f to 0.01 to ensure that the tangency portfolio has a positive slope.

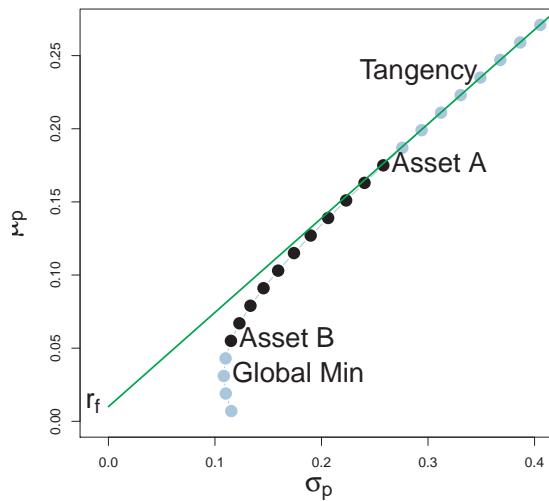


Fig. 13.5 Efficient portfolios of two risky assets plus a risk free asset: $\rho_{AB} = 0.7$ and $r_f = 0.01$. Asset B sold short in unconstrained tangency portfolio. The short sale constrained tangency portfolio is the point labeled “Asset A”.

13.2 Portfolio Theory with Short Sales Constraints in a General Setting

In this section we extend the analysis of the previous section to the more general setting of n risky assets plus a risk free asset.

13.2.1 Multiple risky assets

Example 13.1. Random portfolios of Microsoft, Nordstrom and Starbucks with no-short sales

The example data on Microsoft, Nordstrom and Starbucks from Chapter 12 is used to illustrate the computation of mean-variance efficient portfolios with short sales constraints. This data is created using:

```
asset.names <- c("MSFT", "NORD", "SBUX")
mu.vec = c(0.0427, 0.0015, 0.0285)
names(mu.vec) = asset.names
sigma.mat = matrix(c(0.01, 0.0018, 0.0011, 0.0018, 0.0109, 0.0026, 0.0011, 0.0026,
  0.0199), nrow = 3, ncol = 3)
dimnames(sigma.mat) = list(asset.names, asset.names)
sd.vec = sqrt(diag(sigma.mat))
r.f = 0.005
```

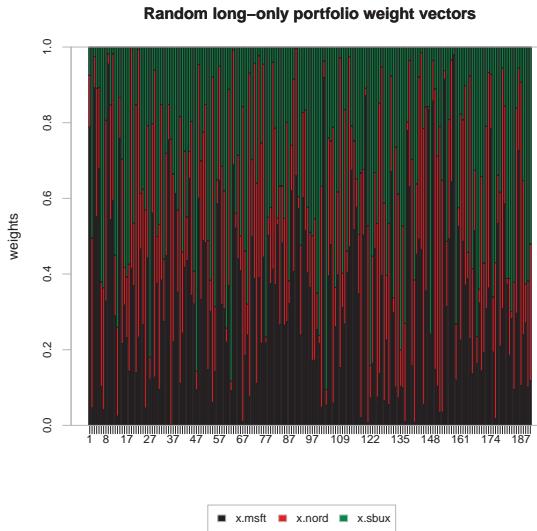


Fig. 13.6 Weights on 191 random long-only portfolios of Microsoft, Nordstrom and Starbucks.

The impact of the no short sales restrictions on the risk-return characteristics of feasible portfolios is nicely illustrated by constructing random long-only portfolios of these three assets:

```
set.seed(123)
x.msft = runif(400, min = 0, max = 1)
x.nord = runif(400, min = 0, max = 1)
x.sbx = 1 - x.msft - x.nord
long.only = which(x.sbx > 0)
x.msft = x.msft[long.only]
x.nord = x.nord[long.only]
x.sbx = x.sbx[long.only]
length(long.only)

## [1] 191
```

Here we first create 400 random weights for Microsoft and Nordstrom that lie between zero and one, respectively. Then we determine the weight on Starbucks and throw out portfolios for which the weight on Starbucks is negative. The remaining 191 portfolios are then long-only portfolios. To be sure, these weights are illustrated in Figure 13.6 created using:

```
library(PerformanceAnalytics)
chart.StackedBar(cbind(x.msft, x.nord, x.sbx), main = "Random long-only portfolio weight vectors",
                 xlab = "portfolio", ylab = "weights", xaxis.labels = as.character(1:length(long.only)))
```

The risk-return characteristics of the random long-only portfolios are illustrated in Figure 13.7 created using:

```

plot(sd.vec, mu.vec, ylim = c(0, 0.05), xlim = c(0, 0.2), ylab = expression(mu[p]),
      xlab = expression(sigma[p]), type = "n")
for (i in 1:length(long.only)) {
  z.vec = c(x.msft[i], x.nord[i], x.sbux[i])
  mu.p = crossprod(z.vec, mu.vec)
  sig.p = sqrt(t(z.vec) %*% sigma.mat %*% z.vec)
  points(sig.p, mu.p, pch = 16, col = "grey", cex = 1.5)
}
points(sd.vec, mu.vec, pch = 16, col = "black", cex = 2.5, cex.lab = 1.75)
text(sd.vec, mu.vec, labels = asset.names, pos = 4, cex = 2)

```

The shape of the risk-return scatter is revealing. No random long-only portfolio has an expected return higher than the expected return on Microsoft, which is the asset with the highest expected return. Similarly, no random portfolio has an expected return lower than the expected return on Nordstrom - which is the asset with the lowest expected return. Finally, no random portfolio has a volatility higher than the volatility on Starbucks, the asset with the highest volatility. The scatter of points has a convex (to the origin) parabolic-shaped outer boundary with endpoints at Microsoft and Nordstrom, respectively. The points inside the boundary taper outward from Starbucks towards the outer boundary. The entire risk-return scatter resembles an umbrella tipped on its side with spires extending from the outer boundary to the individual assets.

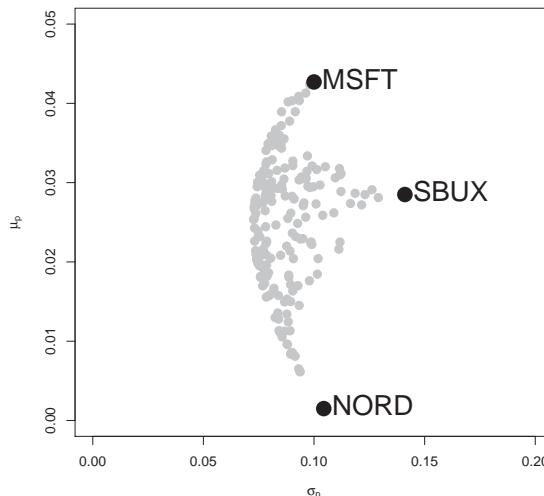


Fig. 13.7 Risk-return characteristics of 191 random long-only portfolios of Microsoft, Nordstrom and Starbucks.

13.2.2 Portfolio optimization imposing no-short sales constraints

Consider an investment universe with n risky assets whose simple returns are described by the CER model with expected return vector μ and covariance matrix Σ . The optimization problem to solve for the global minimum variance portfolio \mathbf{m} with no-short sales constraints on all assets is a straightforward extension of the problem that allows short sales. We simply add the n no-short sales inequality constraints $m_i \geq 0 (i = 1, \dots, n)$. The optimization problem is:

$$\begin{aligned} \min_{\mathbf{m}} \sigma_{p,m}^2 &= \mathbf{m}'\Sigma\mathbf{m} \quad \text{s.t.} \\ \mathbf{m}'\mathbf{1} &= 1, \\ m_i &\geq 0 \quad (i = 1, \dots, n). \end{aligned} \tag{13.2}$$

Here, there is one equality constraint, $\mathbf{m}'\mathbf{1} = 1$, and n inequality constraints, $m_i \geq 0 (i = 1, \dots, n)$.

Similarly, the optimization problem to solve for a minimum variance portfolio \mathbf{x} with target expected return $\mu_{p,0}$ and no short sales allowed for any asset becomes

$$\begin{aligned} \min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\Sigma\mathbf{x} \quad \text{s.t.} \\ \mu_{p,x} &= \mathbf{x}'\mu = \mu_p^0, \\ \mathbf{x}'\mathbf{1} &= 1, \\ x_i &\geq 0 \quad (i = 1, \dots, n). \end{aligned} \tag{13.3}$$

Here there are two equality constraints, $\mathbf{x}'\mu = \mu_p^0$ and $\mathbf{x}'\mathbf{1} = 1$, and n inequality constraints, $x_i \geq 0 (i = 1, \dots, n)$.

Remarks:

1. The optimization problems (13.2) and (13.3), unfortunately, cannot be analytically solved using the method of Lagrange multipliers due to the inequality constraints. They must be solved numerically using an optimization algorithm. Fortunately, the optimization problems (13.2) and (13.3) are special cases of a more general quadratic programming problem for which there is a specialized optimization algorithm that is fast and efficient and available in R in the package **quadprog**.³
2. There may not be a feasible solution to (13.3). That is, there may not exist a no-short sale portfolio \mathbf{x} that reaches the target return μ_p^0 . For example, in Figure 13.7 there is no efficient portfolio that has target expected return higher than the expected return on Microsoft. In general, with n assets there is no feasible solution to (13.3) for target expected returns higher than the maximum expected return among the n assets. If you try to solve (13.3) for an infeasible target expected return, the numerical optimizer will return an error message that indicates no solution is found.

³ See the CRAN Optimization Task View for additional R packages that can be used to solve QP problems.

3. The portfolio variances associated with the solutions to (13.2) and (13.3), respectively, must be as least as large as the variances associated with the solutions that do not impose the short sales restrictions. Intuitively this makes sense. Adding additional constraints to a minimization problem will lead to a higher minimum at the solution if the additional constraints are binding. If the constraints are not binding, then the solution with and without the additional constraints are the same.
4. The entire frontier of minimum variance portfolios without short sales exists only for target expected returns between the expected return on the global minimum variance portfolio with no short sales that solves (13.2) and the target expected return equal to the maximum expected return among the n assets. For these target returns, the no-short sales frontier will lie inside and to the right of the frontier that allows short sales if the no-short sales restrictions are binding for some asset.
5. The entire frontier of minimum variance portfolios without short sales can no longer be constructed from any two frontier portfolios. It has to be computed by brute force for each portfolio solving (13.3) with target expected return μ_p^0 above the expected return on the the global minimum variance portfolio that solves (13.2) and below the target return equal to the maximum expected return among the n assets.

13.2.3 Quadratic Programming Problems

In this sub-section, we show that the inequality constrained portfolio optimization problems (13.2) and (13.3) are special cases of more general quadratic programming problems and we show how to use the function `solve.QP()` from the R package `quadprog` to numerically solve these problems.

Quadratic programming (QP) problems are of the form:

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}' \mathbf{D} \mathbf{x} - \mathbf{d}' \mathbf{x}, \quad (13.4)$$

$$\mathbf{A}'_{eq} \mathbf{x} \geq \mathbf{b}_{eq} \text{ for } l \text{ equality constraints,} \quad (13.5)$$

$$\mathbf{A}'_{neq} \mathbf{x} = \mathbf{b}_{neq} \text{ for } m \text{ inequality constraints,} \quad (13.6)$$

where \mathbf{D} is an $n \times n$ matrix, \mathbf{x} and \mathbf{d} are $n \times 1$ vectors, \mathbf{A}'_{neq} is an $m \times n$ matrix, \mathbf{b}_{neq} is an $m \times 1$ vector, \mathbf{A}'_{eq} is an $l \times n$ matrix, and \mathbf{b}_{eq} is an $l \times 1$ vector.

The `solve.QP()` function from the R package `quadprog` can be used to solve QP functions of the form (13.4) - (13.6). The function `solve.QP()` takes as input the matrices and vectors \mathbf{D} , \mathbf{d} , \mathbf{A}_{eq} , \mathbf{b}_{eq} , \mathbf{A}_{neq} and \mathbf{b}_{neq} :

```
library(quadprog)
args(solve.QP)

## function (Dmat, dvec, Amat, bvec, meq = 0, factorized = FALSE)
## NULL
```

where the arguments `Dmat` and `dvec` correspond to the matrix \mathbf{D} and the vector \mathbf{d} , respectively. The function `solve.QP()`, however, assumes that the equality and inequality matrices (\mathbf{A}_{eq} , \mathbf{A}_{neq}) and vectors (\mathbf{b}_{eq} , \mathbf{b}_{neq}) are combined into a single $(l+m) \times n$ matrix \mathbf{A}' and a single $(l+m) \times 1$ vector \mathbf{b} of the form:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A}'_{eq} \\ \mathbf{A}'_{neq} \end{bmatrix}, \mathbf{b} = \begin{pmatrix} \mathbf{b}_{eq} \\ \mathbf{b}_{neq} \end{pmatrix}.$$

In `solve.QP()`, the argument `Amat` represents the matrix \mathbf{A} (not \mathbf{A}') and the argument `bmat` represents the vector \mathbf{b} . The argument `meq` determines the number of linear equality constraints (i.e., the number of rows l of \mathbf{A}'_{eq}) so that \mathbf{A}' can be separated into \mathbf{A}'_{eq} and \mathbf{A}'_{neq} , respectively.

The portfolio optimization problems (13.2) and (13.3) are special cases of the general QP problem (13.4) - (13.6). Consider first the problem to find the global minimum variance portfolio (13.2). The objective function $\sigma_{p,m}^2 = \mathbf{m}'\Sigma\mathbf{m}$ can be recovered from (13.4) by setting $\mathbf{x} = \mathbf{m}$, $\mathbf{D} = 2 \times \Sigma$ and $\mathbf{d} = (0, \dots, 0)'$. Then

$$\frac{1}{2}\mathbf{x}'\mathbf{D}\mathbf{x} - \mathbf{d}'\mathbf{x} = \mathbf{m}'\Sigma\mathbf{m} = \sigma_{p,m}^2.$$

Here, there is one equality constraint, $\mathbf{m}'\mathbf{1} = 1$, and n inequality constraints, $m_i \geq 0$ ($i = 1, \dots, n$), which can be expressed compactly as $\mathbf{m} \geq \mathbf{0}$. These constraints can be expressed in the form (13.5) and (13.6) by defining the restriction matrices:

$$\begin{aligned} \mathbf{A}'_{eq} &= \mathbf{1}', & \mathbf{b}_{eq} &= 1 \\ &(1 \times n) && (1 \times 1) \\ \mathbf{A}'_{neq} &= \mathbf{I}_n, & \mathbf{b}_{neq} &= (0, \dots, 0)' = \mathbf{0}, \\ &(n \times n) && (n \times 1) \end{aligned}$$

so that

$$\mathbf{A}' = \begin{bmatrix} \mathbf{1}' \\ \mathbf{I}_n \end{bmatrix}, \mathbf{b} = \begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}.$$

Here,

$$\begin{aligned} \mathbf{A}'_{eq}\mathbf{m} &= \mathbf{1}'\mathbf{m} = 1 = b_{eq}, \\ \mathbf{A}'_{neq}\mathbf{m} &= \mathbf{I}_n\mathbf{m} = \mathbf{m} \geq \mathbf{0} = \mathbf{b}_{neq}. \end{aligned}$$

Example 13.2. Global minimum variance portfolio with no short sales for three asset example data

The unconstrained global minimum variance portfolio of Microsoft, Nordstrom and Starbucks is:

```
library(IntroCompFinR)
gmin.port = globalMin.portfolio(mu.vec, sigma.mat)
gmin.port
```

```
## Call:
## globalMin.portfolio(er = mu.vec, cov.mat = sigma.mat)
##
## Portfolio expected return: 0.0249
## Portfolio standard deviation: 0.0727
## Portfolio weights:
## MSFT NORD SBUX
## 0.441 0.366 0.193
```

This portfolio does not have any negative weights and so the no-short sales restriction is not binding. Hence, the short sales constrained global minimum variance is the same as the unconstrained global minimum variance portfolio.

The restriction matrices and vectors required by `solve.QP()` to compute the short sales constrained global minimum variance portfolio are:

```
D.mat = 2 * sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(rep(1, 3), diag(3))
b.vec = c(1, rep(0, 3))
D.mat

##          MSFT    NORD    SBUX
## MSFT 0.0200 0.0036 0.0022
## NORD 0.0036 0.0218 0.0052
## SBUX 0.0022 0.0052 0.0398

d.vec

## [1] 0 0 0

t(A.mat)

##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    0    0
## [3,]    0    1    0
## [4,]    0    0    1

b.vec

## [1] 1 0 0 0
```

To find the short sales constrained global minimum variance portfolio call `solve.QP()` with the above inputs and set `meq=1` to indicate one equality constraint:

```
qp.out = solve.QP(Dmat = D.mat, dvec = d.vec, Amat = A.mat, bvec = b.vec, meq = 1)
class(qp.out)

## [1] "list"

names(qp.out)

## [1] "solution"           "value"
```

```
## [3] "unconstrained.solution" "iterations"
## [5] "Lagrangian"           "iact"
```

The returned object, `qp.out`, is a list with the following components:

```
qp.out

## $solution
## [1] 0.441 0.366 0.193
##
## $value
## [1] 0.00528
##
## $unconstrained.solution
## [1] 0 0 0
##
## $iterations
## [1] 2 0
##
## $Lagrangian
## [1] 0.0106 0.0000 0.0000 0.0000
##
## $iact
## [1] 1
```

The portfolio weights are in the `solution` component, which match the global minimum variance weights allowing short sales, and satisfy the required constraints (weights sum to one and all weights are positive). The minimized value of the objective function (portfolio variance) is in the `value` component. See the help file for `solve.QP()` for explanations of the other components.

You can also use the **IntroCompFinR** function `globalmin.portfolio()` to compute the global minimum variance portfolio subject to short-sales constraints by specifying the optional argument `shorts=FALSE`:

```
gmin.port = globalMin.portfolio(mu.vec, sigma.mat, shorts = FALSE)
gmin.port

## Call:
## globalMin.portfolio(er = mu.vec, cov.mat = sigma.mat, shorts = FALSE)
##
## Portfolio expected return:      0.0249
## Portfolio standard deviation: 0.0727
## Portfolio weights:
## MSFT  NORD  SBUX
## 0.441 0.366 0.193
```

When `shorts=FALSE`, `globalmin.portfolio()` uses `solve.QP()` as described above to do the optimization. ■

Next, consider the problem (13.3) to find a minimum variance portfolio with a given target expected return. The objective function (13.4) has $\mathbf{D} = 2 \times \Sigma$ and $\mathbf{d} = (0, \dots, 0)'$. The two

linear equality constraints, $\mathbf{x}'\boldsymbol{\mu} = \mu_p^0$ and $\mathbf{x}'\mathbf{1} = 1$, and n inequality constraints $\mathbf{x} \geq \mathbf{0}$ have restriction matrices and vectors

$$\mathbf{A}'_{eq} = \begin{pmatrix} \boldsymbol{\mu}' \\ \mathbf{1}' \end{pmatrix}_{(2 \times n)}, \quad \mathbf{b}_{eq} = \begin{pmatrix} \mu_p^0 \\ 1 \end{pmatrix},$$

$$\mathbf{A}'_{neq} = \mathbf{I}_n, \quad \mathbf{b}_{neq} = (0, \dots, 0)'.$$

so that

$$\mathbf{A}' = \begin{pmatrix} \boldsymbol{\mu}' \\ \mathbf{1}' \\ \mathbf{I}_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mu_p^0 \\ 1 \\ \mathbf{0} \end{pmatrix}.$$

Here,

$$\mathbf{A}'_{eq}\mathbf{x} = \begin{pmatrix} \boldsymbol{\mu}' \\ \mathbf{1}' \end{pmatrix}\mathbf{x} = \begin{pmatrix} \boldsymbol{\mu}'\mathbf{x} \\ \mathbf{1}'\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mu_p^0 \\ 1 \end{pmatrix} = \mathbf{b}_{eq},$$

$$\mathbf{A}'_{neq}\mathbf{x} = \mathbf{I}_n\mathbf{x} = \mathbf{x} \geq 0.$$

Example 13.3. Minimum variance portfolio with target expected return and no short sales for three asset example data

Now, we consider finding the minimum variance portfolio that has the same mean as Microsoft but does not allow short sales. First, we find the minimum variance portfolio allowing for short sales using the **IntroCompFinR** function **efficient.portfolio()**:

```
eMsft.port = efficient.portfolio(mu.vec, sigma.mat, target.return = mu.vec["MSFT"])
eMsft.port

## Call:
## efficient.portfolio(er = mu.vec, cov.mat = sigma.mat, target.return = mu.vec["MSFT"])
##
## Portfolio expected return: 0.0427
## Portfolio standard deviation: 0.0917
## Portfolio weights:
##   MSFT      NORD      SBUX
## 0.8275 -0.0907  0.2633
```

This portfolio has a short sale (negative weight) in Nordstrom. Hence, when the short sales restriction is imposed it will be binding. Next, we set up the restriction matrices and vectors required by **solve.QP()** to compute the minimum variance portfolio with no-short sales:

```
D.mat = 2 * sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(mu.vec, rep(1, 3), diag(3))
b.vec = c(mu.vec["MSFT"], 1, rep(0, 3))
t(A.mat)

##           MSFT      NORD      SBUX
```

```

## mu.vec 0.0427 0.0015 0.0285
##      1.0000 1.0000 1.0000
##      1.0000 0.0000 0.0000
##      0.0000 1.0000 0.0000
##      0.0000 0.0000 1.0000

b.vec

## MSFT
## 0.0427 1.0000 0.0000 0.0000 0.0000

```

Then we call `solve.QP()` with `meq=2` to indicate two equality constraints:

```

qp.out = solve.QP(Dmat = D.mat, dvec = d.vec, Amat = A.mat, bvec = b.vec, meq = 2)
names(qp.out$solution) = names(mu.vec)
round(qp.out$solution, digits = 3)

## MSFT NORD SBUX
##   1   0   0

```

With short sales not allowed, the minimum variance portfolio with the same mean as Microsoft is 100% invested in Microsoft. This is consistent with what we see in Figure 13.7. The volatility of this portfolio is slightly higher than the volatility of the portfolio that allows short sales:

```

sqrt(qp.out$value)

## [1] 0.1

eMsft.port$sd

## [1] 0.0917

```

You can also use the **IntroCompFinR** function `efficient.portfolio()` to compute a minimum variance portfolio with target expected return subject to short-sales constraints by specifying the optional argument `shorts=FALSE`:

```

efficient.portfolio(mu.vec, sigma.mat, target.return = mu.vec["MSFT"], shorts = FALSE)

## Call:
## efficient.portfolio(er = mu.vec, cov.mat = sigma.mat, target.return = mu.vec["MSFT"],
##   shorts = FALSE)
##
## Portfolio expected return: 0.0427
## Portfolio standard deviation: 0.1
## Portfolio weights:
## MSFT NORD SBUX
##   1   0   0

```

Suppose you try to find a minimum variance portfolio with target return higher than the mean of Microsoft while imposing no short sales:

```
efficient.portfolio(mu.vec, sigma.mat, target.return = mu.vec["MSFT"] + 0.01, shorts = FALSE)

## Error in quadprog::solve.QP(Dmat = Dmat, dvec = dvec, Amat = Amat, bvec = bvec, :
## constraints are inconsistent, no solution!
```

Here, `solve.QP()` reports the error “constraints are inconsistent, no solution!” This agrees with what we see in Figure 13.7.

■

Example 13.4. Compute efficient frontier portfolios from three asset example data

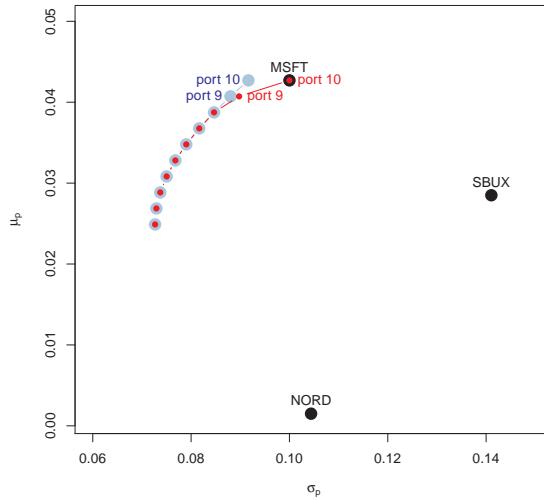


Fig. 13.8 Efficient frontier portfolios with and without short sales. The unconstrained efficient frontier portfolios are in blue, and the short sales constrained efficient portfolios are in red. The unconstrained portfolios labeled “port 9” and “port 10” have short sales in Nordstrom. The constrained portfolio labeled “port 9” has zero weight in Nordstrom, and the constrained portfolio labeled “port 10” has zero weights in Nordstrom and Starbucks.

In this example we compare the efficient frontier portfolios computed with and without short sales. This can be easily done using the **IntroCompFinR** function `efficient.frontier()`. First, we compute the efficient frontier allowing short sales for target returns between the mean of the global minimum variance portfolio and the mean of Microsoft:

```
ef = efficient.frontier(mu.vec, sigma.mat, alpha.min = 0, alpha.max = 1, nport = 10)
ef$weights

##          MSFT      NORD      SBUX
## port 1  0.441  0.3656  0.193
## port 2  0.484  0.3149  0.201
## port 3  0.527  0.2642  0.209
## port 4  0.570  0.2135  0.217
```

```
## port 5  0.613  0.1628 0.224
## port 6  0.656  0.1121 0.232
## port 7  0.699  0.0614 0.240
## port 8  0.742  0.0107 0.248
## port 9  0.785 -0.0400 0.256
## port 10 0.827 -0.0907 0.263
```

Here, portfolio 1 is the global minimum variance portfolio and portfolio 10 is the efficient portfolio with the same mean as Microsoft. Notice that portfolios 9 and 10 have negative weights in Nordstrom. Next, we compute the efficient frontier not allowing short sales for the same range of target returns:

```
ef.ns = efficient.frontier(mu.vec, sigma.mat, alpha.min = 0, alpha.max = 1, nport = 10,
                           shorts = FALSE)
ef.ns$weights

##          MSFT      NORD     SBUX
## port 1  0.441  0.3656 0.193
## port 2  0.484  0.3149 0.201
## port 3  0.527  0.2642 0.209
## port 4  0.570  0.2135 0.217
## port 5  0.613  0.1628 0.224
## port 6  0.656  0.1121 0.232
## port 7  0.699  0.0614 0.240
## port 8  0.742  0.0107 0.248
## port 9  0.861  0.0000 0.139
## port 10 1.000  0.0000 0.000
```

Portfolios 1 - 8 have all positive weights and are the same as portfolios 1 - 8 when short sales are allowed. However, for portfolios 9 and 10 the short sales constraint is binding. For portfolio 9, the weight in Nordstrom is forced to zero and the weight in Starbucks is reduced. For portfolio 10, the weights on Nordstrom and Starbucks are forced to zero. The two frontiers are illustrated in Figure 13.8. For portfolios 1-8, the two frontiers coincide. For portfolios 9 and 10, the no-shorts frontier lies inside and to the right of the short-sales frontier. The cost of the short sales constraint is the increase the volatility of minimum variance portfolios for target returns near the expected return on Microsoft.

13.2.4 Tangency portfolio with short sales restrictions

Consider an investment universe with n risky assets and a single risk-free asset. We assume that short sales constraints only apply to the n risky assets, and that investors can borrow and lend at the risk-free rate r_f . As shown in Chapter 12, with short sales allowed, mean-variance efficient portfolios are combinations of the risk-free asset and the tangency portfolio. The tangency portfolio is the maximum Sharpe ratio portfolio of risky assets and is the solution to the maximization problem:

$$\max_t \frac{\mu_t - r_f}{\sigma_t} = \frac{\mathbf{t}'\mu - r_f}{(\mathbf{t}'\Sigma\mathbf{t})^{1/2}} \text{ s.t.} \\ \mathbf{t}'\mathbf{1} = 1.$$

Example 13.5. Compute efficient portfolios allowing short sales from three asset example data when there is a risk-free asset.

Assuming a risk-free rate $r_f = 0.005$, we can compute the unconstrained tangency portfolio for the three asset example data using the **IntroCompFinR** function `tangency.portfolio()`:

```
r.f = 0.005
tan.port = tangency.portfolio(mu.vec, sigma.mat, r.f)
summary(tan.port, r.f)

## Call:
## tangency.portfolio(er = mu.vec, cov.mat = sigma.mat, risk.free = r.f)
##
## Portfolio expected return: 0.0519
## Portfolio standard deviation: 0.112
## Portfolio Sharpe Ratio: 0.42
## Portfolio weights:
##   MSFT   NORD   SBUX
## 1.027 -0.326  0.299
```

Here, the unconstrained tangency portfolio has a negative weight in Nordstrom so that the short sales constraint on the risky assets will be binding. The unconstrained set of efficient portfolios that are combinations of the risk-free asset and the unconstrained tangency portfolio are illustrated in Figure 13.9.

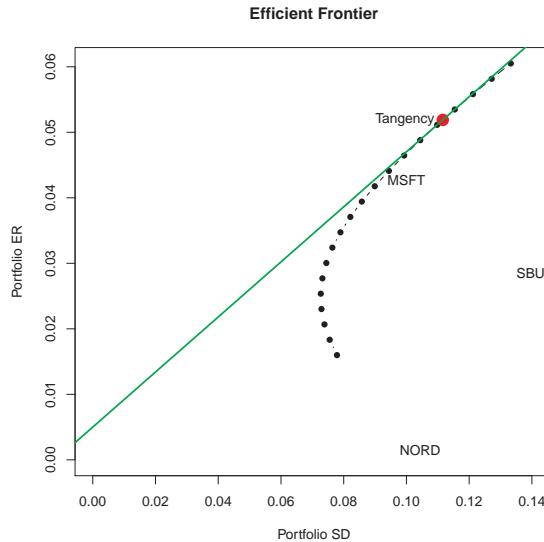


Fig. 13.9 Efficient portfolios of three risky assets and a risk-free asset allowing short sales. Nordstrom is sold short in the unconstrained tangency portfolio.

Here, we see that the unconstrained tangency portfolio is located on the frontier of risky asset portfolios above the point labeled “MSFT”. The green line represents portfolios of the risk-free asset and the unconstrained tangency portfolio.

■

Under short sales constraints on the risky assets, the maximum Sharpe ratio portfolio solves:

$$\begin{aligned} \max_{\mathbf{t}} \frac{\mu_t - r_f}{\sigma_t} &= \frac{\mathbf{t}'\mu - r_f}{(\mathbf{t}'\Sigma\mathbf{t})^{1/2}} \text{ s.t.} \\ \mathbf{t}'\mathbf{1} &= 1, \\ t_i &\geq 0, i = 1, \dots, n. \end{aligned}$$

This optimization problem cannot be written as a QP optimization as expressed in (13.4) - (13.6). Hence, it looks like we cannot use the function `solve.QP()` to find the short sales constrained tangency portfolio. It turns out we can use `solve.QP()` to find the short sales constrained tangency portfolio. To do this we utilize the alternative derivation of the tangency portfolio presented in Chapter 12. The alternative way to compute the tangency portfolio is to first find a minimum variance portfolio of risky assets and a risk-free asset whose excess expected return, $\tilde{\mu}_{p,x} = \mu'x - r_f\mathbf{1}$, achieves a target excess return $\tilde{\mu}_{p,0} = \mu_{p,0} - r_f > 0$. This portfolio solves the minimization problem:

$$\min_{\mathbf{x}} \sigma_{p,x}^2 = \mathbf{x}'\Sigma\mathbf{x} \text{ s.t. } \tilde{\mu}_{p,x} = \tilde{\mu}_{p,0},$$

where the weight vector \mathbf{x} is not constrained to satisfy $\mathbf{x}'\mathbf{1} = \mathbf{1}$. The tangency portfolio is then determined by normalizing the weight vector \mathbf{x} so that its elements sum to one:⁴

$$\mathbf{t} = \frac{\mathbf{x}}{\mathbf{x}'\mathbf{1}} = \frac{\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}{\mathbf{1}'\Sigma^{-1}(\mu - r_f \cdot \mathbf{1})}.$$

An interesting feature of this result is that it does not depend on the value of the target excess return value $\tilde{\mu}_{p,0} = \mu_{p,0} - r_f > 0$. That is, every value of $\tilde{\mu}_{p,0} = \mu_{p,0} - r_f > 0$ gives the same value for the tangency portfolio.⁵ We can utilize this alternative derivation of the tangency portfolio to find the tangency portfolio subject to short-sales restrictions on risky assets. First we solve the short-sales constrained minimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \sigma_{p,x}^2 &= \mathbf{x}'\Sigma\mathbf{x} \text{ s.t.} \\ \tilde{\mu}_{p,x} &= \tilde{\mu}_{p,0}, \\ x_i &\geq 0, i = 1, \dots, n. \end{aligned}$$

Here we can use any value for $\tilde{\mu}_{p,0}$ so for convenience we use $\tilde{\mu}_{p,0} = 1$. This is a QP problem with $\mathbf{D} = 2 \times \Sigma$ and $\mathbf{d} = (0, \dots, 0)'$, one linear equality constraint, $\tilde{\mu}_{p,x} = \tilde{\mu}_{p,0}$, and n inequality constraints $\mathbf{x} \geq \mathbf{0}$. The restriction matrices and vectors are:

$$\begin{aligned} \mathbf{A}'_{eq} &= (\mu - r_f\mathbf{1})', \quad \mathbf{b}_{eq} = 1, \\ &\quad (1 \times n) \quad (1 \times 1) \\ \mathbf{A}'_{neq} &= \mathbf{I}_n, \quad \mathbf{b}_{neq} = (0, \dots, 0)', \\ &\quad (n \times n) \quad (n \times 1) \end{aligned}$$

⁴ This normalization ensures that the tangency portfolio is 100% invested in the risky assets.

⁵ If $\tilde{\mu}_{p,0} = 0$ then the portfolio of risky assets and the risk-free asset has expected return equal to r_f . In this case, $x = 0$ and $x_f = 1$.

so that

$$\mathbf{A}' = \begin{pmatrix} (\mu - r_f \mathbf{1})' \\ \mathbf{I}_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix}.$$

Here,

$$\begin{aligned}\mathbf{A}'_{eq} \mathbf{x} &= (\mu - r_f \mathbf{1})' \mathbf{x} = \tilde{\mu}_{\mathbf{p}, \mathbf{x}} = 1, \\ \mathbf{A}'_{neq} \mathbf{x} &= \mathbf{I}_n \mathbf{x} = \mathbf{x} \geq 0.\end{aligned}$$

After solving the QP problem for \mathbf{x} , we then determine the short-sales constrained tangency portfolio by normalizing the weight vector \mathbf{x} so that its elements sum to one:

$$\mathbf{t} = \frac{\mathbf{x}}{\mathbf{x}' \mathbf{1}}.$$

Example 13.6. Compute efficient portfolios no allowing short sales from three asset example data when there is a risk-free asset.

First, we use `solve.QP()` to find the short sales restricted tangency portfolio. The restriction matrices for the short sales constrained optimization () - () are:

```
D.mat = 2 * sigma.mat
d.vec = rep(0, 3)
A.mat = cbind(mu.vec - rep(1, 3) * r.f, diag(3))
b.vec = c(1, rep(0, 3))
```

The un-normalized portfolio \mathbf{x} is found using:

```
qp.out = solve.QP(Dmat = D.mat, dvec = d.vec, Amat = A.mat, bvec = b.vec, meq = 1)
x.ns = qp.out$solution
names(x.ns) = names(mu.vec)
round(x.ns, digits = 3)

## MSFT NORD SBUX
## 22.74 0.00 6.08
```

The short sales constrained tangency portfolio is then:

```
t.ns = x.ns/sum(x.ns)
round(t.ns, digits = 3)

## MSFT NORD SBUX
## 0.789 0.000 0.211
```

In this portfolio, the allocation to Nordstrom, which was negative in the unconstrained tangency portfolio, is set to zero.

To verify that the derivation of the short sales constrained tangency portfolio does not depend on $\tilde{\mu}_{p,0} = \mu_{p,0} - r_f > 0$, we repeat the calculations with $\tilde{\mu}_{p,0} = 0.5$ instead of $\tilde{\mu}_{p,0} = 1$:

```
b.vec = c(0.5, rep(0, 3))
qp.out = solve.QP(Dmat = D.mat, dvec = d.vec, Amat = A.mat, bvec = b.vec, meq = 1)
x.ns = qp.out$solution
names(x.ns) = names(mu.vec)
t.ns = x.ns/sum(x.ns)
round(t.ns, digits = 3)

## MSFT NORD SBUX
## 0.789 0.000 0.211
```

You can compute the short sales restricted tangency portfolio using the **IntroCompFinR** function `tangency.portfolio()` with the optional argument `shorts=FALSE`:

```
tan.port.ns = tangency.portfolio(mu.vec, sigma.mat, r.f, shorts = FALSE)
summary(tan.port.ns, r.f)

## Call:
## tangency.portfolio(er = mu.vec, cov.mat = sigma.mat, risk.free = r.f,
##   shorts = FALSE)
##
## Portfolio expected return: 0.0397
## Portfolio standard deviation: 0.0865
## Portfolio Sharpe Ratio: 0.401
## Portfolio weights:
## MSFT NORD SBUX
## 0.789 0.000 0.211
```

Notice that the Sharpe ratio on the short sales restricted tangency portfolio is slightly smaller than the Sharpe ratio on the unrestricted tangency portfolio.

The set of efficient portfolios are combinations of the risk-free asset and the short sales restricted tangency portfolio. These portfolios are illustrated in Figure 13.10, created using:

```
ef.ns = efficient.frontier(mu.vec, sigma.mat, alpha.min = 0, alpha.max = 1, nport = 10,
  shorts = FALSE)
plot(ef.ns, plot.assets = TRUE, pch = 16)
points(tan.port.ns$sd, tan.port.ns$er, col = "red", pch = 16, cex = 2)
sr.tan.ns = (tan.port.ns$er - r.f)/tan.port.ns$sd
abline(a = r.f, b = sr.tan.ns, col = "green", lwd = 2)
text(tan.port$sd, tan.port$er, labels = "Tangency", pos = 2)
```

13.3 Further Reading

- Mention other quadratic programming packages: ROI etc. See Doug's book
- Mention Pfaff's R book
- Mention Grilli's book etc.

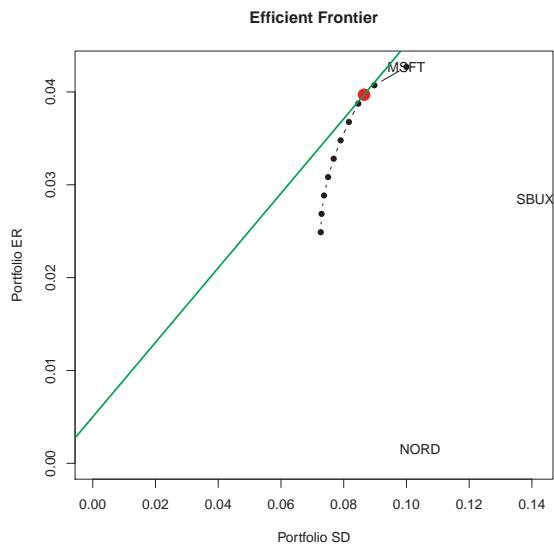


Fig. 13.10 Efficient

13.4 Problems

References

1. Ruppert, D.A.. *Statistics and Data Analysis for Financial Engineering*, Springer-Verlag.
2. Martin, R.D., Scherer, B., and Yollin, G. (2016).

Chapter 14

Portfolio Risk Budgeting

Updated: October 21, 2015
Copyright © Eric Zivot 2015

The idea of portfolio risk budgeting is to decompose a measure of portfolio risk into risk contributions from the individual assets in the portfolio. In this way, an investor can see which assets are most responsible for portfolio risk and can then make informed decisions, if necessary, about rebalancing the portfolio to alter the risk. Portfolio risk budgeting reports, which summarize asset contributions to portfolio risk, are becoming standard in industry practice. In addition, it has become popular to consider portfolio construction using risk budgeting to create so-called “risk-parity” portfolios. In risk-parity portfolios, portfolio weights are constructed so that each asset has equal risk contributions to portfolio risk.¹

We first motivate risk budgeting using the portfolio return variance and standard deviation as measures of portfolio risk. We then show how Euler’s theorem can be used to do risk budgeting for portfolio risk measures that are homogenous of degree one in the portfolio weights. Such risk measures include portfolio return standard deviation (volatility) and Value-at-Risk for general return distributions.

The R packages used in this chapter are **IntroCompFinR** and **PerformanceAnalytics**.

14.1 Risk Budgeting Using Portfolio Variance and Portfolio Standard Deviation

We motivate portfolio risk budgeting in the simple context of a two risky asset portfolio. To illustrate, consider forming a portfolio consisting of two risky assets (asset 1 and asset 2) with portfolio shares x_1 and x_2 such that $x_1 + x_2 = 1$. We assume that the CER model holds for the simple returns on each asset. The portfolio return is given by $R_p = x_1R_1 + x_2R_2$, and the portfolio expected return and variance are given by $\mu_p = x_1\mu_1 + x_2\mu_2$ and $\sigma_p^2 =$

¹ Two hedge funds, Bridgewater and AQR capital management have helped to make risk parity investing popular.

$x_1^2\sigma_1^2 + x_2^2\sigma_2^2 + 2x_1x_2\sigma_{12}$, respectively. The portfolio variance, σ_p^2 , and standard deviation, σ_p , are natural measures of portfolio risk. The advantage of using σ_p is that it is in the same units as the portfolio return. Risk budgeting concerns the following question: what are the contributions of assets 1 and 2 to portfolio risk captured by σ_p^2 or σ_p ?

To answer this question, consider the formula for portfolio variance. If the asset returns are uncorrelated ($\sigma_{12} = 0$) then

$$\sigma_p^2 = x_1^2\sigma_1^2 + x_2^2\sigma_2^2$$

gives an additive decomposition of portfolio risk: $x_1^2\sigma_1^2$ is the variance contribution from asset 1, and $x_2^2\sigma_2^2$ is the variance contribution from asset 2. Clearly, asset 1 has a higher contribution than asset 2 if $x_1^2\sigma_1^2 > x_2^2\sigma_2^2$. For example, in an equally weighted portfolio asset 1's risk contribution is higher than asset 2's contribution if asset 1's return is more variable than asset 2's return. We can also define asset percent contributions to risk, which divide the asset contributions by portfolio variance, as $x_1^2\sigma_1^2/\sigma_p^2$ and $x_2^2\sigma_2^2/\sigma_p^2$, respectively. The percent contributions have the property that they add to 100%.

If we measure risk by portfolio volatility, $\sigma_p = \sqrt{x_1^2\sigma_1^2 + x_2^2\sigma_2^2}$, we may be tempted to define the contributions of assets 1 and 2 to risk as $x_1\sigma_1$ and $x_2\sigma_2$, respectively. However, this would not be correct because we would not get an additive decomposition

$$\sigma_p = \sqrt{x_1^2\sigma_1^2 + x_2^2\sigma_2^2} \neq x_1\sigma_1 + x_2\sigma_2.$$

To get an additive decomposition we have to define the risk contributions as $x_1^2\sigma_1^2/\sigma_p^2$ and $x_2^2\sigma_2^2/\sigma_p^2$, respectively. Then

$$\frac{x_1^2\sigma_1^2}{\sigma_p^2} + \frac{x_2^2\sigma_2^2}{\sigma_p^2} = \frac{\sigma_p^2}{\sigma_p^2} = 1.$$

Notice that the percent contributions to portfolio volatility are the same as the percent contributions to portfolio variance.

If the asset returns are correlated ($\sigma_{12} \neq 0$) then the risk decomposition becomes a bit more complicated because we have to decide what to do with the covariance contribution to portfolio variance. However, the formula for σ_p^2 suggests a simple additive decomposition

$$\begin{aligned}\sigma_p^2 &= x_1^2\sigma_1^2 + x_2^2\sigma_2^2 + 2x_1x_2\sigma_{12} \\ &= (x_1^2\sigma_1^2 + x_1x_2\sigma_{12}) + (x_2^2\sigma_2^2 + x_1x_2\sigma_{12}) \\ &= (x_1^2\sigma_1^2 + x_1x_2\rho_{12}\sigma_1\sigma_2) + (x_2^2\sigma_2^2 + x_1x_2\rho_{12}\sigma_1\sigma_2)\end{aligned}$$

Now, $(x_1^2\sigma_1^2 + x_1x_2\sigma_{12})$ and $(x_2^2\sigma_2^2 + x_1x_2\sigma_{12})$ are portfolio variance contributions of assets 1 and 2, respectively. Here, we split the covariance contribution to portfolio variance, $2x_1x_2\sigma_{12}$, evenly between the two assets. Now, the correlation between the two asset returns also plays a role in the risk decomposition. The additive decomposition for σ_p is

$$\sigma_p = \frac{x_1^2\sigma_1^2 + x_1x_2\sigma_{12}}{\sigma_p} + \frac{x_2^2\sigma_2^2 + x_1x_2\sigma_{12}}{\sigma_p},$$

which gives $\left(\frac{x_1^2\sigma_1^2+x_1x_2\sigma_{12}}{\sigma_p}\right)$ and $\left(\frac{x_2^2\sigma_2^2+x_1x_2\sigma_{12}}{\sigma_p}\right)$ as the asset contributions to portfolio volatility, respectively.

The risk budgeting decompositions presented above for two asset portfolios extend naturally to general N asset portfolios. In the case where all assets are mutually uncorrelated asset i 's contributions to portfolio variance and volatility are $x_i^2\sigma_i^2$ and $x_i^2\sigma_i^2/\sigma_p^2$, respectively. In the case of correlated asset returns, all pairwise covariances enter into asset i 's contributions. The portfolio variance contribution is $x_i^2\sigma_i^2 + \sum_{j \neq i} x_i x_j \sigma_{ij}$ and the portfolio volatility contribution is $(x_i^2\sigma_i^2 + \sum_{j \neq i} x_i x_j \sigma_{ij})/\sigma_p$. As one might expect, these formulae can be simplified using matrix algebra.

14.2 Euler's Theorem and Risk Decompositions

When we used σ_p^2 or σ_p to measure portfolio risk, we were able to easily derive sensible risk decompositions in the two risky asset case. However, if we measure portfolio risk by value-at-risk it is not so obvious how to define individual asset risk contributions. For portfolio risk measures that are homogenous functions of degree one in the portfolio weights, Euler's theorem provides a general method for decomposing risk into asset specific contributions.

14.2.1 Homogenous functions of degree one

First we define a homogenous function of degree one.

Definition 14.1. homogenous function of degree one

Let $f(x_1, x_2, \dots, x_N)$ be a continuous and differentiable function of the variables x_1, x_2, \dots, x_N . The function f is homogeneous of degree one if for any constant c , $f(cx_1, cx_2, \dots, cx_N) = cf(x_1, x_2, \dots, x_N)$. In matrix notation we have $f(x_1, \dots, x_N) = f(\mathbf{x})$ where $\mathbf{x} = (x_1, \dots, x_N)'$. Then f is homogeneous of degree one if $f(c \cdot \mathbf{x}) = c \cdot f(\mathbf{x})$.

Example 14.1. homogenous functions of degree one

Consider the function $f(x_1, x_2) = x_1 + x_2$. Then $f(cx_1, cx_2) = cx_1 + cx_2 = c(x_1 + x_2) = cf(x_1, x_2)$ so that $x_1 + x_2$ is homogenous of degree one. Let $f(x_1, x_2) = x_1^2 + x_2^2$. Then $f(cx_1, cx_2) = c^2x_1^2 + c^2x_2^2 = c^2(x_1^2 + x_2^2) \neq cf(x_1, x_2)$ so that $x_1^2 + x_2^2$ is not homogenous of degree one. Let $f(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$. Then $f(cx_1, cx_2) = \sqrt{c^2x_1^2 + c^2x_2^2} = c\sqrt{(x_1^2 + x_2^2)} = cf(x_1, x_2)$ so that $\sqrt{x_1^2 + x_2^2}$ is homogenous of degree one. In matrix notation, define $\mathbf{x} = (x_1, x_2)'$ and $\mathbf{1} = (1, 1)'$. Let $f(x_1, x_2) = x_1 + x_2 = \mathbf{x}'\mathbf{1} = \mathbf{f}(\mathbf{x})$. Then $f(c \cdot \mathbf{x}) = (c \cdot \mathbf{x})' \mathbf{1} = c \cdot (\mathbf{x}' \mathbf{1}) = c \cdot f(\mathbf{x})$. Let $f(x_1, x_2) = x_1^2 + x_2^2 = \mathbf{x}'\mathbf{x} = f(\mathbf{x})$. Then $f(c \cdot \mathbf{x}) = (c \cdot \mathbf{x})'(c \cdot \mathbf{x}) = c^2 \cdot \mathbf{x}'\mathbf{x} \neq c \cdot f(\mathbf{x})$. Let $f(x_1, x_2) = \sqrt{x_1^2 + x_2^2} = (\mathbf{x}'\mathbf{x})^{1/2} = f(\mathbf{x})$. Then $f(c \cdot \mathbf{x}) = ((c \cdot \mathbf{x})'(c \cdot \mathbf{x}))^{1/2} = c \cdot (\mathbf{x}'\mathbf{x})^{1/2} = c \cdot f(\mathbf{x})$.

■

Consider a portfolio of N assets with weight vector \mathbf{x} , return vector \mathbf{R} , expected return vector μ and covariance matrix Σ . The portfolio return, expected return, variance, and volatility, are functions of the portfolio weight vector \mathbf{x} :

$$\begin{aligned} R_p &= R_p(\mathbf{x}) = \mathbf{x}'\mathbf{R}, \\ \mu_p &= \mu_p(\mathbf{x}) = \mathbf{x}'\mu, \\ \sigma_p^2 &= \sigma_p^2(\mathbf{x}) = \mathbf{x}'\Sigma\mathbf{x}, \\ \sigma_p &= \sigma_p(\mathbf{x}) = (\mathbf{x}'\Sigma\mathbf{x})^{1/2}. \end{aligned}$$

A key result for portfolio risk budgeting is that $R_p(\mathbf{x})$, $\mu_p(\mathbf{x})$, $\sigma_p(\mathbf{x})$, and the normal quantile $q_{\alpha}^{R_p}(x) = \mu_p(\mathbf{x}) + \sigma_p(\mathbf{x})q_{\alpha}^Z$ are homogenous functions of degree one in the portfolio weight vector \mathbf{x} . The result for $R_p(\mathbf{x})$ and $\mu_p(\mathbf{x})$ is trivial since they are linear functions of \mathbf{x} . For example, $R_p(c\mathbf{x}) = (c\mathbf{x})'\mathbf{R} = c(\mathbf{x}'\mathbf{R}) = cR_p(\mathbf{x})$. The result for $\sigma_p(\mathbf{x})$ is straightforward to show:

$$\begin{aligned} \sigma_p(c \cdot \mathbf{x}) &= ((c \cdot \mathbf{x})'\Sigma(c \cdot \mathbf{x}))^{1/2} \\ &= c \cdot (\mathbf{x}'\Sigma\mathbf{x})^{1/2} \\ &= c \cdot \sigma_p(\mathbf{x}). \end{aligned}$$

The result for $q_{\alpha}^{R_p}(\mathbf{x})$ follows because it is a linear function of $\mu_p(\mathbf{x})$ and $\sigma_p(\mathbf{x})$.

Example 14.2. Normal portfolio Value-at-Risk

Normal portfolio VaR treated as a function of the portfolio weight vector \mathbf{x} is a homogenous function of degree one. This result follows from the linear homogeneity of the normal return quantile:

$$\begin{aligned} \text{VaR}_{p,\alpha}(c\mathbf{x}) &= q_{\alpha}^{R_p}(c\mathbf{x})W_0 \\ &= cq_{\alpha}^{R_p}(\mathbf{x})W_0 = c\text{VaR}_{p,\alpha}(\mathbf{x}), \end{aligned}$$

which uses the result that the portfolio quantile $q_{\alpha}^{R_p}(\mathbf{x})$ is homogenous of degree one. ■

14.2.2 Euler's theorem

Euler's theorem gives an additive decomposition of a homogenous function of degree one.

Theorem 14.1. *Euler's theorem*

Let $f(x_1, \dots, x_N) = f(\mathbf{x})$ be a continuous, differentiable and homogenous of degree one function of the variables $\mathbf{x} = (x_1, \dots, x_N)'$. Then,

$$\begin{aligned} f(\mathbf{x}) &= x_1 \cdot \frac{\partial f(\mathbf{x})}{\partial x_1} + x_2 \cdot \frac{\partial f(\mathbf{x})}{\partial x_2} + \dots + x_N \cdot \frac{\partial f(\mathbf{x})}{\partial x_N} \\ &= \mathbf{x}' \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}, \end{aligned}$$

where,

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}_{(N \times 1)} = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_N} \end{pmatrix}.$$

■

Example 14.3. Verifying Euler's theorem

The function $f(x_1, x_2) = x_1 + x_2 = f(\mathbf{x}) = \mathbf{x}'\mathbf{1}$ is homogenous of degree one, and:

$$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial x_1} &= \frac{\partial f(\mathbf{x})}{\partial x_2} = 1, \\ \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} &= \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \mathbf{1}. \end{aligned}$$

By Euler's theorem,

$$\begin{aligned} f(x) &= x_1 \cdot 1 + x_2 \cdot 1 = x_1 + x_2 \\ &= \mathbf{x}'\mathbf{1} \end{aligned}$$

The function $f(x_1, x_2) = (x_1^2 + x_2^2)^{1/2} = f(\mathbf{x}) = (\mathbf{x}'\mathbf{x})^{1/2}$ is homogenous of degree one, and:

$$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial x_1} &= \frac{1}{2} (x_1^2 + x_2^2)^{-1/2} 2x_1 = x_1 (x_1^2 + x_2^2)^{-1/2}, \\ \frac{\partial f(\mathbf{x})}{\partial x_2} &= \frac{1}{2} (x_1^2 + x_2^2)^{-1/2} 2x_2 = x_2 (x_1^2 + x_2^2)^{-1/2}. \end{aligned}$$

By Euler's theorem,

$$\begin{aligned} f(\mathbf{x}) &= x_1 \cdot x_1 (x_1^2 + x_2^2)^{-1/2} + x_2 \cdot x_2 (x_1^2 + x_2^2)^{-1/2} \\ &= (x_1^2 + x_2^2) (x_1^2 + x_2^2)^{-1/2} \\ &= (x_1^2 + x_2^2)^{1/2}. \end{aligned}$$

Using matrix algebra

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial (\mathbf{x}'\mathbf{x})^{1/2}}{\partial \mathbf{x}} = \frac{1}{2} (\mathbf{x}'\mathbf{x})^{-1/2} 2\mathbf{x} = (\mathbf{x}'\mathbf{x})^{-1/2} \mathbf{x} = \mathbf{x} (\mathbf{x}'\mathbf{x})^{-1/2}.$$

Then by Euler's theorem

$$f(\mathbf{x}) = \mathbf{x}' \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}' \mathbf{x} (\mathbf{x}'\mathbf{x})^{-1/2} = (\mathbf{x}'\mathbf{x})^{1/2}.$$

■

14.2.3 Risk decomposition using Euler's theorem

Let $\text{RM}_p(\mathbf{x})$ denote a portfolio risk measure that is a homogenous function of degree one in the portfolio weight vector \mathbf{x} . For example, $\text{RM}_p(\mathbf{x}) = \sigma_p(\mathbf{x})$ or $\text{RM}_p(\mathbf{x}) = \text{VaR}_{p,\alpha}(\mathbf{x})$. Then, Euler's theorem gives the additive risk decomposition

$$\begin{aligned}\text{RM}_p(\mathbf{x}) &= x_1 \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_1} + x_2 \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_2} + \cdots + x_N \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_N} \\ &= \sum_{i=1}^N x_i \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_i} \\ &= \mathbf{x}' \frac{\partial \text{RM}_p(\mathbf{x})}{\partial \mathbf{x}}.\end{aligned}\quad (14.1)$$

The partial derivatives in (14.1) are called asset *marginal contributions to risk* (MCRs):

$$\text{MCR}_i^{RM} = \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_i} = \text{marginal contribution of asset } i, \quad (14.2)$$

The asset *contributions to risk* (CRs) are defined as the weighted marginal contributions:

$$\text{CR}_i^{RM} = x_i \cdot \text{MCR}_i^{RM} = \text{contribution of asset } i, \quad (14.3)$$

Then we can re-express the decomposition (14.1) as

$$\begin{aligned}\text{RM}_p(\mathbf{x}) &= x_1 \cdot \text{MCR}_1^{RM} + x_2 \cdot \text{MCR}_2^{RM} + \cdots + x_N \cdot \text{MCR}_N^{RM} \\ &= \text{CR}_1^{RM} + \text{CR}_2^{RM} + \cdots + \text{CR}_N^{RM}.\end{aligned}\quad (14.4)$$

If we divide both sides of (14.4) by $\text{RM}_p(\mathbf{x})$ we get the asset *percent contributions to risk* (PCRs)

$$\begin{aligned}1 &= \frac{\text{CR}_1^{RM}}{\text{RM}_p(\mathbf{x})} + \frac{\text{CR}_2^{RM}}{\text{RM}_p(\mathbf{x})} + \cdots + \frac{\text{CR}_N^{RM}}{\text{RM}_p(\mathbf{x})} \\ &= \text{PCR}_1^{RM} + \text{PCR}_2^{RM} + \cdots + \text{PCR}_N^{RM},\end{aligned}$$

where

$$\text{PCR}_i^{RM} = \frac{\text{CR}_i^{RM}}{\text{RM}_p(\mathbf{x})} = \text{percent contribution of asset } i. \quad (14.5)$$

By construction the asset PCRs sum to one.

Risk decomposition using $\sigma_p(\mathbf{x})$

Let $\text{RM}_p(\mathbf{x}) = \sigma_p(\mathbf{x}) = (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}$. Because $\sigma_p(\mathbf{x})$ is homogenous of degree 1 in \mathbf{x} , by Euler's theorem

$$\sigma_p(\mathbf{x}) = x_1 \frac{\partial \sigma_p(\mathbf{x})}{\partial x_1} + x_2 \frac{\partial \sigma_p(\mathbf{x})}{\partial x_2} + \cdots + x_n \frac{\partial \sigma_p(\mathbf{x})}{\partial x_n} = \mathbf{x}' \frac{\partial \sigma_p(\mathbf{x})}{\partial \mathbf{x}} \quad (14.6)$$

Now

$$\begin{aligned}\frac{\partial \sigma_p(\mathbf{x})}{\partial \mathbf{x}} &= \frac{\partial (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}}{\partial \mathbf{x}} = \frac{1}{2} (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{-1/2} 2 \boldsymbol{\Sigma} \mathbf{x} \\ &= \frac{\boldsymbol{\Sigma} \mathbf{x}}{(\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2}} = \frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p(\mathbf{x})}.\end{aligned}\quad (14.7)$$

Then

$$\frac{\partial \sigma_p(\mathbf{x})}{\partial x_i} = \text{MCR}_i^\sigma = \text{ith row of } \frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p(\mathbf{x})} = \frac{(\boldsymbol{\Sigma} \mathbf{x})_i}{\sigma_p(\mathbf{x})}, \quad (14.8)$$

and

$$\text{CR}_i^\sigma = x_i \times \frac{(\boldsymbol{\Sigma} \mathbf{x})_i}{\sigma_p(\mathbf{x})}, \quad (14.9)$$

$$\text{PCR}_i^\sigma = x_i \times \frac{(\boldsymbol{\Sigma} \mathbf{x})_i}{\sigma_p^2(\mathbf{x})}. \quad (14.10)$$

Notice that

$$\sigma_p(\mathbf{x}) = \mathbf{x}' \frac{\partial \sigma_p(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{x}' \frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p(\mathbf{x})} = \frac{\sigma_p^2(\mathbf{x})}{\sigma_p(\mathbf{x})} = \sigma_p(\mathbf{x}).$$

Example 14.4. Two asset example

For a two asset portfolio we have

$$\begin{aligned}\sigma_p(\mathbf{x}) &= (\mathbf{x}' \boldsymbol{\Sigma} \mathbf{x})^{1/2} = (x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + 2x_1 x_2 \sigma_{12})^{1/2}, \\ \boldsymbol{\Sigma} \mathbf{x} &= \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \sigma_1^2 + x_2 \sigma_{12} \\ x_2 \sigma_2^2 + x_1 \sigma_{12} \end{pmatrix}, \\ \frac{\boldsymbol{\Sigma} \mathbf{x}}{\sigma_p(\mathbf{x})} &= \begin{pmatrix} (x_1 \sigma_1^2 + x_2 \sigma_{12}) / \sigma_p(\mathbf{x}) \\ (x_2 \sigma_2^2 + x_1 \sigma_{12}) / \sigma_p(\mathbf{x}) \end{pmatrix},\end{aligned}$$

so that

$$\begin{aligned}\text{MCR}_1^\sigma &= (x_1 \sigma_1^2 + x_2 \sigma_{12}) / \sigma_p(\mathbf{x}), \\ \text{MCR}_2^\sigma &= (x_2 \sigma_2^2 + x_1 \sigma_{12}) / \sigma_p(\mathbf{x}).\end{aligned}$$

Then

$$\begin{aligned}\text{MCR}_1^\sigma &= (x_1 \sigma_1^2 + x_2 \sigma_{12}) / \sigma_p(\mathbf{x}), \\ \text{MCR}_2^\sigma &= (x_2 \sigma_2^2 + x_1 \sigma_{12}) / \sigma_p(\mathbf{x}), \\ \text{CR}_1^\sigma &= x_1 \times (x_1 \sigma_1^2 + x_2 \sigma_{12}) / \sigma_p(\mathbf{x}) = (x_1^2 \sigma_1^2 + x_1 x_2 \sigma_{12}) / \sigma_p(\mathbf{x}), \\ \text{CR}_2^\sigma &= x_2 \times (x_2 \sigma_2^2 + x_1 \sigma_{12}) / \sigma_p(\mathbf{x}) = (x_2^2 \sigma_2^2 + x_1 x_2 \sigma_{12}) / \sigma_p(\mathbf{x}),\end{aligned}$$

and

$$\begin{aligned}\text{PCR}_1^\sigma &= \text{CR}_1^\sigma / \sigma_p(\mathbf{x}) = (x_1^2 \sigma_1^2 + x_1 x_2 \sigma_{12}) / \sigma_p^2(\mathbf{x}), \\ \text{PCR}_2^\sigma &= \text{CR}_2^\sigma / \sigma_p(\mathbf{x}) = (x_2^2 \sigma_2^2 + x_1 x_2 \sigma_{12}) / \sigma_p^2(\mathbf{x}).\end{aligned}$$

Notice that risk decomposition from Euler's theorem above is the same decomposition we motived at the beginning of the chapter.

■

Risk decomposition using $\text{VaR}_{p,\alpha}(\mathbf{x})$

Let $\text{RM}(\mathbf{x}) = \text{VaR}_{p,\alpha}(\mathbf{x})$. Because $\text{VaR}_{p,\alpha}(\mathbf{x})$ is homogenous of degree 1 in \mathbf{x} , by Euler's theorem

$$\text{VaR}_{p,\alpha}(\mathbf{x}) = x_1 \frac{\partial \text{VaR}_{p,\alpha}(\mathbf{x})}{\partial x_1} + x_2 \frac{\partial \text{VaR}_{p,\alpha}(\mathbf{x})}{\partial x_2} + \cdots + x_n \frac{\partial \text{VaR}_{p,\alpha}(\mathbf{x})}{\partial x_n} = \mathbf{x}' \frac{\partial \text{VaR}_{p,\alpha}(\mathbf{x})}{\partial \mathbf{x}}. \quad (14.11)$$

Now,

$$\begin{aligned} \text{VaR}_\alpha(\mathbf{x}) &= W_0 \times q_{1-\alpha}^{R_p}(\mathbf{x}) = W_0 \times (\mu_p(\mathbf{x}) + \sigma_p(\mathbf{x}) \times q_\alpha^Z) \\ &= W_0 \times \left(\mathbf{x}' \mu + (\mathbf{x}' \Sigma \mathbf{x})^{1/2} \times q_\alpha^Z \right) \\ \frac{\partial \text{VaR}_\alpha(\mathbf{x})}{\partial \mathbf{x}} &= W_0 \times \frac{\partial}{\partial \mathbf{x}} \left(\mathbf{x}' \mu + (\mathbf{x}' \Sigma \mathbf{x})^{1/2} \times q_\alpha^Z \right) \\ &= W_0 \times \left(\mu + \frac{\Sigma \mathbf{x}}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right) \end{aligned}$$

Then

$$\text{MCR}_i^{\text{VaR}} = W_0 \times \left(\mu_i + \frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right), \quad (14.12)$$

$$\text{CR}_i^{\text{VaR}} = x_i \times W_0 \times \left(\mu_i + \frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right), \quad (14.13)$$

$$\text{PCR}_i^{\text{VaR}} = x_i \times W_0 \times \left(\mu_i + \frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right) / (W_0 \times (\mu_p(\mathbf{x}) + \sigma_p(\mathbf{x}) \times q_\alpha^Z)). \quad (14.14)$$

It is often common practice to set $\mu = 0$ when computing $\text{VaR}_{p,\alpha}(\mathbf{x})$. In this case,

$$\begin{aligned} \text{MCR}_i^{\text{VaR}} &= W_0 \times \left(\frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right) \propto \text{MCR}_i^\sigma, \\ \text{CR}_i^{\text{VaR}} &= x_i \times W_0 \times \left(\frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})} \times q_\alpha^Z \right) \propto \text{CR}_i^\sigma, \\ \text{PCR}_i^{\text{VaR}} &= x_i \times \frac{(\Sigma \mathbf{x})_i}{\sigma_p^2(\mathbf{x})} = \text{PCR}_i^\sigma, \end{aligned}$$

and we see that the portfolio normal VaR risk decompositions above give the same information as the portfolio volatility risk decompositions (14.8) - (14.10).

Interpreting marginal contributions to risk

The risk decomposition (14.1) shows that any risk measure that is homogenous of degree one in the portfolio weights can be additively decomposed into a portfolio weighted average of marginal contributions to risk. The marginal contributions to risk (14.2) are the partial derivatives of the risk measure $\text{RM}_p(\mathbf{x})$ with respect to the portfolio weights, and so one may think that they can be interpreted as the change in the risk measure associated with a one unit change in a portfolio weight holding the other portfolio weight fixed

$$\text{MCR}_i^\sigma = \frac{\partial \sigma_p(\mathbf{x})}{\partial x_i} \approx \frac{\Delta \sigma_p}{\Delta x_i} \Rightarrow \Delta \sigma_p \approx \text{MCR}_i^\sigma \cdot \Delta x_i.$$

If the portfolio weights were unconstrained this would be the correct interpretation. However, the portfolio weights are constrained to sum to one, $\sum_{i=1}^N x_i = 1$, so the increase in one weight implies an offsetting decrease in the other weights. Hence, the formula $\Delta \sigma_p \approx \text{MCR}_i^\sigma \cdot \Delta x_i$ ignores this re-allocation effect.

To properly interpret the marginal contributions to risk, consider the total derivative of $\text{RM}_p(\mathbf{x})$

$$\begin{aligned} d\text{RM}_p(\mathbf{x}) &= \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_1} dx_1 + \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_2} dx_2 + \cdots + \frac{\partial \text{RM}_p(\mathbf{x})}{\partial x_N} dx_N \\ &= \text{MCR}_1^{RM} dx_1 + \text{MCR}_2^{RM} dx_2 + \cdots + \text{MCR}_N^{RM} dx_N. \end{aligned} \quad (14.15)$$

First, consider a small change in x_i offset by an equal change in x_j , $\Delta x_i = -\Delta x_j$. That is, the small increase in allocation to asset i is matched by a corresponding decrease in allocation to asset j . From (14.15), the approximate change in the portfolio risk measure is

$$\Delta \text{RM}_p(\mathbf{x}) \approx \left(\text{MCR}_i^{RM} - \text{MCR}_j^{RM} \right) \Delta x_i. \quad (14.16)$$

When rebalancing a portfolio, the reallocation does not have to be limited to two assets. Suppose, for example, the reallocation is spread across all of the other assets $j \neq i$ so that

$$\Delta x_j = -\alpha_j \Delta x_i \text{ s.t. } \sum_{j \neq i} \alpha_j = 1$$

Then

$$\sum_{j \neq i} \Delta x_j = -\sum_{j \neq i} \alpha_j \Delta x_i = -\Delta x_i \sum_{j \neq i} \alpha_j = -\Delta x_i$$

and

$$\begin{aligned} \Delta \text{RM}(\mathbf{x}) &\approx \text{MCR}_i^{RM} \cdot \Delta x_i + \sum_{j \neq i} \text{MCR}_j^{RM} \cdot \Delta x_j = \\ &= \left(\text{MCR}_i^{RM} - \sum_{j \neq i} \alpha_j \cdot \text{MCR}_j^{RM} \right) \Delta x_i \end{aligned} \quad (14.17)$$

In matrix notation the result (14.17) be written as

$$\begin{aligned}\Delta \text{RM}(\mathbf{w}) &\approx \left((\mathbf{MCR}^{RM})' \boldsymbol{\alpha} \right) \Delta x_i, \\ \mathbf{MCR}^{RM} &= (\text{MCR}_1^{RM}, \dots, \text{MCR}_n^{RM})', \\ \boldsymbol{\alpha} &= (-\alpha_1, \dots, -\alpha_{i-1}, 1, -\alpha_{i+1}, \dots, -\alpha_n)'.\end{aligned}\tag{14.18}$$

Portfolio risk reports

A portfolio risk report summarizes asset and portfolio risk measures as well as risk budgets. Table 14.2.3 shows a typical portfolio risk report. The individual asset information is in rows with the portfolio information at the bottom. The total dollar amount invested in the portfolio is W_0 and the dollar amounts invested in each asset are $d_i = x_i W_0$. The asset specific (standalone) risk measures are RM_i and the portfolio risk measure is $\text{RM}(\mathbf{x})$.

Asset	\$d _i	x _i	RM _i	MCR _i ^{RM}	CR _i ^{RM}	PCR _i ^{RM}
Asset 1	\$d ₁	x ₁	RM ₁	MCR ₁ ^{RM}	CR ₁ ^{RM}	PCR ₁ ^{RM}
Asset 2	\$d ₂	x ₂	RM ₂	MCR ₂ ^{RM}	CR ₂ ^{RM}	PCR ₂ ^{RM}
:	:	:	:	:	:	:
Asset N	\$d _N	x _N	RM _N	MCR _N ^{RM}	CR _N ^{RM}	PCR _N ^{RM}
Portfolio (Sum)	\$W_0	1			RM(\mathbf{x})	1

Table 14.1 Portfolio Risk Report

Example 14.5. Portfolio volatility risk report

Consider creating a portfolio volatility risk report from an equally weighted portfolio of Microsoft, Nordstrom, and Starbucks stock. The initial wealth invested in the portfolio is \$100,000. The expected return vector and covariance matrix is based on sample statistics computed over the five-year period January, 1995 through January, 2000. The asset and portfolio expected return and volatility information is

```
# asset information
asset.names <- c("MSFT", "NORD", "SBUX")
mu.vec = c(0.0427, 0.0015, 0.0285)
sigma.mat = matrix(c(0.01, 0.0018, 0.0011, 0.0018, 0.0109, 0.0026, 0.0011, 0.0026,
  0.0199), nrow = 3, ncol = 3)
sig.vec = sqrt(diag(sigma.mat))
names(mu.vec) = names(sig.vec) = asset.names
dimnames(sigma.mat) = list(asset.names, asset.names)
# equally weighted portfolio information
W0 = 1e+05
x = rep(1/3, 3)
```

```
d = x * W0
names(x) = asset.names
mu.px = as.numeric(crossprod(x, mu.vec))
sig.px = as.numeric(sqrt(t(x) %*% sigma.mat %*% x))
```

The volatility risk budgeting calculations are:

```
MCR.vol.x = (sigma.mat %*% x)/sig.px
CR.vol.x = x * MCR.vol.x
PCR.vol.x = CR.vol.x/sig.px
```

The volatility risk report is computed using:

```
riskReportVol.px = cbind(d, x, sig.vec, MCR.vol.x, CR.vol.x, PCR.vol.x)
PORT = c(W0, 1, NA, NA, sum(CR.vol.x), sum(PCR.vol.x))
riskReportVol.px = rbind(riskReportVol.px, PORT)
colnames(riskReportVol.px) = c("Dollar", "Weight", "Vol", "MCR", "CR", "PCR")
riskReportVol.px

##      Dollar Weight   Vol    MCR     CR    PCR
## MSFT  33333  0.333 0.100 0.0567 0.0189 0.249
## NORD  33333  0.333 0.104 0.0672 0.0224 0.295
## SBUX  33333  0.333 0.141 0.1037 0.0346 0.456
## PORT 100000  1.000    NA     NA 0.0759 1.000
```

In the equally weighted portfolio, the risk contributions are not equal across assets. Predictably, the ranking of the risk contributions follows the ranking of the individual asset volatilities σ_i , with Starbucks giving the highest contributions to portfolio volatility. So if the portfolio manager wants to reduce portfolio volatility the allocation to Starbucks should be reduced first.

To interpret the marginal contributions to risk, suppose that the portfolio manager wants to reduce portfolio volatility and chooses the rebalancing strategy: $\Delta x_{MSFT} = -\Delta x_{SBUX} = 0.1$ (i.e., x_{MSFT} increases to 0.433 and x_{SBUX} decreases to 0.233). Then, from (14.16) the predicted change in portfolio standard deviation is

```
delta.vol.px = (MCR.vol.x["MSFT", ] - MCR.vol.x["SBUX", ]) * 0.1
delta.vol.px

##      MSFT
## -0.0047
```

Hence, the predicted volatility after rebalancing is

```
sig.px + delta.vol.px

##      MSFT
## 0.0712
```

The exact change in volatility from rebalancing is

```

x1 = x + c(0.1, 0, -0.1)
sig.pxi = as.numeric(sqrt(t(x1) %*% sigma.mat %*% x1))
sig.pxi

## [1] 0.0729

sig.pxi - sig.px

## [1] -0.00293

```

■

Example 14.6. Portfolio normal VaR risk report

Using the asset and portfolio data from the previous example, consider creating a normal portfolio VaR report, where VaR is computed with 5% probability. The R calculations are

```

# portfolio 5% normal VaR
alpha = 0.05
# risk budget calculations
VaR.px = abs(W0 * (mu.px + sig.px * qnorm(alpha)))
MCR.VaR.x = abs(W0 * (mu.vec + MCR.vol.x * qnorm(alpha)))
CR.VaR.x = x * MCR.VaR.x
PCR.VaR.x = CR.VaR.x/VaR.px
# risk report
VaR.vec = abs(W0 * (mu.vec + sig.vec * qnorm(alpha)))
riskReportVaR.px = cbind(d, x, VaR.vec, MCR.VaR.x, CR.VaR.x, PCR.VaR.x)
PORT = c(W0, 1, NA, NA, sum(CR.VaR.x), sum(PCR.VaR.x))
riskReportVaR.px = rbind(riskReportVaR.px, PORT)
colnames(riskReportVaR.px) = c("Dollar", "Weight", "VaR", "MCR", "CR", "PCR")
riskReportVaR.px

##      Dollar Weight   VaR    MCR     CR    PCR
## MSFT  33333  0.333 12179  5053  1684  0.168
## NORD  33333  0.333 17023 10907  3636  0.362
## SBUX  33333  0.333 20354 14206  4735  0.471
## PORT 100000  1.000    NA     NA 10055  1.000

```

The qualitative information in the portfolio normal VaR report is essentially the same as in the portfolio volatility report. Because the VaR calculation involve the asset expected returns, μ , the results are not exactly the same. If we set $\mu = \mathbf{0}$ in the VaR calculations then the two reports would give same risk information but on different scales (volatility is in return and VaR is in dollars).

To interpret the marginal contributions to risk, suppose that the portfolio manager wants to reduce portfolio VaR and chooses the rebalancing strategy: $\Delta x_{MSFT} = -\Delta x_{SBUX} = 0.1$ (i.e., x_{MSFT} increases to 0.433 and x_{SBUX} decreases to 0.233). Then, from (14.16) the predicted change in portfolio VaR and new portfolio VaR are

```

delta.VaR.px = (MCR.VaR.x["MSFT", ] - MCR.VaR.x["SBUX", ]) * 0.1
delta.VaR.px

```

```

## MSFT
## -915

VaR.px + delta.VaR.px

## MSFT
## 9140

```

The exact change in portfolio VaR from rebalancing is

```

mu.px1 = as.numeric(crossprod(x1, mu.vec))
VaR.px1 = abs(W0 * (mu.px1 + sig.px1 * qnorm(alpha)))
VaR.px1 - VaR.px

## [1] -624

VaR.px1

## [1] 9431

```

Here, the predicted reduction in VaR is about \$300 more than the actual change in VaR.

■

14.3 Understanding Portfolio Volatility Risk Decompositions

Portfolio volatility risk reports described in the previous sub-section are commonly used in practice. As such, it is important to have a good understanding of the volatility risk decompositions and, in particular, the asset contributions to portfolio volatility. In the portfolio volatility risk decomposition (14.6), the marginal contribution from asset i given by

$$\text{MCR}_i^\sigma = \text{ith row of } \frac{\Sigma \mathbf{x}}{\sigma_p(\mathbf{x})} = \frac{(\Sigma \mathbf{x})_i}{\sigma_p(\mathbf{x})}. \quad (14.19)$$

The interpretation of this formula, however, is not particularly intuitive. With a little bit of algebra, we can derive two alternative representations of (14.19) that are easier to interpret and give better intuition about the asset contributions to portfolio volatility.

14.3.1 $\sigma - \rho$ decomposition for MCR_i^σ

The $\sigma - \rho$ decomposition for MCR_i^σ is

$$\text{MCR}_i^\sigma = \sigma_i \rho_{i,p}(\mathbf{x}), \quad (14.20)$$

where $\rho_{i,p}(\mathbf{x}) = \text{corr}(R_i, R_p(\mathbf{x}))$ is the correlation between the return on asset i and the return on the portfolio.

Remarks:

1. Equation (14.20) shows that an asset's marginal contribution to portfolio volatility depends on two components: (1) the asset's return volatility, σ_i (sometimes called standalone volatility); (2) the asset's correlation with the portfolio return, $\rho_{i,p}(\mathbf{x})$.
2. For a given standalone volatility, σ_i , the sign and magnitude of MCR_i^σ depends on $\rho_{i,p}(\mathbf{x})$. Because $-1 \leq \rho_{i,p}(\mathbf{x}) \leq 1$, it follows that $MCR_i^\sigma \leq \sigma_i$ and $MCR_i^\sigma = \sigma_i$ only if $\rho_{i,p}(\mathbf{x}) = 1$.
3. If an asset's return is uncorrelated with all of the returns in the portfolio then $\rho_{i,p}(\mathbf{x}) = (x_i\sigma_i)/\sigma_p$ and $MCR_i^\sigma = (x_i\sigma_i^2)/\sigma_p$.

Using (14.20) we arrive at the $x - \sigma - \rho$ decomposition for $CR_i^\sigma = x_i \times MCR_i^\sigma$:

$$\begin{aligned} CR_i^\sigma &= x_i \times \sigma_i \times \rho_{i,p}(\mathbf{x}) \\ &= \text{asset allocation} \times \text{standalone volatility} \times \text{correlation with portfolio} \end{aligned} \quad (14.21)$$

Remarks:

1. Equation (14.21) shows that an asset's contribution to portfolio volatility depends on three components: (1) the asset's allocation weight x_i ; (2) the asset's standalone return volatility, σ_i ; (3) the asset's correlation with the portfolio return, $\rho_{i,p}(\mathbf{x})$.
2. $x_i \times \sigma_i$ = standalone contribution to portfolio volatility, which ignores correlation effects with other assets. In the typical situation, $\rho_{i,p} \neq 1$ which implies that $CR_i^\sigma < w_i \times \sigma_i$. Hence, an asset's contribution to portfolio volatility will almost always be less than its standalone contribution to portfolio volatility.
3. $CR_i^\sigma = x_i \times \sigma_i$ only when $\rho_{i,p}(\mathbf{x}) = 1$. That is, an asset's contribution to portfolio volatility is equal to its standalone contribution only when its return is perfectly correlated with the portfolio return.
4. If all assets are perfectly correlated (i.e., $\rho_{ij} = 1$ for all i and j) then all asset contributions to portfolio volatility are equal their standalone contributions, $x_i \times \sigma_i$, and $\sigma_p(\mathbf{x}) = x_1\sigma_1 + x_2\sigma_2 + \dots + x_N\sigma_N$. This is the risk decomposition that would occur if there are no diversification effects.

The derivation of (14.20) is straightforward. Recall, the vector of asset marginal contributions to portfolio volatility is given by

$$\frac{\partial \sigma_p(\mathbf{x})}{\partial \mathbf{x}} = \frac{\Sigma \mathbf{x}}{\sigma_p(\mathbf{x})}.$$

Now,

$$\Sigma \mathbf{x} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Without loss of generality, consider the first row of $\Sigma \mathbf{x}$

$$x_1\sigma_1^2 + x_2\sigma_{12} + \cdots + x_n\sigma_{1n}.$$

This expression is the covariance between the return on asset 1, R_1 , and the return on the portfolio, $R_p(\mathbf{x})$:

$$\begin{aligned}\text{cov}(R_1, R_p(\mathbf{x})) &= \text{cov}(R_1, x_1R_1 + R_1x_2R_2 + \cdots + x_nR_n) \\ &= \text{cov}(R_1, x_1R_1) + \text{cov}(R_1, x_1R_1) + \cdots + \text{cov}(R_1, x_nR_n) \\ &= x_1\sigma_1^2 + x_2\sigma_{12} + \cdots + x_n\sigma_{1n} \\ &= \sigma_{1,p}(\mathbf{x}).\end{aligned}$$

Then we can re-write (14.19) for $i = 1$ as

$$\text{MCR}_1^\sigma = \frac{(\Sigma\mathbf{x})_1}{\sigma_p(\mathbf{x})} = \frac{\sigma_{1,p}(\mathbf{x})}{\sigma_p(\mathbf{x})}. \quad (14.22)$$

Define the correlation between the return on asset 1 and the return on the portfolio as

$$\rho_{1,p}(\mathbf{x}) = \text{corr}(R_1, R_p(\mathbf{x})) = \frac{\text{cov}(R_1, R_p(\mathbf{x}))}{\sigma_1\sigma_p(\mathbf{x})} = \frac{\sigma_{1,p}(\mathbf{x})}{\sigma_1\sigma_p(\mathbf{x})} \quad (14.23)$$

Then we can write

$$\sigma_{1,p}(\mathbf{x}) = \sigma_1\sigma_p(\mathbf{x})\rho_{1,p}(\mathbf{x}) \quad (14.24)$$

Substituting (14.24) into (14.22) gives the $\sigma - \rho$ decomposition for MCR_1^σ

$$\text{MCR}_1^\sigma = \frac{\sigma_1\sigma_p(\mathbf{x})\rho_{1,p}(\mathbf{x})}{\sigma_p(\mathbf{x})} = \sigma_1\rho_{1,p}(\mathbf{x}).$$

Example 14.7. Adding $\rho_{i,p}(\mathbf{x})$ to the portfolio volatility risk report

Once you have the volatility risk report, you can easily calculate $\rho_{i,p}(\mathbf{x})$ from (14.20) using

$$\rho_{i,p}(\mathbf{x}) = \frac{\text{MCR}_i^\sigma}{\sigma_i}.$$

For example, to calculate $\rho_{i,p}(\mathbf{x})$ for the volatility risk report for the equally weighted portfolio of Microsoft, Nordstrom, and Starbucks use:

```
rho.x = MCR.vol.x/sig.vec
rho.x

##      [,1]
## MSFT  0.567
## NORD  0.644
## SBUX  0.735
```

Here we see that all assets are positively correlated with the portfolio, Microsoft has the smallest correlation (0.567), and Starbucks has the largest correlation (0.735). In this respect, Microsoft is most beneficial in terms of diversification and Starbucks is least beneficial. If all correlations were equal to one, then each asset MCR would be equal to its standalone volatility.



14.3.2 $\sigma - \beta$ decomposition for MCR_i^σ

The $\sigma - \beta$ decomposition for MCR_i^σ is

$$MCR_i^\sigma = \beta_{i,p}(\mathbf{x})\sigma_p(\mathbf{x}), \quad (14.25)$$

where

$$\beta_{i,p}(\mathbf{x}) = \frac{\text{cov}(R_i, R_p(\mathbf{x}))}{\text{var}(R_p(\mathbf{x}))} = \frac{\text{cov}(R_i, R_p(\mathbf{x}))}{\sigma_p^2(\mathbf{x})}. \quad (14.26)$$

Here, $\beta_{i,p}(\mathbf{x})$ is called asset i 's “beta” to the portfolio.² This decomposition follows directly from the $\sigma - \rho$ decomposition (14.20) since

$$\beta_{i,p}(\mathbf{x}) = \frac{\text{cov}(R_i, R_p(\mathbf{x}))}{\sigma_p^2(\mathbf{x})} = \frac{\rho_{i,p}(\mathbf{x})\sigma_i\sigma_p(\mathbf{x})}{\sigma_p^2(\mathbf{x})} \Rightarrow \rho_{i,p}(x) = \frac{\beta_{i,p}(\mathbf{x})\sigma_p(\mathbf{x})}{\sigma_i}.$$

It follows that

$$CR_i^\sigma = x_i \times \beta_{i,p}(\mathbf{x}) \times \sigma_p(\mathbf{x}), \quad (14.27)$$

$$PCR_i^\sigma = x_i \times \beta_{i,p}(\mathbf{x}). \quad (14.28)$$

Remarks:

- By construction, the beta of the portfolio is 1

$$\beta_{p,p}(\mathbf{x}) = \frac{\text{cov}(R_p(\mathbf{x}), R_p(\mathbf{x}))}{\text{var}(R_p(\mathbf{x}))} = \frac{\text{var}(R_p(\mathbf{x}))}{\text{var}(R_p(\mathbf{x}))} = 1.$$

- When $\beta_{i,p}(\mathbf{x}) = 1$, $MCR_i^\sigma = \sigma_p(\mathbf{x})$, $CR_i^\sigma = x_i\sigma_p(\mathbf{x})$, and $PCR_i^\sigma = x_i$. In this case, an asset's marginal contribution to portfolio volatility is portfolio volatility and its percent contribution to portfolio volatility is its allocation weight. Intuitively, when $\beta_{i,p}(x) = 1$ the asset has the same risk, in terms of volatility contribution, as the portfolio.
- When $\beta_{i,p}(\mathbf{x}) > 1$, $MCR_i^\sigma > \sigma_p(\mathbf{x})$, $CR_i^\sigma > w_i\sigma_p(\mathbf{x})$, and $PCR_i^\sigma > x_i$. In this case, the asset's marginal contribution to portfolio volatility is more than portfolio volatility and its percent contribution to portfolio volatility is more than its allocation weight. Intuitively, when $\beta_{i,p}(\mathbf{x}) > 1$ the asset has more risk, in terms of volatility contribution, than the portfolio. That is, having the asset in the portfolio increases the portfolio volatility.
- When $\beta_{i,p}(\mathbf{x}) < 1$, $MCR_i^\sigma < \sigma_p(\mathbf{x})$, $CR_i^\sigma < x_i\sigma_p(\mathbf{x})$, and $PCR_i^\sigma < x_i$. In this case, the asset's marginal contribution to portfolio volatility is less than portfolio volatility and its percent contribution to portfolio volatility is less than its allocation weight. Intuitively, when $\beta_{i,p}(\mathbf{x}) < 1$ the asset has less risk, in terms of volatility contribution, than the portfolio. That is, having the asset in the portfolio decreases the portfolio volatility.

Example 14.8. Adding $\beta_{i,p}(\mathbf{x})$ to the portfolio volatility risk report

Once you have the volatility risk report, you can easily calculate $\beta_{i,p}(\mathbf{x})$ from (14.25) using

² It can be thought of as the population regression coefficient obtained by regressing the asset return, R_i , on the portfolio return, $R_p(\mathbf{x})$.

$$\beta_{i,p}(\mathbf{x}) = \frac{\text{MCR}_i^\sigma}{\sigma_p(\mathbf{x})}.$$

For example, to calculate $\beta_{i,p}(\mathbf{x})$ for the volatility risk report for the equally weighted portfolio of Microsoft, Nordstrom, and Starbucks use:

```
beta.x = MCR.vol.x/sig.px
beta.x

##      [,1]
## MSFT 0.747
## NORD 0.886
## SBUX 1.367
```

Here, both $\beta_{MSFT,p}(\mathbf{x})$ and $\beta_{NORD,p}(\mathbf{x})$ are less than one whereas $\beta_{SBUX,p}(\mathbf{x})$ is greater than one. We can say that Microsoft and Nordstrom are portfolio volatility reducers whereas Starbucks is a portfolio volatility enhancer.

■

14.4 Risk Budgeting for Optimized Portfolios

To be completed

- For global minimum variance portfolio, FOCs show that $MCR_i = MCR_j$. Also plugging in solution for m shows that vector of MCR is the same.
- For an efficient portfolio with target return equal to the mean return on an asset in the portfolio, the portfolio beta is 1: $\beta_{i,p} = 1$ when $\mu_p = \mu_i$. Hence, $CR_i = x_i \sigma_p$
- For the tangency portfolio, FOCs (from re-written equivalent problem) show that

$$\frac{\mu_T - r_f}{\sigma_T^2} = \frac{\mu_i - r_f}{\sigma_{i,T}}$$

Volatility risk budgets for optimized portfolios have some interesting properties. For example, consider the global minimum variance portfolio allowing for short sales. In chapter 12 we showed that

$$\mathbf{m} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}'\Sigma^{-1}\mathbf{1}}.$$

Plugging this portfolio weight vector into (14.7) gives

$$\frac{\partial \sigma_p(\mathbf{m})}{\partial \mathbf{m}} = \frac{\Sigma \mathbf{m}}{\sigma_p(\mathbf{m})} = \frac{\Sigma \Sigma^{-1}\mathbf{1}}{\sigma_p(\mathbf{m}) \times \mathbf{1}'\Sigma^{-1}\mathbf{1}} = \left(\frac{1}{\sigma_p(\mathbf{m}) \times \mathbf{1}'\Sigma^{-1}\mathbf{1}} \right) \times \mathbf{1} = c \times \mathbf{1}$$

which is a constant c times the one vector where

$$c = \frac{1}{\sigma_p(\mathbf{m}) \times \mathbf{1}'\Sigma^{-1}\mathbf{1}}.$$

As a result, for each asset $i = 1, \dots, n$ we have

$$\text{MCR}_i^\sigma = c.$$

That is, each asset in the global minimum variance portfolio has the same marginal contribution to portfolio volatility. This has to be the case, otherwise we could rebalance the portfolio and lower the volatility. To see this, suppose $\text{MCR}_i^\sigma > c$ for some asset i . Then we could lower portfolio volatility by reducing the allocation to asset i and increasing the allocation to any other asset in the portfolio. Similarly, if $\text{MCR}_i^\sigma < c$ we could lower portfolio volatility by increasing the allocation to asset i and decreasing the allocation to any other asset in the portfolio.

Example 14.9. Volatility risk budgeting for the global minimum variance portfolio

to be completed

■

14.5 Reverse Optimization and Implied Returns

- Standard portfolio optimization begins with a set of expected returns and risk forecasts.
- These inputs are fed into an optimization routine, which then produces the portfolio weights that maximizes some risk-to-reward ratio (typically subject to some constraints).
- Reverse optimization, by contrast, begins with a set of portfolio weights and risk forecasts, and then infers what the implied expected returns must be to satisfy optimality.

Suppose that the objective is to form a portfolio by maximizing a generalized expected return-to-risk (Sharpe) ratio:

$$\begin{aligned} \max_{\mathbf{x}} \text{GSR}(\mathbf{x}) &= \frac{\mu_p(\mathbf{x}) - r_f}{\text{RM}(\mathbf{x})}, \\ \mu_p(\mathbf{x}) &= \mathbf{x}'\boldsymbol{\mu} \\ \text{RM}(\mathbf{x}) &= \text{linearly homogenous risk measure} \end{aligned}$$

where $\mu_p(\mathbf{x}) = \mathbf{x}'\boldsymbol{\mu}$ and $\text{RM}(\mathbf{x})$ is a linearly homogenous risk measure. For example, if $\text{RM}(\mathbf{x}) = \sigma_p(\mathbf{x})$ then $\text{GSR}(\mathbf{x}) = \text{SR}(\mathbf{x}) = \frac{\mu_p(\mathbf{x}) - r_f}{\sigma_p(\mathbf{x})}$. Using the chain-rule, the F.O.C.'s of the optimization are ($i = 1, \dots, n$)

$$0 = \frac{\partial}{\partial x_i} \left(\frac{\mu_p(\mathbf{x}) - r_f}{\text{RM}(\mathbf{x})} \right) = \frac{1}{\text{RM}(\mathbf{x})} \frac{\partial (\mu_p(\mathbf{x}) - r_f)}{\partial x_i} - \frac{\mu_p(\mathbf{x}) - r_f}{\text{RM}(\mathbf{x})^2} \frac{\partial \text{RM}(\mathbf{x})}{\partial x_i}$$

Reverse optimization uses the above optimality condition with fixed portfolio weights to determine the optimal fund expected returns. These optimal expected returns are called *implied returns*. The implied returns satisfy

$$\begin{aligned} \mu_i^{\text{implied}}(\mathbf{x}) &= \frac{\mu_p(\mathbf{x}) - r_f}{\text{RM}(\mathbf{x})} \times \frac{\partial \text{RM}(\mathbf{x})}{\partial x_i} \\ &= \text{GSR}(\mathbf{x}) \times \text{MCR}_i \end{aligned}$$

Result: fund i 's implied return is proportional to its marginal contribution to risk, with the constant of proportionality being the generalized Sharpe ratio of the portfolio.

14.5.1 How to Use Implied Returns

- For a given value of GSR(\mathbf{x}), $\mu_i^{\text{implied}}(\mathbf{x})$ is large if MCR $_i$ is large.
- If the actual or forecast expected return for fund i is less than its implied return, then one should reduce one's holdings of that asset
- If the actual or forecast expected return for fund i is greater than its implied return, then one should increase one's holdings of that asset

14.6 Risk Parity Portfolios

To be completed.

14.7 Further Reading

To be completed

Chapter 15

Statistical Analysis of Portfolios

Updated: August 11, 2016
 Copyright © Eric Zivot 2015, 2016

15.1 Statistical Analysis of Portfolios: Two Assets

Most of the issues involved with the statistical analysis of portfolios can be illustrated in the simple case of a two asset portfolio. Let R_A and R_B denote the simple returns on two risky assets and assume these returns are characterized by the CER model:

$$\begin{pmatrix} R_A \\ R_B \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \begin{pmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{pmatrix} \right). \quad (15.1)$$

In matrix notation we have:

$$\mathbf{R} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where

$$\mathbf{R} = \begin{pmatrix} R_A \\ R_B \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_A \\ \mu_B \end{pmatrix}, \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{pmatrix}.$$

A portfolio with weight vector $\mathbf{x} = (x_A, x_B)'$ has return $R_p = \mathbf{x}'\mathbf{R} = x_A R_A + x_B R_B$ which is also described by the CER model

$$\begin{aligned} R_p &\sim N(\mu_p, \sigma_p^2), \\ \mu_p &= \mathbf{x}'\boldsymbol{\mu} = x_A \mu_A + x_B \mu_B, \\ \sigma_p^2 &= \mathbf{x}'\boldsymbol{\Sigma}\mathbf{x} = x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}. \end{aligned}$$

We observe a sample of asset returns of size T , $\{r_t\}_{t=1}^T$, where $\mathbf{r}_t = (r_{At}, r_{Bt})'$, that we assume is generated from the CER model (15.1) from which we create the sample portfolio returns $r_{p,t} = \mathbf{x}'\mathbf{r}_t = x_A r_{At} + x_B r_{Bt}$.

The true CER model parameters are unknown in practice and must be estimated from the observed data. From chapter xxx, the CER model parameter estimates are the corresponding sample statistics:

$$\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^T r_{it}, \hat{\sigma}_i^2 = \frac{1}{T-1} \sum_{t=1}^T (r_{it} - \hat{\mu}_i)^2, \hat{\sigma}_i = \sqrt{\hat{\sigma}_i^2}, \hat{\sigma}_{ij}^2 = \frac{1}{T-1} \sum_{t=1}^T (r_{it} - \hat{\mu}_i)(r_{jt} - \hat{\mu}_j). \quad (15.2)$$

These estimates are unbiased, asymptotically normally distributed, and have estimation errors which can be quantified by standard errors and 95% confidence intervals.¹ For $\hat{\mu}_i$, $\hat{\sigma}_i^2$, $\hat{\sigma}_i$ we have analytic standard error formulas:

$$\text{se}(\hat{\mu}_i) = \frac{\sigma_i}{\sqrt{T}}, \text{se}(\hat{\sigma}_i^2) \approx \frac{\sigma_i^2}{\sqrt{T/2}}, \text{se}(\hat{\sigma}_i) \approx \frac{\sigma_i}{\sqrt{2T}}. \quad (15.3)$$

For $\hat{\sigma}_{ij}^2$ there is no easy formula but we can use the bootstrap to compute a numerical estimate of the standard error. For 95% confidence intervals, we use the rule of thumb “estimate $\pm 2 \times \text{standard error}$ ”.

We can estimate the CER model parameters μ_p , σ_p^2 and σ_p for the portfolio return, R_p , using two equivalent methods. In the first method, we use the sample portfolio returns $\{r_{p,t}\}_{t=1}^T$ and compute sample statistics:

$$\hat{\mu}_p = \frac{1}{T} \sum_{t=1}^T r_{p,t}, \hat{\sigma}_p^2 = \frac{1}{T-1} \sum_{t=1}^T (r_{p,t} - \hat{\mu}_p)^2, \hat{\sigma}_p = \sqrt{\hat{\sigma}_p^2}. \quad (15.4)$$

These estimates are unbiased and asymptotically normal with standard errors:

$$\text{se}(\hat{\mu}_p) = \frac{\sigma_p}{\sqrt{T}}, \text{se}(\hat{\sigma}_p^2) \approx \frac{\sigma_p^2}{\sqrt{T/2}}, \text{se}(\hat{\sigma}_p) \approx \frac{\sigma_p}{\sqrt{2T}}. \quad (15.5)$$

In the second method, we compute estimates of μ_p and σ_p^2 directly from the individual asset estimates in (15.2):

$$\hat{\mu}_p = \mathbf{x}' \hat{\mu} = x_A \hat{\mu}_A + x_B \hat{\mu}_B, \quad (15.6)$$

$$\hat{\sigma}_p^2 = \mathbf{x}' \hat{\Sigma} \mathbf{x} = x_A^2 \hat{\sigma}_A^2 + x_B^2 \hat{\sigma}_B^2 + 2x_A x_B \hat{\sigma}_{AB}. \quad (15.7)$$

The estimates of μ_p and σ_p^2 in (15.6) and (15.7), respectively, are numerically identical to the estimates in (15.4). To see this, consider the calculation of $\hat{\mu}_p$ in (15.6):

$$\begin{aligned} \hat{\mu}_p &= x_A \hat{\mu}_A + x_B \hat{\mu}_B = x_A \left(\frac{1}{T} \sum_{t=1}^T r_{At} \right) + x_B \left(\frac{1}{T} \sum_{t=1}^T r_{Bt} \right) \\ &= \frac{1}{T} \sum_{t=1}^T (x_A r_{At} + x_B r_{Bt}) = \frac{1}{T} \sum_{t=1}^T r_{p,t}. \end{aligned}$$

¹ $\hat{\sigma}_i$ is asymptotically unbiased and the bias for finite T is typically very small so that it is essentially unbiased for moderately sized T .

The proof of the equivalence of $\hat{\sigma}_p^2$ in (15.7) with $\hat{\sigma}_p^2$ in (15.4) is left as an end-of-chapter exercise.

The standard errors of $\hat{\mu}_p$ and $\hat{\sigma}_p^2$ can also be directly calculated from (15.6) and (15.7). Consider the calculation for $\text{se}(\hat{\mu}_p)$:

$$\text{var}(\hat{\mu}_p) = \text{var}(x_A \hat{\mu}_A + x_B \hat{\mu}_B) = x_A^2 \text{var}(\hat{\mu}_A) + x_B^2 \text{var}(\hat{\mu}_B) + 2x_A x_B \text{cov}(\hat{\mu}_A, \hat{\mu}_B). \quad (15.8)$$

Now, from chapter 7,

$$\text{var}(\hat{\mu}_A) = \frac{\sigma_A^2}{T}, \quad \text{var}(\hat{\mu}_B) = \frac{\sigma_B^2}{T}.$$

However, what is $\text{cov}(\hat{\mu}_A, \hat{\mu}_B)$? It can be shown that

$$\text{cov}(\hat{\mu}_A, \hat{\mu}_B) = \frac{\sigma_{AB}}{T}.$$

Then (15.8) can be rewritten as

$$\begin{aligned} \text{var}(\hat{\mu}_p) &= x_A \frac{\sigma_A^2}{T} + x_B \frac{\sigma_B^2}{T} + 2x_A x_B \frac{\sigma_{AB}}{T} \\ &= \frac{1}{T} (x_A^2 \sigma_A^2 + x_B^2 \sigma_B^2 + 2x_A x_B \sigma_{AB}) \\ &= \frac{\sigma_p^2}{T}. \end{aligned}$$

Hence,

$$\text{se}(\hat{\mu}_p) = \sqrt{\text{var}(\hat{\mu}_p)} = \frac{\sigma_p}{\sqrt{T}},$$

which is the same formula given in (15.5).

Unfortunately, the calculation of $\text{se}(\hat{\sigma}_p^2)$ is a bit horrific and we only sketch out some of the details. Now,

$$\begin{aligned} \text{var}(\hat{\sigma}_p^2) &= \text{var}(x_A^2 \hat{\sigma}_A^2 + x_B^2 \hat{\sigma}_B^2 + 2x_A x_B \hat{\sigma}_{AB}) \\ &= x_A^4 \text{var}(\hat{\sigma}_A^2) + x_B^4 \text{var}(\hat{\sigma}_B^2) + 4x_A^2 x_B^2 \text{var}(\hat{\sigma}_{AB}) \\ &\quad + 2x_A^2 x_B^2 \text{cov}(\hat{\sigma}_A^2, \hat{\sigma}_B^2) + 4x_A^3 x_B \text{cov}(\hat{\sigma}_A^2, \hat{\sigma}_{AB}) \\ &\quad + 4x_A x_B^3 \text{cov}(\hat{\sigma}_B^2, \hat{\sigma}_{AB}). \end{aligned} \quad (15.9)$$

Unfortunately, there are no easy formulas for $\text{var}(\hat{\sigma}_{AB})$, $\text{cov}(\hat{\sigma}_A^2, \hat{\sigma}_B^2)$, $\text{cov}(\hat{\sigma}_A^2, \hat{\sigma}_{AB})$ and $\text{cov}(\hat{\sigma}_B^2, \hat{\sigma}_{AB})$. With a CLT approximation and much tedious calculation it can be shown that (15.9) is approximately equal to the square of $\text{se}(\hat{\sigma}_p^2)$ given in (15.5). Obviously, the method of computing standard errors for $\hat{\mu}_p$ and $\hat{\sigma}_p^2$ directly from the CER model estimates computed from the sample portfolio returns is much easier than the indirect calculations based on the individual asset estimates.

Example 15.1. Example data for two risky asset portfolio

Table 11.1 gives annual return distribution parameters for two hypothetical assets *A* and *B*. We use this distribution but re-scale the values to represent a monthly return distribution.

Asset A is the high risk asset with a monthly return of $\mu_A = 1.46\%$ and monthly standard deviation of $\sigma_A = 7.45\%$. Asset B is a lower risk asset with monthly return $\mu_B = 0.458\%$ and monthly standard deviation of $\sigma_B = 3.32\%$. The assets are assumed to be slightly negatively correlated with correlation coefficient $\rho_{AB} = -0.164$. Given the standard deviations and the correlation, the covariance can be determined from $\sigma_{AB} = \rho_{AB}\sigma_A\sigma_B = -0.0004$. In R, these CER model parameters are created using:

```
mu.A = 0.175/12
sig.A = 0.258/sqrt(12)
sig2.A = sig.A^2
mu.B = 0.055/12
sig.B = 0.115/sqrt(12)
sig2.B = sig.B^2
rho.AB = -0.164
sig.AB = rho.AB * sig.A * sig.B
sigma.mat = matrix(c(sig2.A, sig.AB, sig.AB, sig2.B), 2, 2, byrow = TRUE)
mu.vec = c(mu.A, mu.B)
sd.vec = c(sig.A, sig.B)
names(mu.vec) = names(sd.vec) = c("asset.A", "asset.B")
dimnames(sigma.mat) = list(names(mu.vec), names(mu.vec))
```

For portfolio analysis, we consider an equally weighted portfolio of assets A and B . The CER model parameters for this portfolio are:

```
x1.A = 0.5
x1.B = 1 - x1.A
x1.vec = c(x1.A, x1.B)
names(x1.vec) = names(mu.vec)
mu.p1 = x1.A * mu.A + x1.B * mu.B
sig2.p1 = x1.A^2 * sig2.A + x1.B^2 * sig2.B + 2 * x1.A * x1.B * sig.AB
sig.p1 = sqrt(sig2.p1)
cbind(mu.p1, sig2.p1, sig.p1)

##      mu.p1 sig2.p1 sig.p1
## [1,] 0.00958 0.00146 0.0382
```

We use the above CER model parameters to simulate $T = 60$ hypothetical returns for assets A and B and the equally weighted portfolio:

```
library(mvtnorm)
n.obs = 60
set.seed(125)
returns.sim = rmvnorm(n.obs, mean = mu.vec, sigma = sigma.mat)
colnames(returns.sim) = names(mu.vec)
rp1.sim = returns.sim %*% x1.vec
```

The estimates of the CER model parameters from the simulated returns for the individual assets are:

```
muhat.vals = colMeans(returns.sim)
sig2hat.vals = apply(returns.sim, 2, var)
sigmahat.vals = apply(returns.sim, 2, sd)
covmat.hat = cov(returns.sim)
covhat = covmat.hat[1, 2]
```

```

rbind(muhat.vals, sig2hat.vals, sigmahat.vals)

##           asset.A asset.B
## muhat.vals 0.03077 0.00212
## sig2hat.vals 0.00539 0.00127
## sigmahat.vals 0.07344 0.03560

covhat

## [1] -0.000848

```

The estimated standard errors and 95% confidence intervals for $\hat{\mu}_i$ and $\hat{\sigma}_i$ ($i = A, B$) are:

```

# estimated standard errors
se.muhat = sigmahat.vals/sqrt(n.obs)
se.sigmahat = sigmahat.vals/sqrt(2 * n.obs)
rbind(se.muhat, se.sigmahat)

##           asset.A asset.B
## se.muhat   0.00948 0.00460
## se.sigmahat 0.00670 0.00325

# 95% confidence intervals
lower.mu = muhat.vals - 2 * se.muhat
upper.mu = muhat.vals + 2 * se.muhat
lower.sigma = sigmahat.vals - 2 * se.sigmahat
upper.sigma = sigmahat.vals + 2 * se.sigmahat
cbind(lower.mu, upper.mu)

##           lower.mu upper.mu
## asset.A    0.01181 0.0497
## asset.B   -0.00707 0.0113

cbind(lower.sigma, upper.sigma)

##           lower.sigma upper.sigma
## asset.A      0.0600     0.0869
## asset.B      0.0291     0.0421

```

Here, we see that the means are not estimated as precisely as the volatilities and that all of the true values are contained in the 95% confidence intervals.

For the equally weighted portfolio, the CER model estimates computed directly from the simulated portfolio returns using (15.4) are:

```

muhat.p1 = mean(rp1.sim)
sig2hat.p1 = as.numeric(var(rp1.sim))
sigmahat.p1 = sd(rp1.sim)
cbind(muhat.p1, sig2hat.p1, sigmahat.p1)

##           muhat.p1 sig2hat.p1 sigmahat.p1
## [1,] 0.0164    0.00124    0.0352

```

The portfolio estimates computed from the asset estimates (15.6) and (15.7) are numerically equivalent:

```

muhat.p1 = as.numeric(x1.A * muhat.vals[1] + x1.B * muhat.vals[2])
sig2hat.p1 = as.numeric(x1.A^2 * sig2hat.vals[1] + x1.B^2 * sig2hat.vals[2] + 2 *
    x1.A * x1.B * covhat)
sighat.p1 = sqrt(sig2hat.p1)
cbind(muhat.p1, sig2hat.p1, sigmahat.p1)

##      muhat.p1 sig2hat.p1 sigmahat.p1
## [1,] 0.0164    0.00124    0.0352

```

The estimated standard errors for $\hat{\mu}_p$ and $\hat{\sigma}_p$, and 95% confidence intervals for μ_p and σ_p computed using (15.5) are:

```

# estimated standard errors
se.muhat.p1 = sigmahat.p1/sqrt(n.obs)
se.sigmahat.p1 = sigmahat.p1/sqrt(2 * n.obs)
cbind(se.muhat.p1, se.sigmahat.p1)

##      se.muhat.p1 se.sigmahat.p1
## [1,] 0.00455     0.00322

# 95% confidence intervals
lower.mu = muhat.p1 - 2 * se.muhat.p1
upper.mu = muhat.p1 + 2 * se.muhat.p1
lower.sigma = sigmahat.p1 - 2 * se.sigmahat.p1
upper.sigma = sigmahat.p1 + 2 * se.sigmahat.p1
ans = rbind(c(lower.mu, upper.mu), c(lower.sigma, upper.sigma))
colnames(ans) = c("lower", "upper")
rownames(ans) = c("mu.p", "sigma.p")
ans

##      lower upper
## mu.p 0.00735 0.0255
## sigma.p 0.02880 0.0417

```

As with the individual assets, the portfolio mean is estimated less precisely than the portfolio volatility and the true values are in the 95% confidence intervals.

In our analysis of portfolios, we visualized the risk return trade-off between portfolios by plotting the volatility-expected return pairs $(\hat{\sigma}_p, \hat{\mu}_p)$ of different portfolios. We did this without considering estimation error in $\hat{\sigma}_p$ and $\hat{\mu}_p$. Estimation error in $\hat{\sigma}_p$ and $\hat{\mu}_p$ creates uncertainty about the location of the true values (σ_p, μ_p) in the risk-return diagram. ■

Example 15.2. True and estimated risk return diagram for example data.

Figure 15.1 shows the true values and sample estimates of the pairs (σ_A, μ_A) , (σ_B, μ_B) , and (σ_p, μ_p) for the equally weighted portfolio created using:

```

# true values
plot(sd.vec, mu.vec, pch = 16, col = "black", ylim = c(-0.01, 0.04), xlim = c(0,
    0.1), xlab = expression(sigma[p]), ylab = expression(mu[p]), cex = 2)
abline(h = 0)

```

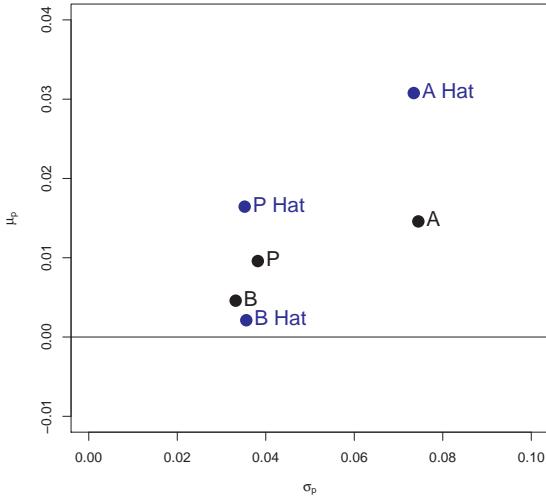


Fig. 15.1 Risk-return diagram for example data. $A = (\sigma_A, \mu_A)$, $B = (\sigma_B, \mu_B)$, $A \text{ Hat} = (\hat{\sigma}_A, \hat{\mu}_A)$, $B \text{ Hat} = (\hat{\sigma}_B, \hat{\mu}_B)$.

```

points(sig.p1, mu.p1, pch = 16, col = "black", cex = 2)
text(x = sig.A, y = mu.A, labels = "A", pos = 4, cex = 1.5)
text(x = sig.B, y = mu.B, labels = "B", pos = 4, cex = 1.5)
text(x = sig.p1, y = mu.p1, labels = "P", pos = 4, cex = 1.5)
# estimates
points(sigmahat.vals, muhat.vals, pch = 16, col = "blue", cex = 2)
points(sigmahat.p1, muhat.p1, pch = 16, col = "blue", cex = 2)
text(x = sigmahat.vals[1], y = muhat.vals[1], labels = "A Hat", col = "blue", pos = 4,
     cex = 1.5)
text(x = sigmahat.vals[2], y = muhat.vals[2], labels = "B Hat", col = "blue", pos = 4,
     cex = 1.5)
text(x = sigmahat.p1, y = muhat.p1, labels = "P Hat", col = "blue", pos = 4, cex = 1.5)

```

We see that $(\hat{\sigma}_B, \hat{\mu}_B)$ is fairly close to (σ_B, μ_B) but that $(\hat{\sigma}_A, \hat{\mu}_A)$ is quite far above (σ_A, μ_A) and $(\hat{\sigma}_P, \hat{\mu}_P)$ is moderately far above (σ_P, μ_P) . The large positive estimation errors in $\hat{\mu}_P$ and $\hat{\mu}_A$ greatly overstate the risk-return characteristics of the equally weighted portfolio and asset A.

■

To illustrate estimation error in the risk return diagram, the individual 95% confidence intervals for μ_i and σ_i could be superimposed on the plot as rectangles centered at the pairs $(\hat{\sigma}_i, \hat{\mu}_i)$. However, the probability that these rectangles contain the true pairs (σ_i, μ_i) is not equal to 95% because the confidence intervals for μ_i are created independently from the confidence intervals for σ_i . To create a joint confidence set that cover the pair (σ_i, μ_i) with probability 95% requires knowing the joint probability distribution of $(\hat{\sigma}_i, \hat{\mu}_i)$. In chapter 7, it was shown that in the CER model:

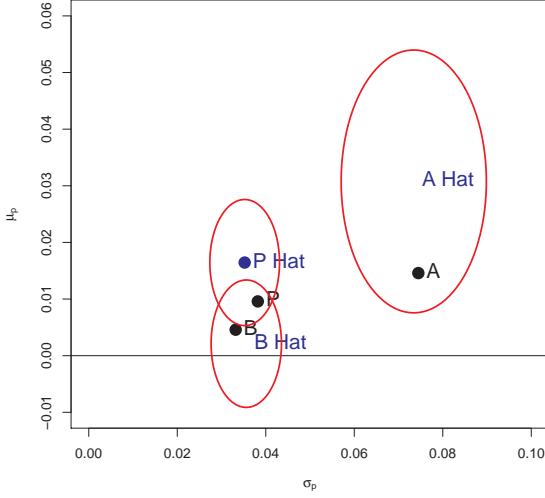


Fig. 15.2 Risk return tradeoff with 95% confidence ellipses

$$\begin{pmatrix} \hat{\sigma}_i \\ \hat{\mu}_i \end{pmatrix} \sim N \left(\begin{pmatrix} \sigma_i \\ \mu_i \end{pmatrix}, \begin{pmatrix} \text{se}(\hat{\sigma}_i)^2 & 0 \\ 0 & \text{se}(\hat{\mu}_i)^2 \end{pmatrix} \right), \quad (15.10)$$

for large enough T . Hence, $\hat{\sigma}_i$ and $\hat{\mu}_i$ are (asymptotically) jointly normally distributed and $\text{cov}(\hat{\sigma}_i, \hat{\mu}_i) = 0$ which implies that they are also independent. Let $\theta = (\sigma_i, \mu_i)', \hat{\theta} = (\hat{\sigma}_i, \hat{\mu}_i)'$ and $\mathbf{V} = \text{diag}(\text{se}(\hat{\sigma}_i)^2, \text{se}(\hat{\mu}_i)^2)$. Then (15.10) can be expressed as $\hat{\theta} \sim N(\theta, \mathbf{V})$. It follows that the quadratic form $(\hat{\theta} - \theta)' \mathbf{V}^{-1} (\hat{\theta} - \theta) \sim \chi^2(2)$, and so

$$Pr \left\{ (\hat{\theta} - \theta)' \mathbf{V}^{-1} (\hat{\theta} - \theta) \leq q_{.95}^{\chi^2(2)} \right\} = 0.95 \quad (15.11)$$

where $q_{.95}^{\chi^2(2)}$ is the 95% quantile of the $\chi^2(2)$ distribution. The equation $(\hat{\theta} - \theta)' \mathbf{V}^{-1} (\hat{\theta} - \theta) = q_{.95}^{\chi^2(2)}$ defines an un-tilted ellipse in $\sigma_i - \mu_i$ space centered at $(\hat{\sigma}_i, \hat{\mu}_i)$ with axes proportional to $\text{se}(\hat{\sigma}_i)$ and $\text{se}(\hat{\mu}_i)$, respectively. This ellipse is the joint 95% confidence set for (σ_i, μ_i) .

Example 15.3. True and estimated risk return diagram with confidence ellipses for example data.

Figure 15.2 repeats Figure 15.1 with the addition of the joint confidence sets for (σ_A, μ_A) , (σ_B, μ_B) , and (σ_p, μ_p) , created using

```
library(ellipse)
plot(sd.vec, mu.vec, pch = 16, col = "black", ylim = c(-0.01, 0.06), xlim = c(0,
  0.1), xlab = expression(sigma[p]), ylab = expression(mu[p]), cex = 2)
abline(h = 0)
```

```

points(sig.p1, mu.p1, pch = 16, col = "black", cex = 2)
text(x = sig.A, y = mu.A, labels = "A", pos = 4, cex = 1.5)
text(x = sig.B, y = mu.B, labels = "B", pos = 4, cex = 1.5)
text(x = sig.p1, y = mu.p1, labels = "P", pos = 4, cex = 1.5)
# estimates points(sigmahat.vals, muhat.vals, pch=16, col='blue', cex=2)
points(sigmahat.p1, muhat.p1, pch = 16, col = "blue", cex = 2)
text(x = sigmahat.vals[1], y = muhat.vals[1], labels = "A Hat", col = "blue", pos = 4,
      cex = 1.5)
text(x = sigmahat.vals[2], y = muhat.vals[2], labels = "B Hat", col = "blue", pos = 4,
      cex = 1.5)
text(x = sigmahat.p1, y = muhat.p1, labels = "P Hat", col = "blue", pos = 4, cex = 1.5)
# Create asymptotic variances
V.A = matrix(c(se.sigmahat[1]^2, 0, 0, se.muhat[1]^2), 2, 2, byrow = TRUE)
V.B = matrix(c(se.sigmahat[2]^2, 0, 0, se.muhat[2]^2), 2, 2, byrow = TRUE)
V.P = matrix(c(se.sigmahat.p1^2, 0, 0, se.muhat.p1^2), 2, 2, byrow = TRUE)
# plot confidence ellipses
lines(ellipse(V.A, centre = c(sigmahat.vals[1], muhat.vals[1]), level = 0.95),
      col = "red", lwd = 2)
lines(ellipse(V.B, centre = c(sigmahat.vals[2], muhat.vals[2]), level = 0.95),
      col = "red", lwd = 2)
lines(ellipse(V.P, centre = c(sigmahat.p1, muhat.p1), level = 0.95), col = "red",
      lwd = 2)

```

The `ellipse()` function from the R package `ellipse` is used to draw the 95% confidence sets. The confidence ellipses are longer in the μ_i direction because $se(\hat{\mu}_i)$ is larger than $se(\hat{\sigma}_i)$ and indicate more uncertainty about the true values of μ_i than the true values of σ_i . The ellipse for (σ_A, μ_A) is much bigger than the ellipses for (σ_B, μ_B) and (σ_p, μ_p) and indicates a wide range of possible values for (σ_A, μ_A) . For all pairs, the true values are inside the 95% confidence ellipses.²

The estimation errors in the points $(\hat{\sigma}_i, \hat{\mu}_i)$ can also be illustrated using the bootstrap. We simply sample with replacement from the observed returns B times and compute estimates of (σ_i, μ_i) from each bootstrap sample. We can then plot the B bootstrap pairs $(\hat{\sigma}_i^*, \hat{\mu}_i^*)$ on the risk return diagram instead of the confidence ellipses.

Example 15.4. True and estimated risk return diagram with bootstrap estimates for example data.

The following R code creates $B = 500$ bootstrap estimates of (σ_i, μ_i) for $i = A, B, P$:

```

n.boot = 500
mu.boot = matrix(0, n.boot, 3)
sd.boot = matrix(0, n.boot, 3)
colnames(mu.boot) = colnames(sd.boot) = c("A", "B", "P")
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace = TRUE)

```

² Notice, however, that the pair (σ_p, μ_p) is inside the confidence ellipse for (σ_B, μ_B) but that (σ_B, μ_B) is not in the confidence ellipse for (σ_p, μ_p) .

```

ret.boot = returns.sim[boot.idx, ]
rp1.boot = ret.boot %*% x1.vec
ret.boot = cbind(ret.boot, rp1.boot)
mu.boot[i, ] = colMeans(ret.boot)
sd.boot[i, ] = apply(ret.boot, 2, sd)
}

```

Figure 15.3 repeats Figure 15.2 and adds the bootstrap estimates of (σ_i, μ_i) for $i = A, B, P$. Notice that the bootstrap estimates for each asset produce a scatter that fills the 95% confidence ellipses with just a few estimates lying outside the ellipses. Hence, the bootstrap is an easy and effective way to illustrate estimation error in the risk-return diagram.

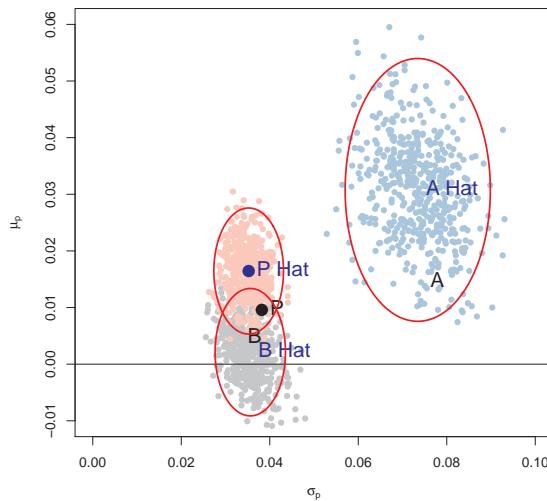


Fig. 15.3 Risk return trade off with bootstrap estimates.

15.1.1 Estimation error in the portfolio frontier

In the case of two risky assets, the portfolio frontier is a plot of the risk-return characteristics of all feasible portfolios. From this frontier, we can readily determine the set of efficient portfolios which are those portfolios that have the highest expected return for a given risk level. If we construct the portfolio frontier from portfolio risk and return estimates from the CER model, then the estimation error in these estimates leads to estimation error in the entire portfolio frontier.

Example 15.5. True and estimated portfolio frontier for example data.

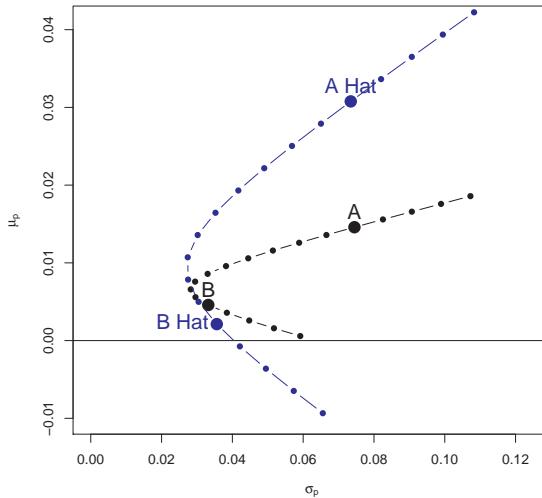


Fig. 15.4 True and estimated portfolio frontiers

Using the example data, we can construct the true unobservable portfolio frontier as well as an estimate of this frontier from the simulated returns. For example, consider creating the true and estimated portfolio frontiers for the following portfolios:

```
x.A = seq(from = -0.4, to = 1.4, by = 0.1)
x.B = 1 - x.A
```

The risk-return characteristics of the portfolios on the true portfolio frontier are:

```
mu.p = x.A * mu.A + x.B * mu.B
sig2.p = x.A^2 * sig2.A + x.B^2 * sig2.B + 2 * x.A * x.B * sig.AB
sig.p = sqrt(sig2.p)
```

The risk-return characteristics of the portfolios on the estimated frontier are:

```
muhat.p = x.A * muhat.vals["asset.A"] + x.B * muhat.vals["asset.B"]
sig2hat.p = x.A^2 * sig2hat.vals["asset.A"] + x.B^2 * sig2hat.vals["asset.B"] +
  2 * x.A * x.B * covhat
sighat.p = sqrt(sig2hat.p)
```

The true and estimated portfolios are illustrated in Figure 15.4, created with

```
plot(sig.p, mu.p, pch = 16, type = "b", col = "black", ylim = c(-0.01, 0.041),
  xlim = c(0, 0.125), xlab = expression(sigma[p]), ylab = expression(mu[p]),
  cex = 1)
abline(h = 0)
points(sd.vec, mu.vec, pch = 16, col = "black", cex = 2)
text(x = sig.A, y = mu.A, labels = "A", pos = 3, cex = 1.5)
text(x = sig.B, y = mu.B, labels = "B", pos = 3, cex = 1.5)
# estimated assets and frontier
```

```

points(sighat.p, muhat.p, type = "b", pch = 16, col = "blue", cex = 1)
points(sigmahat.vals, muhat.vals, pch = 16, col = "blue", cex = 2)
text(x = sigmahat.vals[1], y = muhat.vals[1], labels = "A Hat", col = "blue", pos = 3,
      cex = 1.5)
text(x = sigmahat.vals[2], y = muhat.vals[2], labels = "B Hat", col = "blue", pos = 2,
      cex = 1.5)

```

Due to the large estimation error in $\hat{\mu}_A$, the estimated frontier is considerably higher than the true frontier. As a result, portfolios on the estimated frontier appear to have higher reward-to-risk properties than they actually do.

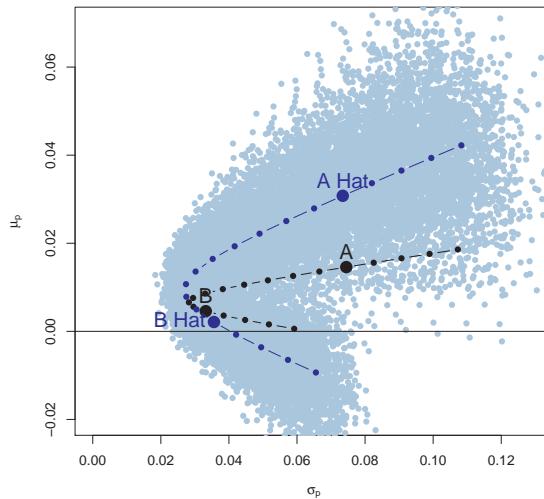


Fig. 15.5 Estimated frontier with bootstrap estimates.

Sampling uncertainty about the estimated frontier can be easily computed using the bootstrap. The process is the same as in the case of a single asset. For each bootstrap sample, estimate the expected return and volatility of each portfolio on the frontier and then plot these bootstrap pairs on the plot showing the estimated frontier from the sample returns.

Example 15.6. True and estimated risk portfolio frontier with bootstrap estimates for example data.

The R code to create $B = 1000$ bootstrap estimates of risk and return pairs for the frontier portfolios is

```

# initialize matrices
n.boot = 1000
mu.boot = matrix(0, n.boot, length(x.A))
sd.boot = matrix(0, n.boot, length(x.A))

```

```

colnames(mu.boot) = colnames(sd.boot) = paste("P", 1:length(x.A), sep = ".")
# bootstrap loop
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace = TRUE)
  ret.boot = returns.sim[boot.idx, ]
  # CER model estimates
  muhat.boot = colMeans(ret.boot)
  sig2hat.boot = apply(ret.boot, 2, var)
  sigmahat.boot = sqrt(sig2hat.boot)
  covhat.boot = cov(ret.boot)[1, 2]
  # portfolio risk return estimates
  mu.boot[i, ] = x.A * muhat.boot[1] + x.B * muhat.boot[2]
  sig2.boot = x.A^2 * sig2hat.boot[1] + x.B^2 * sig2hat.boot[2] + 2 * x.A * x.B *
    covhat.boot
  sd.boot[i, ] = sqrt(sig2.boot)
}

```

The true and estimated frontier together with the bootstrap risk-return pairs for each portfolio on the estimated frontier is illustrated in Figure 15.5, create using

```

# set up plot area
plot(sig.p, mu.p, type = "n", ylim = c(-0.02, 0.07), xlim = c(0, 0.13), xlab = expression(sigma[p]),
     ylab = expression(mu[p]))
# plot bootstrap estimates
for (i in 1:length(x.A)) {
  points(sd.boot[, i], mu.boot[, i], pch = 16, col = "lightblue")
}
# plot true frontier
points(sig.p, mu.p, pch = 16, type = "b", col = "black", cex = 1)
abline(h = 0)
# plot true assets
points(sd.vec, mu.vec, pch = 16, col = "black", cex = 2)
text(x = sig.A, y = mu.A, labels = "A", pos = 3, cex = 1.5)
text(x = sig.B, y = mu.B, labels = "B", pos = 3, cex = 1.5)
# plot estimated frontier
points(sighat.p, muhat.p, type = "b", pch = 16, col = "blue", cex = 1)
# plot estimated assets
points(sigmahat.vals, muhat.vals, pch = 16, col = "blue", cex = 2)
text(x = sigmahat.vals[1], y = muhat.vals[1], labels = "A Hat", col = "blue", pos = 3,
     cex = 1.5)
text(x = sigmahat.vals[2], y = muhat.vals[2], labels = "B Hat", col = "blue", pos = 2,
     cex = 1.5)

```

The true frontier portfolios are the black dots, the estimated frontier portfolios are the dark blue dots and the bootstrap risk-return pairs are the light blue dots. The bootstrap estimates form a light blue cloud around the estimated frontier. The bootstrap cloud can be interpreted as approximating a confidence interval around estimated frontier. From (15.11), this confidence interval can be thought of the union of all of the confidence ellipses about the portfolios on the frontier. The true frontier of portfolios (black dots) is within the blue cloud.



15.1.2 Statistical properties of the global minimum variance portfolio

For portfolios of two risky assets, the global minimum variance portfolio weights satisfy

$$x_A^{\min} = \frac{\sigma_B^2 - \sigma_{AB}}{\sigma_A^2 + \sigma_B^2 - 2\sigma_{AB}}, \quad x_B^{\min} = 1 - x_A^{\min} = \frac{\sigma_A^2 - \sigma_{AB}}{\sigma_A^2 + \sigma_B^2 - 2\sigma_{AB}}. \quad (15.12)$$

The expected return and variance of the global minimum variance portfolio are

$$\begin{aligned} \mu_{p,\min} &= x_A^{\min}\mu_A + x_B^{\min}\mu_B, \\ \sigma_{p,\min}^2 &= (x_A^{\min})^2\sigma_A^2 + (x_B^{\min})^2\sigma_B^2 + 2x_A^{\min}x_B^{\min}\sigma_{AB}. \end{aligned}$$

The estimated global minimum variance portfolio weights are then

$$\hat{x}_A^{\min} = \frac{\hat{\sigma}_B^2 - \hat{\sigma}_{AB}}{\hat{\sigma}_A^2 + \hat{\sigma}_B^2 - 2\hat{\sigma}_{AB}}, \quad \hat{x}_B^{\min} = 1 - \hat{x}_A^{\min} = \frac{\hat{\sigma}_A^2 - \hat{\sigma}_{AB}}{\hat{\sigma}_A^2 + \hat{\sigma}_B^2 - 2\hat{\sigma}_{AB}}. \quad (15.13)$$

From (15.13) we see that estimation error in the global minimum variance weights is related to estimation error in the asset variances and the covariance in a complicated nonlinear way. However, the estimation error does not depend on estimation error in the expected returns. The estimated expected returns and variance of the global minimum variance portfolio are

$$\hat{\mu}_{p,\min} = \hat{x}_A^{\min}\hat{\mu}_A + \hat{x}_B^{\min}\hat{\mu}_B, \quad (15.14)$$

$$\hat{\sigma}_{p,\min}^2 = (\hat{x}_A^{\min})^2\hat{\sigma}_A^2 + (\hat{x}_B^{\min})^2\hat{\sigma}_B^2 + 2\hat{x}_A^{\min}\hat{x}_B^{\min}\hat{\sigma}_{AB} \quad (15.15)$$

Here, estimation error in $\hat{\mu}_{p,\min}$ depends on two sources: estimation error in the global minimum variance weights (which depend on estimation error in the asset variances and the covariance), and estimation error in the asset expected returns. However, estimation error in $\hat{\sigma}_{p,\min}^2$ only depends on estimation error in the asset variances and the covariance. Because there is more estimation error in the asset expected returns than the asset variances $\hat{\mu}_{p,\min}$, will be estimated more imprecisely than $\hat{\sigma}_{p,\min}^2$.

Example 15.7. True and estimated global minimum variance portfolio for example data.

For the example data, the true and estimated global minimum variance portfolio are:

```
# True global min variance portfolio weights
xA.min = (sig2.B - sig.AB)/(sig2.A + sig2.B - 2 * sig.AB)
xB.min = 1 - xA.min
c(xA.min, xB.min)

## [1] 0.202 0.798

# Estimated min variance portfolio weights
xA.min.hat = (sig2hat.vals[2] - covhat)/(sig2hat.vals[1] + sig2hat.vals[2] - 2 *
covhat)
xB.min.hat = 1 - xA.min.hat
c(xA.min.hat, xB.min.hat)
```

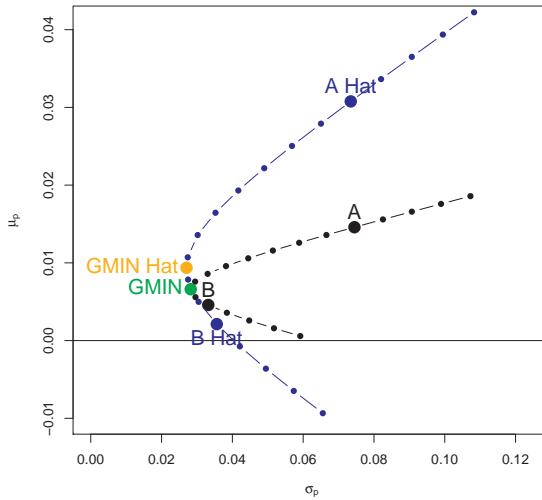


Fig. 15.6 True and estimated global minimum variance portfolios.

```
## asset.B asset.B
##   0.253   0.747
```

The estimated weights are close to the true weights. The true and estimated expected return and volatility of the global minimum variance portfolio are:

```
# Expected return and volatility of true global minimum variance portfolio
mu.p.min = xA.min * mu.A + xB.min * mu.B
sig2.p.min = xA.min^2 * sig2.A + xB.min^2 * sig2.B + 2 * xA.min * xB.min * sig.AB
sig.p.min = sqrt(sig2.p.min)
c(mu.p.min, sig.p.min)

## [1] 0.0066 0.0282

# Expected return and volatility of estimated global minimum variance portfolio
mu.p.min.hat = as.numeric(xA.min.hat * muhat.vals[1] + xB.min.hat * muhat.vals[2])
sig2.p.min.hat = xA.min.hat^2 * sig2hat.vals[1] + xB.min.hat^2 * sig2hat.vals[2] +
  2 * xA.min.hat * xB.min.hat * covhat
sig.p.min.hat = as.numeric(sqrt(sig2.p.min.hat))
c(mu.p.min.hat, sig.p.min.hat)

## [1] 0.00937 0.02706
```

The estimated volatility of the global minimum variance portfolio is close to the true volatility but the estimated expected return is much larger than the true expected return. These portfolios are illustrated in Figure 15.6.

The statistical properties of (15.13), (15.14) and (15.15) are difficult to derive analytically. For example, suppose we would like to evaluate the bias in the global minimum variance weights. We would need to evaluate

$$E[\hat{x}_A^{\min}] = E\left[\frac{\hat{\sigma}_B^2 - \hat{\sigma}_{AB}}{\hat{\sigma}_A^2 + \hat{\sigma}_B^2 - 2\hat{\sigma}_{AB}}\right]. \quad (15.16)$$

Now,

$$E\left[\frac{\hat{\sigma}_B^2 - \hat{\sigma}_{AB}}{\hat{\sigma}_A^2 + \hat{\sigma}_B^2 - 2\hat{\sigma}_{AB}}\right] \neq \frac{E[\hat{\sigma}_B^2] - E[\hat{\sigma}_{AB}]}{E[\hat{\sigma}_A^2] + E[\hat{\sigma}_B^2] - 2E[\hat{\sigma}_{AB}]}$$

because $E[g(X)] \neq g(E[X])$ for nonlinear functions $g(\cdot)$. Hence, (15.16) is extremely difficult to evaluate analytically. Similarly, suppose we would like to compute $\text{se}(\hat{x}_A^{\min})$. We would need to compute

$$\text{var}(\hat{x}_A^{\min}) = \text{var}\left(\frac{\hat{\sigma}_B^2 - \hat{\sigma}_{AB}}{\hat{\sigma}_A^2 + \hat{\sigma}_B^2 - 2\hat{\sigma}_{AB}}\right),$$

which is a difficult and tedious calculation and can only be approximated based on the CLT.

The computations are even more difficult for evaluating bias and computing standard errors for (15.14) and (15.15). For example, to evaluate the bias of $\hat{\mu}_{p,\min}$ we need to calculate

$$E[\hat{\mu}_{p,\min}] = E[\hat{x}_A^{\min}\hat{\mu}_A + \hat{x}_B^{\min}\hat{\mu}_B] = E[\hat{x}_A^{\min}\hat{\mu}_A] + E[\hat{x}_B^{\min}\hat{\mu}_B]. \quad (15.17)$$

Without knowing more about the joint distributions of the asset means and weights we cannot simplify (15.17). To compute the standard error of $\hat{\mu}_{p,\min}$ we need to calculate

$$\text{var}(\hat{\mu}_{p,\min}) = \text{var}(\hat{x}_A^{\min}\hat{\mu}_A + \hat{x}_B^{\min}\hat{\mu}_B) = \text{var}(\hat{x}_A^{\min}\hat{\mu}_A) + \text{var}(\hat{x}_B^{\min}\hat{\mu}_B) + 2\text{cov}(\hat{x}_A^{\min}\hat{\mu}_A, \hat{x}_B^{\min}\hat{\mu}_B).$$

Again, without knowing more about the joint distributions of the asset means and estimated weights we cannot simplify (15.17).

Fortunately, statistical properties of (15.13), (15.14) and (15.15) can be easily quantified using the bootstrap. For each bootstrap sample we calculate these statisticss and from the bootstrap distributions we can then evaluate bias and compute standard errors and confidence intervals.

Example 15.8. True and estimated global minimum variance portfolio with bootstrap estimates for example data.

To compute $B = 1000$ bootstrap estimates of (15.13), (15.14) and (15.15) use:

```
# initialize matrices
n.boot = 1000
weights.boot = matrix(0, n.boot, 2)
stats.boot = matrix(0, n.boot, 2)
colnames(weights.boot) = names(mu.vec)
colnames(stats.boot) = c("mu", "sigma")
# bootstrap loop
set.seed(123)
```

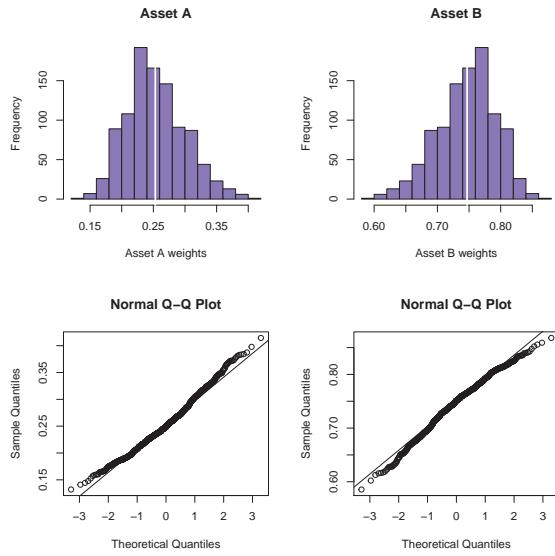


Fig. 15.7 Empirical distribution of bootstrap estimates of global minimum variance portfolio weights.

```

for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace = TRUE)
  ret.boot = returns.sim[boot.idx, ]
  # CER model estimates
  muhat.boot = colMeans(ret.boot)
  sig2hat.boot = apply(ret.boot, 2, var)
  sigmahat.boot = sqrt(sig2hat.boot)
  covhat.boot = cov(ret.boot)[1, 2]
  # global minimum variance portfolio weights
  weights.boot[i, 1] = (sig2hat.boot[2] - covhat.boot)/(sig2hat.boot[1] + sig2hat.boot[2] -
    2 * covhat.boot)
  weights.boot[i, 2] = 1 - weights.boot[i, 1]
  # portfolio risk return estimates
  stats.boot[i, "mu"] = weights.boot[i, 1] * muhat.boot[1] + weights.boot[i,
    2] * muhat.boot[2]
  sig2.boot = (weights.boot[i, 1]^2 * sig2hat.boot[1] + weights.boot[i, 2]^2 *
    sig2hat.boot[2] + 2 * weights.boot[i, 1] * weights.boot[i, 2] * covhat.boot)
  stats.boot[i, "sigma"] = sqrt(sig2.boot)
}

```

The bootstrap bias estimates for the global minimum variance portfolio weights are:

```

colMeans(weights.boot) - c(xA.min.hat, xB.min.hat)

##   asset.A   asset.B
##  0.000747 -0.000747

```

These values are close to zero suggesting that the estimated weights are unbiased. Notice that the estimates are identical but opposite in sign. This arises because the bootstrap estimates are perfectly negatively correlated:

```
cor(weights.boot)[1, 2]

## [1] -1
```

The bootstrap standard error estimates for the weights are:

```
apply(weights.boot, 2, sd)

## asset.A asset.B
## 0.0462 0.0462
```

The standard errors estimates are identical and close to 0.05 which is not too large. Figure 15.7 shows histograms and normal QQ-plots for the bootstrap estimates of the weights. These distributions are centered at the sample estimates (white vertical lines) and look slightly asymmetric.

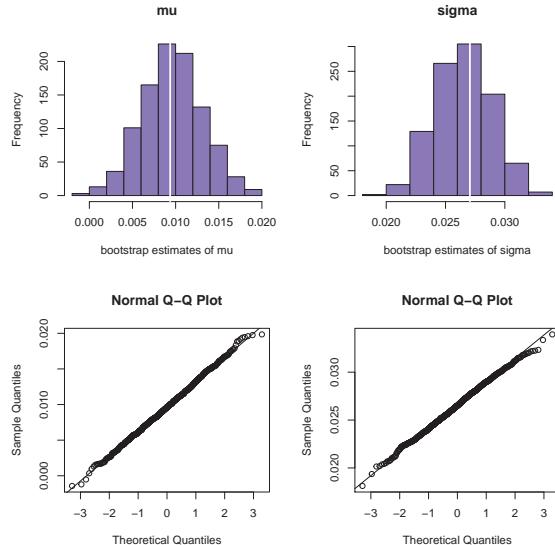


Fig. 15.8 Empirical distribution of bootstrap estimates of $\mu_{p,\min}$ and $\sigma_{p,\min}$.

The bootstrap estimates of bias for the estimated expected return and volatility of the global minimum variance portfolio are:

```
colMeans(stats.boot) - c(mu.p.min.hat, sig.p.min.hat)

##          mu      sigma
## 0.000280 -0.000528
```

These small values indicate that the estimates are roughly unbiased. The bootstrap standard error estimates are:

```
apply(stats.boot, 2, sd)

##      mu    sigma
## 0.00351 0.00238
```

Here, the bootstrap standard error estimate for $\hat{\mu}_{p,\min}$ is larger than the estimate for $\hat{\sigma}_{p,\min}$ indicating more uncertainty about the mean of the global minimum variance portfolio than the volatility of the portfolio. Interestingly, the bootstrap standard error estimates are close to the naive analytical standard error formulas based on the CLT that ignore estimation error in the weights:

```
sig.p.min.hat/sqrt(n.obs)

## [1] 0.00349

sig.p.min.hat/sqrt(2 * n.obs)

## [1] 0.00247
```

That is,

$$\text{se}_{\text{boot}}(\hat{\mu}_{p,\min}) \approx \frac{\hat{\sigma}_{p,\min}}{\sqrt{T}}, \quad \text{se}_{\text{boot}}(\hat{\sigma}_{p,\min}) \approx \frac{\hat{\sigma}_{p,\min}}{\sqrt{2 \cdot T}}.$$

The histograms and normal QQ-plots of the bootstrap estimates of $\mu_{p,\min}$ and $\sigma_{p,\min}$ are illustrated in Figure 15.8, and look like normal distributions.

The sampling uncertainty in $\hat{\mu}_{p,\min}$ and $\hat{\sigma}_{p,\min}$ can be visualized by plotting the bootstrap estimates of $\mu_{p,\min}$ and $\sigma_{p,\min}$ on the risk return diagram, as shown in Figure 15.9. Clearly there is much more uncertainty about the location of $\mu_{p,\min}$ than the location of $\sigma_{p,\min}$. ■

To sum up, for the global minimum variance portfolio we have roughly unbiased estimates of the weights, expected return and variance. We have a fairly precise estimate of volatility but an imprecise estimate of expected return.

15.1.3 Statistical properties of the Sharpe ratio and the tangency portfolio

Let r_f denote the rate of return on a risk free asset with maturity equal to the investment horizon of the return R_i on risky asset i . Recall from chapter 11, the Sharpe ratio of asset i is defined as

$$\text{SR}_i = \frac{\mu_i - r_f}{\sigma_i},$$

and is a measure of risk-adjusted expected return. Graphically in the risk-return diagram, SR_i is the slope of a straight line from the risk-free rate that passes through the point (σ_i, μ_i) and represents the risk-return tradeoff of portfolios of the risk-free asset and the risky asset. The estimated Sharpe ratio is

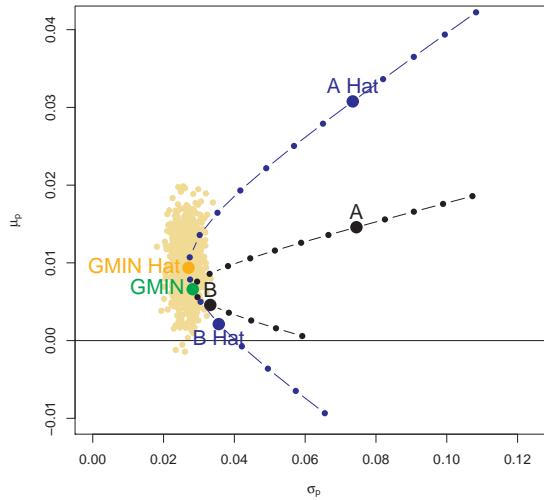


Fig. 15.9 Estimation error in estimates of the global minimum variance portfolio.

$$\widehat{SR}_i = \frac{\widehat{\mu}_i - r_f}{\widehat{\sigma}_i},$$

which inherits estimation error from $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ in a nonlinear way. As a result, analytically computing the bias and standard error for \widehat{SR}_i is complicated and typically involves approximations based on the CLT. However, computing numerical estimates of the bias and the standard error for \widehat{SR}_i using the bootstrap is simple.

Example 15.9. True and estimated Sharpe ratios for example data.

For the example data, assume a monthly risk-free rate of $r_f = 0.03/12 = 0.0025$. The true Sharpe ratios are:

```
r.f = 0.03/12
SR.true = (mu.vec - r.f)/sd.vec
SR.true

## asset.A asset.B
## 0.1622 0.0628
```

Here, asset A has a much higher Sharpe ratio than asset B. For the simulated data, the estimated Sharpe ratios are:

```
SR.hat = (muhat.vals - r.f)/sigmahat.vals
SR.hat

## asset.A asset.B
## 0.3849 -0.0107
```

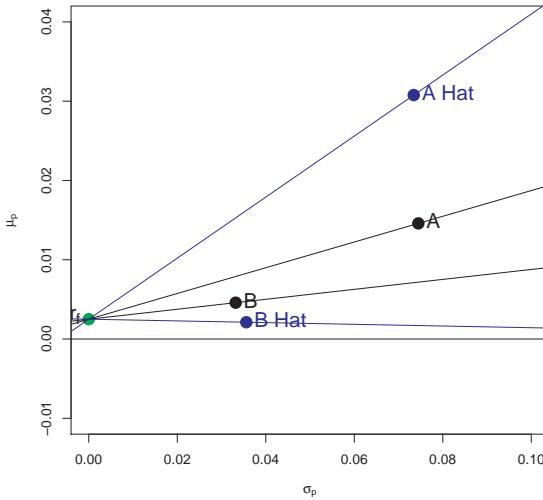


Fig. 15.10 True and estimated Sharpe ratios.

For both assets the estimated Sharpe ratios are quite different from the true Sharpe ratios and are highly misleading. The estimated Sharpe ratio for asset A is more than two times larger than the true Sharpe ratio and indicates a much higher risk adjusted performance than is actually available. For asset B, the estimated Sharpe ratio is slightly negative when, in fact, the true Sharpe ratio is slightly positive. The true and estimated Sharpe ratios are illustrated in Figure 15.10. The slopes of the black lines are the true Sharpe ratios, and the slopes of the blue lines are the estimated Sharpe ratios.

Example 15.10. True and estimated Sharpe ratios for example data.

To illustrate the magnitude of the estimation errors in the estimated Sharpe ratios, Figure 15.11 shows 500 bootstrap estimates of the risk-return pairs (σ_A, μ_A) and (σ_B, μ_B) along with the maximum and minimum bootstrap Sharpe ratio estimates for each asset. The bootstrap Sharpe ratio estimates are computed using:

```
n.boot = 500
mu.boot = matrix(0, n.boot, 2)
sd.boot = matrix(0, n.boot, 2)
SR.boot = matrix(0, n.boot, 2)
colnames(mu.boot) = colnames(sd.boot) = colnames(SR.boot) = c("A", "B")
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace = TRUE)
  ret.boot = returns.sim[boot.idx, ]
  mu.boot[i, ] = colMeans(ret.boot)
  sd.boot[i, ] = apply(ret.boot, 2, sd)
  SR.boot[i, ] = (mu.boot[i] - mu.mean) / sd.boot[i]
}
```

```

SR.boot[i, ] = (mu.boot[i, ] - r.f)/sd.boot[i, ]
}

# find index of max and min bootstrap Sharpe ratios
maxSRidx.A = which(SR.boot[, 1] == max(SR.boot[, 1]))
minSRidx.A = which(SR.boot[, 1] == min(SR.boot[, 1]))
maxSRidx.B = which(SR.boot[, 2] == max(SR.boot[, 2]))
minSRidx.B = which(SR.boot[, 2] == min(SR.boot[, 2]))

```

The maximum and minimum bootstrap Sharpe ratio estimates for asset B are the two grey straight lines from the risk-free rate that have the highest and lowest slopes, respectively. The maximum and minimum bootstrap Sharpe ratio estimates for asset A are the two light blue lines with the highest and lowest slopes, respectively. These maximum and minimum estimates are:

```

apply(SR.boot, 2, max)

##      A      B
## 0.914 0.376

apply(SR.boot, 2, min)

##      A      B
## 0.0598 -0.3600

```

For each asset the difference in the maximum and minimum slopes is substantial and indicates a large about of estimation error in the estimated Sharpe ratios.

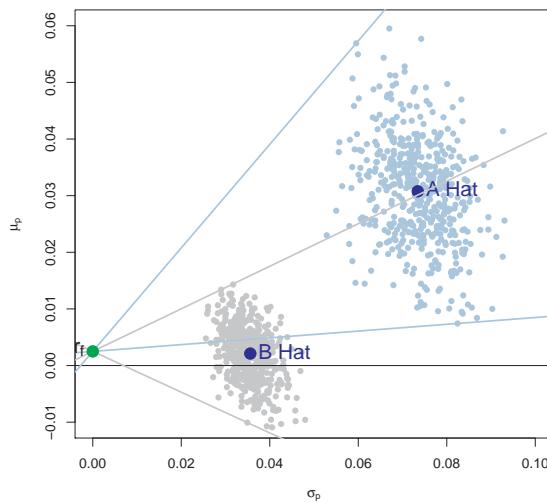


Fig. 15.11 Bootstrap risk-return estimates with maximum and minimum Sharpe ratios.

The statistical properties of the estimated Sharpe ratios can be estimated from the bootstrap samples. The bootstrap estimates of bias are:

```
# bias
biasSR.boot = colMeans(SR.boot) - c(SR.true)
biasSR.boot

##      A      B
## 0.2291 -0.0661

# bias relative to true values
biasSR.boot/SR.true

##      A      B
## 1.41 -1.05
```

Here we see there is a substantial upward bias in \widehat{SR}_A (bias relative to true Sharpe ratio is 141%) and a substantial downward bias in \widehat{SR}_B (bias relative to true Sharpe ratio is -105%). A crude bias adjusted Sharpe ratio estimate, which subtracts the estimated bias from the sample estimate, gives results much closer to the true Sharpe ratios:

```
SR.hat.biasAdjust = SR.hat - biasSR.boot
SR.hat.biasAdjust

## asset.A asset.B
## 0.1558 0.0554

SR.true

## asset.A asset.B
## 0.1622 0.0628
```

The bootstrap estimated standard errors for both assets are big and indicate that the Sharpe ratios are not estimated well:

```
apply(SR.boot, 2, sd)

##      A      B
## 0.144 0.135
```

Figure 15.12 shows the histograms and normal QQ-plots of the bootstrap distributions for the estimated Sharpe ratios. The histograms for both assets show the wide dispersion of the bootstrap Sharpe ratio estimates that was illustrated in Figure 15.11. The histogram and QQ-plot for asset A shows a positive skewness whereas the histogram and QQ-plot for asset B show a more symmetric distribution. The quantile-based 95% confidence intervals for the true Sharpe ratios are:

```
# 95% CI for SR.A
quantile(SR.boot[, 1], probs = c(0.025, 0.975))

## 2.5% 97.5%
## 0.113 0.707

# 95% CI for SR.B
quantile(SR.boot[, 2], probs = c(0.025, 0.975))
```

```
##   2.5%  97.5%
## -0.257  0.274
```

These intervals are quite large and indicate much uncertainty about the true Sharpe ratio values.

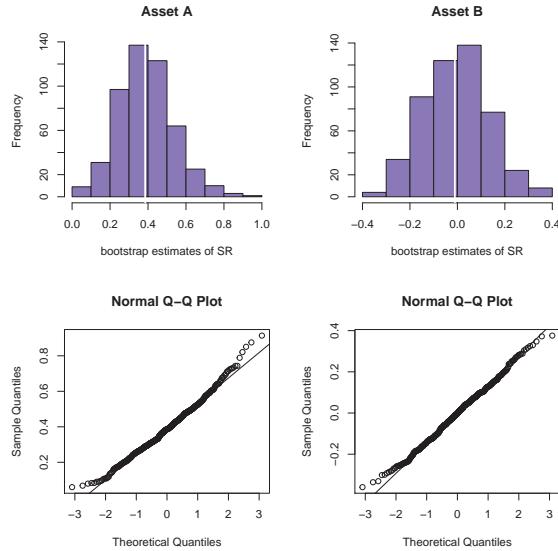


Fig. 15.12 Bootstrap distributions of estimated Sharpe ratios.

15.2 Statistical Analysis of Portfolios: General Case

15.3 Rolling Analysis of Portfolios

15.4 Further Reading

- Mention other quadratic programming packages: ROI etc. See Doug's book
- Mention Pfaff's R book
- Mention Grilli's book etc.

15.5 Problems

References

1. Ruppert, D.A.. *Statistics and Data Analysis for Financial Engineering*, Springer-Verlag.
2. Martin, R.D., Scherer, B., and Yollin, G. (2016).

Chapter 16

Single Index Model

Updated: May 25, 2016

Copyright © Eric Zivot 2015

Outline

1. Single Index Model and Portfolio Theory
 - a. portfolio optimization using SI covariance matrix
 - i. always pd, reduces number of estimated components in matrix.
 - ii. Compare results with sample covariance matrix
 - b. Risk analysis of asset and portfolios: factor risk reports
2. Statistical Properties of Least Squares Estimates
 - a. Bias
 - b. Standard errors
 - c. Asymptotic distributions and confidence intervals
3. Hypothesis Testing in the Single Index Model
 - a. Tests for coefficients
 - b. Tests for model assumptions: check to see if residual correlation matrix is diagonal!
 - c. Diagnostics for covariance stationarity
 - d. Rolling estimates from the single index model

16.1 Motivation

The CER model for monthly asset returns assumes that returns are jointly normally distributed with a constant mean vector and covariance matrix. This model captures many of the stylized facts for monthly asset returns. While the CER model captures the covariances and correlations between different asset returns it does not explain where these covariances and correlations come from. The single index model is an extension of the CER model that

explains the covariance and correlation structure among asset returns as resulting from common exposures to an underlying market index. The intuition behind the single index model can be illustrated by examining some example monthly returns.

Example 16.1. Monthly returns for single index model examples.

For the single index model examples in this chapter, consider the monthly adjusted closing prices on four Northwest stocks (with tickers in parentheses): Boeing (BA), Nordstrom (JWN), Microsoft (MSFT), Starbucks (SBUX). We will also use the monthly adjusted closing prices on the S&P 500 index (\wedge GSPC) as the proxy for the market index. These prices are extracted from the **IntroCompFinR** package as follows:

```
# get data from IntroCompFin package
data(baDailyPrices, jwnDailyPrices, msftDailyPrices, sbuxDailyPrices, sp500DailyPrices)
baPrices = to.monthly(baDailyPrices, OHLC = FALSE)
jwnPrices = to.monthly(jwnDailyPrices, OHLC = FALSE)
msftPrices = to.monthly(msftDailyPrices, OHLC = FALSE)
sbuxPrices = to.monthly(sbuxDailyPrices, OHLC = FALSE)
sp500Prices = to.monthly(sp500DailyPrices, OHLC = FALSE)
siPrices = merge(baPrices, jwnPrices, msftPrices, sbuxPrices, sp500Prices)
```

The data sample is from January, 1998 through May, 2012 and returns are simple returns:

```
smpl = "1998-01::2012-05"
siPrices = siPrices[smpl]
siRetS = na.omit(Return.calculate(siPrices, method = "simple"))
head(siRetS, n = 3)

##           BA      JWN      MSFT      SBUX      SP500
## Feb 1998  0.1425  0.1298  0.13640  0.0825  0.07045
## Mar 1998 -0.0393  0.1121  0.05570  0.1438  0.04995
## Apr 1998 -0.0396  0.0262  0.00691  0.0629  0.00908
```

Prices are shown in Figure 16.1, created using:

```
plot.zoo(siPrices, main = "", lwd = 2, col = "blue")
```

The time plots of prices show some common movements among the stocks that are similar to movements of the S&P 500 index . During the dot-com boom-bust at the beginning of the sample, except for Starbucks, prices rise during the boom and fall during the bust. For all assets, prices rise during the five year boom period prior to the financial crisis, fall sharply after 2008 during the bust, and then rise afterward.

Figure 16.2 shows time plots of the monthly returns on each stock together with the return on the S&P 500 index, created using

```
par(mfrow = c(2, 2))
plot.zoo(siRetS[, c("SP500", "BA")], plot.type = "single", main = "S&P 500 and Boeing",
         ylab = "returns", col = c("blue", "orange"), lwd = c(2, 2))
abline(h = 0)
plot.zoo(siRetS[, c("SP500", "JWN")], plot.type = "single", main = "S&P 500 and Nordstrom",
```

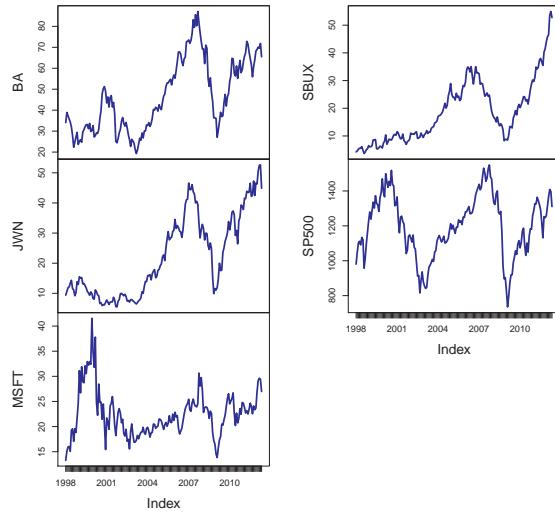


Fig. 16.1 Monthly closing prices on four Northwest stocks and the S&P 500 index.

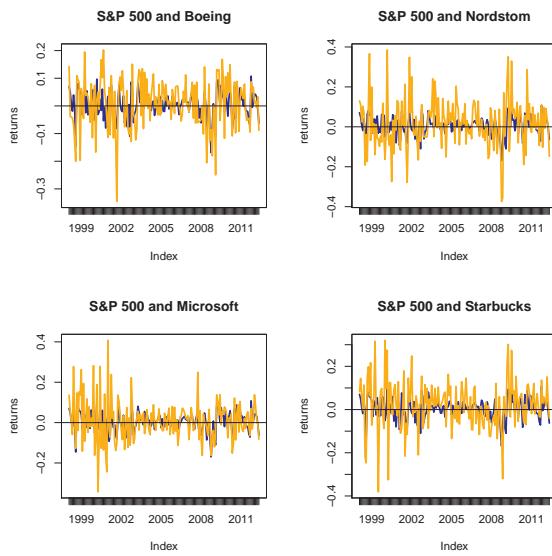


Fig. 16.2 Monthly returns on four Northwest stocks. The orange line in each panel is the monthly return on the stock, and the blue line is the monthly return on the S&P 500 index.

```

    ylab = "returns", col = c("blue", "orange"), lwd = c(2, 2))
abline(h = 0)
plot.zoo(siRetS[, c("SP500", "MSFT")], plot.type = "single", main = "S&P 500 and Microsoft",
          ylab = "returns", col = c("blue", "orange"), lwd = c(2, 2))
abline(h = 0)
plot.zoo(siRetS[, c("SP500", "SBUX")], plot.type = "single", main = "S&P 500 and Starbucks",
          ylab = "returns", col = c("blue", "orange"), lwd = c(2, 2))
abline(h = 0)
par(mfrow = c(1, 1))

```

Figure 16.2 shows that the individual stock returns are more volatile than the S&P 500 returns, and that the movement in stock returns (orange lines) tends to follow the movements in the S&P 500 returns indicating positive covariances and correlations.

The sample return covariance and correlation matrices are computed using:

```

covmatHat = cov(siRetS)
covmatHat

##           BA      JWN     MSFT     SBUX     SP500
## BA     0.00796 0.00334 0.00119 0.00245 0.00223
## JWN    0.00334 0.01492 0.00421 0.00534 0.00339
## MSFT   0.00119 0.00421 0.01030 0.00348 0.00298
## SBUX   0.00245 0.00534 0.00348 0.01191 0.00241
## SP500  0.00223 0.00339 0.00298 0.00241 0.00228

cormatHat = cov2cor(covmatHat)
cormatHat

##           BA      JWN     MSFT     SBUX     SP500
## BA     1.000 0.306 0.131 0.251 0.524
## JWN    0.306 1.000 0.340 0.401 0.581
## MSFT   0.131 0.340 1.000 0.314 0.614
## SBUX   0.251 0.401 0.314 1.000 0.463
## SP500  0.524 0.581 0.614 0.463 1.000

```

The sample correlation matrix is visualized in Figure 16.3 using the **corrplot** function **corrplot.mixed()**:

```
corrplot.mixed(cormatHat, upper = "ellipse")
```

All returns are positively correlated and each stock return has the highest positive correlation with the S&P 500 index.

The positive covariance and correlation of each stock return with the market return can also be visualized with scatterplots as illustrated in Figure 16.4, created with:

```

siRetSmat = coredata(siRetS)
par(mfrow = c(2, 2))
plot(siRetSmat[, "SP500"], siRetSmat[, "BA"], main = "S&P 500 and Boeing", col = "cornflowerblue",
      lwd = 2, pch = 16, cex = 2, xlab = "S&P 500", ylab = "BA", ylim = c(-0.4, 0.4))
abline(h = 0, v = 0)

```

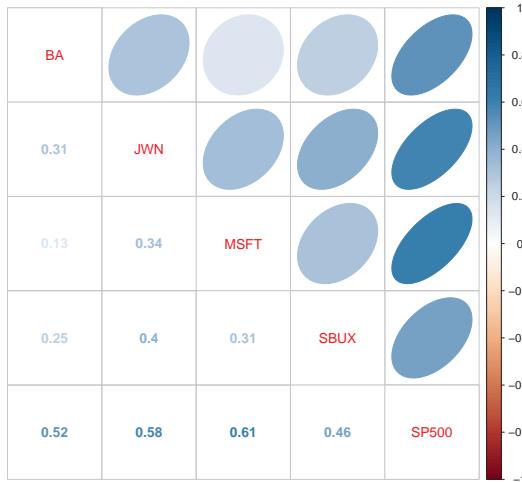


Fig. 16.3 Sample correlation matrix of monthly returns.

```

plot(siRetSmat[, "SP500"], siRetSmat[, "JWN"], main = "S&P 500 and Nordstrom",
      col = "cornflowerblue", lwd = 2, pch = 16, cex = 2, xlab = "S&P 500", ylab = "JWN",
      ylim = c(-0.4, 0.4))
abline(h = 0, v = 0)
plot(siRetSmat[, "SP500"], siRetSmat[, "MSFT"], main = "S&P 500 and Microsoft",
      col = "cornflowerblue", lwd = 2, pch = 16, cex = 2, xlab = "S&P 500", ylab = "MSFT",
      ylim = c(-0.4, 0.4))
abline(h = 0, v = 0)
plot(siRetSmat[, "SP500"], siRetSmat[, "SBUX"], main = "S&P 500 and Starbucks",
      col = "cornflowerblue", lwd = 2, pch = 16, cex = 2, xlab = "S&P 500", ylab = "SBUX",
      ylim = c(-0.4, 0.4))
abline(h = 0, v = 0)

```

The scatterplots show that as the market return increases, the returns on each stock increase in a linear way.

16.2 William Sharpe's Single Index Model

- What is SI model used for? Extension of the CER model to capture the stylized fact of a common component in the returns of many assets.
 - Provides an explanation as to why assets are correlated with each other: share an exposure to a common source.
 - Provides a simplification of the covariance matrix of a large number of asset returns.

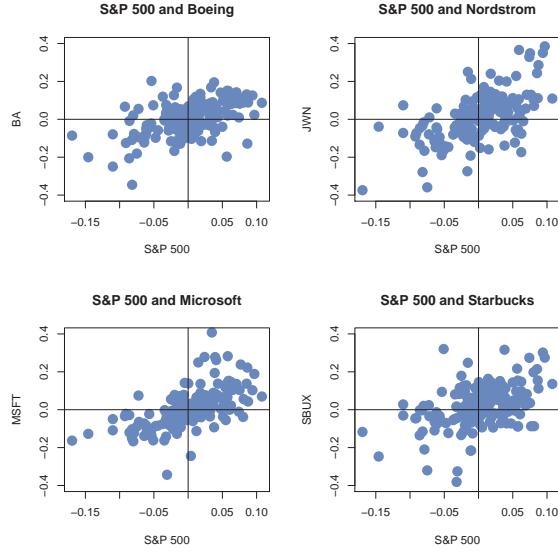


Fig. 16.4 Monthly return scatterplots of each stock vs. the S&P 500 index.

- Provides additional intuition about risk reduction through diversification.
- Example of a factor model for asset returns. Factor models are used heavily in academic theories of asset pricing and in industry for explaining asset returns, portfolio construction and risk analysis. Sharpe's SI model is the most widely used.
- Motivate by showing positive correlation between individual asset returns and market index (sp500)
- Mostly used for simple returns; but can be used for cc returns for risk analysis.

Let R_{it} denote the simple return on asset i over the investment horizon between times $t-1$ and t . Let R_{Mt} denote the simple return on a well diversified market index portfolio, such as the S&P 500 index. The *single index (SI) model* for R_{it} has the form:¹

$$R_{it} = \alpha_i + \beta_i R_{Mt} + \varepsilon_{it}, \quad (16.1)$$

$$R_{Mt} \sim iid N(0, \sigma_M^2), \quad (16.2)$$

$$\varepsilon_{it} \sim GWN(0, \sigma_{\varepsilon,i}^2), \quad (16.3)$$

$$\text{cov}(R_{Mt}, \varepsilon_{is}) = 0, \text{ for all } t \text{ and } s, \quad (16.4)$$

$$\text{cov}(\varepsilon_{it}, \varepsilon_{js}) = 0, \text{ for all } i \neq j \text{ and all } t \text{ and } s. \quad (16.5)$$

The SI model (16.1) - (16.5) assumes that all individual asset returns, R_{it} , are covariance stationary and are a linear function of the market index return, R_{Mt} , and an independent error term, ε_{it} . The SI model is an important extension of the regression form of the CER model. In the SI model individual asset returns are explained by two distinct sources: (1) a common (to all assets) market-wide source R_{Mt} ; (2) and asset specific source ε_{it} .

¹ The single index model is also called the *market model* or the *single factor model*.

16.2.1 Economic interpretation of the SI model

Interpretation of β_i

First, consider the interpretation of β_i in (16.1). The coefficient β_i is called the asset's *market exposure* or *market "beta"*. It represents the slope coefficient in the linear relationship between R_{it} and R_{Mt} . Because it is assumed that R_{Mt} and ϵ_{it} are independent, β_i can be interpreted as the partial derivative

$$\frac{\partial R_{it}}{\partial R_{Mt}} = \frac{\partial}{\partial R_{Mt}}(\alpha_i + \beta_i R_{Mt} + \epsilon_{it}) = \beta_i.$$

Then, for small changes in R_{Mt} denoted ΔR_{Mt} and holding ϵ_{it} fixed, we have the approximation

$$\frac{\Delta R_{it}}{\Delta R_{Mt}} \approx \beta_i \Rightarrow \Delta R_{it} \approx \beta_i \times \Delta R_{Mt},$$

Hence, for a given change in the market return β_i determines the magnitude of the response of asset i 's return. The larger (smaller) is β_i the larger (smaller) is the response of asset i to the movement in the market return.

The coefficient β_i in (16.1) has another interpretation that is directly related to portfolio risk budgeting. In particular, from chapter 14

$$\beta_i = \frac{\text{cov}(R_{it}, R_{Mt})}{\text{var}(R_{Mt})} = \frac{\sigma_{iM}}{\sigma_M^2}, \quad (16.6)$$

and

$$\text{MCR}_i^{\sigma_M} = \beta_i \sigma_M.$$

Here, we see that β_i is the "beta" of asset i with respect to the market portfolio and so it is proportional to the marginal contribution of asset i to the volatility of the market portfolio. Therefore, assets with large (small) values of β_i have large (small) contributions to the volatility of the market portfolio. In this regard, β_i can be thought of as a measure of portfolio risk. Increasing allocations to assets with high (low) β_i values will increase (decrease) portfolio risk (as measured by portfolio volatility). In particular, when $\beta_i = 1$ asset i 's percent contribution to the volatility of the market portfolio is its allocation weight. When $\beta_i > 1$ asset i 's percent contribution to the volatility of the market portfolio is greater than its allocation weight, and when $\beta_i < 1$ asset i 's percent contribution to the volatility of the market portfolio is less than its allocation weight.

The derivation of (16.6) is straightforward. Using (16.1) we can write

$$\begin{aligned} \text{cov}(R_{it}, R_{Mt}) &= \text{cov}(\alpha_i + \beta_i R_{Mt} + \epsilon_{it}, R_{Mt}) \\ &= \text{cov}(\beta_i R_{Mt}, R_{Mt}) + \text{cov}(\epsilon_{it}, R_{Mt}) \\ &= \beta_i \text{var}(R_{Mt}) \quad (\text{since } \text{cov}(\epsilon_{it}, R_{Mt}) = 0) \\ \Rightarrow \beta_i &= \frac{\text{cov}(R_{it}, R_{Mt})}{\text{var}(R_{Mt})}. \end{aligned}$$

Interpretation of R_{Mt} and ϵ_{it}

To aid in the interpretation of R_{Mt} and ϵ_{it} in (16.1), re-write (16.3) as

$$\varepsilon_{it} = R_{it} - \alpha_i - \beta_i R_{Mt}. \quad (16.7)$$

In (16.3), we see that ϵ_{it} is the difference between asset i 's return, R_{it} , and the portion of asset i 's return that is explained by the market return, $\beta_i R_{Mt}$, and the intercept, α_i . We can think of R_{Mt} as capturing “market-wide” news at time t that is common to all assets, and β_i captures the sensitivity or exposure of asset i to this market-wide news. An example of market-wide news is the release of information by the government about the national unemployment rate. If the news is good then we might expect R_{Mt} to increase because general business conditions are good. Different assets will respond differently to this news and this differential impact is captured by β_i . Assets with positive values of β_i will see their returns increase because of good market-wide news, and assets with negative values of β_i will see their returns decrease because of this good news. Hence, the magnitude and direction of correlation between asset returns can be partially explained by their exposures to common market-wide news. Because ϵ_{it} is assumed to be independent of R_{Mt} and of ϵ_{jt} , we can think of ϵ_{it} as capturing specific news to asset i that is unrelated to market news or to specific news to any other asset j . Examples of asset-specific news are company earnings reports and reports about corporate restructuring (e.g., CEO resigning). Hence, specific news for asset i only effects the return on asset i and not the return on any other asset j .

The CER model does not distinguish between overall market and asset specific news and so allows the unexpected news shocks ϵ_{it} to be correlated across assets. Some news is common to all assets but some is specific to a given asset and the CER model error includes both types of news. In this regard, the CER model for an asset's return is not a special case of the SI model when $\beta_i = 0$ because the SI model assumes that ϵ_{it} is uncorrelated across assets.

16.2.2 Statistical properties of returns in the SI model

In this sub-section we present and derive the statistical properties of returns in the SI model (16.1) - (16.5). We will derive two types of statistical properties: unconditional and conditional. The unconditional properties are based on unconditional or marginal distribution of returns. The conditional properties are based on the distribution of returns conditional on the value of the market return.

Unconditional properties

The unconditional properties of returns in the SI model (16.1) - (16.5) are:

$$E[R_{it}] = \mu_i = \alpha_i + \beta_i \mu_M, \quad (16.8)$$

$$\text{var}(R_{it}) = \sigma_i^2 = \beta_i^2 \sigma_M^2 + \sigma_{\epsilon,i}^2, \quad (16.9)$$

$$\text{cov}(R_{it}, R_{jt}) = \sigma_{ij} = \beta_i \beta_j \sigma_M^2, \quad (16.10)$$

$$\text{cor}(R_{it}, R_{jt}) = \rho_{ij} = \frac{\beta_i \beta_j \sigma_M^2}{\sqrt{(\beta_i^2 \sigma_M^2 + \sigma_{\epsilon,i}^2)(\beta_j^2 \sigma_M^2 + \sigma_{\epsilon,j}^2)}}, \quad (16.11)$$

$$R_{it} \sim \text{iid } N(\mu_i, \sigma_i^2) = N(\alpha_i + \beta_i \mu_M, \beta_i^2 \sigma_M^2 + \sigma_{\epsilon,i}^2). \quad (16.12)$$

The derivations of these properties are straightforward and are left as end-of-chapter exercises.

Property (16.8) shows that

$$\alpha_i = \mu_i - \beta_i \mu_M. \quad (16.13)$$

Hence, the intercept term α_i can be interpreted as the average return on asset i that is in excess of the average return due to the market.

From property (16.9), an asset's return variance is additively decomposed into two independent components:

$$\text{var}(R_{it}) = \sigma_i^2 = \beta_i^2 \sigma_M^2 + \sigma_{\epsilon,i}^2 \quad (16.14)$$

(total asset i variance = market variance + asset specific variance)

Here, $\beta_i^2 \sigma_M^2$ is the contribution of the market index return R_{Mt} to the total variance of asset i , and $\sigma_{\epsilon,i}^2$ is the contribution of the asset specific component ϵ_{it} to the total return variance, respectively. If we divide both sides of (16.14) by σ_i^2 we get

$$\begin{aligned} 1 &= \frac{\beta_i^2 \sigma_M^2}{\sigma_i^2} + \frac{\sigma_{\epsilon,i}^2}{\sigma_i^2} \\ &= R + (1 - R^2), \end{aligned}$$

where

$$R^2 = \frac{\beta_i^2 \sigma_M^2}{\sigma_i^2} \quad (16.15)$$

is the proportion of asset i 's variance that is explained by the variability of the market return R_{Mt} and

$$1 - R^2 = \frac{\sigma_{\epsilon,i}^2}{\sigma_i^2} \quad (16.16)$$

is the proportion of asset i 's variance that is explained by the variability of the asset specific component ϵ_{it} .

- Sharpe's rule of thumb. A typical stock has $R^2 = 0.30$. That is, 30% of an asset's variability is explained by the market movements.
- R^2 can be interpreted as the fraction of risk that is non-diversifiable, $1 - R^2$ gives the fraction of risk that is diversifiable. Come back to this point after discussing the SI model and portfolios below.

- This is an example of factor risk budgeting

Properties (16.10) and (16.11) show how assets are correlated in the SI model. In particular,

- $\sigma_{ij} = 0$ if $\beta_i = 0$ or $\beta_j = 0$ or both. Assets i and j are uncorrelated if asset i or asset j or both do not respond to market news.
- $\sigma_{ij} > 0$ if $\beta_i, \beta_j > 0$ or $\beta_i, \beta_j < 0$. Assets i and j are positively correlated if both assets respond to market news in the same direction.
- $\sigma_{ij} < 0$ if $\beta_i > 0$ and $\beta_j < 0$ or if $\beta_i < 0$ and $\beta_j > 0$. Assets i and j are negatively correlated if they respond to market news in opposite directions.

From (16.11), assets i and j are perfectly correlated ($\rho_{ij} = \pm 1$) only if $\sigma_{\epsilon,i} = \sigma_{\epsilon,j} = 0$.

Property (16.12) shows that the distribution of asset returns in the SI model is normal with mean and variance given by () and (), respectively.

In summary, the unconditional properties of returns in the SI model are similar to the properties of returns in the CER model: Returns are covariance stationary with constant means, variances, and covariances. Returns on different assets can be contemporaneously correlated and all asset returns uncorrelated over time. The SI model puts more structure on the expected returns, variances and covariances than the CER model and this allows for a deeper understanding of the behavior of asset returns.

Conditional properties

The properties of returns in the SI model (16.1) - (16.5) conditional on $R_{Mt} = r_{Mt}$ are:

$$E[R_{it}|R_{Mt} = r_{Mt}] = \alpha_i + \beta_i r_{Mt}, \quad (16.17)$$

$$\text{var}(R_{it}|R_{Mt} = r_{Mt}) = \sigma_{\epsilon,i}^2, \quad (16.18)$$

$$\text{cov}(R_{it}, R_{jt}|R_{Mt} = r_{Mt}) = 0, \quad (16.19)$$

$$\text{cor}(R_{it}, R_{jt}|R_{Mt} = r_{Mt}) = 0, \quad (16.20)$$

$$R_{it}|R_{Mt} \sim \text{iid } N(\alpha_i + \beta_i r_{Mt}, \sigma_{\epsilon,i}^2). \quad (16.21)$$

Recall, conditioning on a random variable means we observe its value. In the SI model, once we observe the market return two important things happen: (1) an asset's return variance reduces to its asset specific variance; and (2) asset returns become uncorrelated.

16.2.3 SI model and portfolios

A nice feature of the SI model for asset returns is that it also holds for a portfolio of asset returns. This property follows because asset returns are a linear function of the market return. To illustrate, consider a two asset portfolio with investment weights x_1 and x_2 where each asset return is explained by the SI model:

$$\begin{aligned} R_{1t} &= \alpha_1 + \beta_1 R_{Mt} + \epsilon_{1t}, \\ R_{2t} &= \alpha_2 + \beta_2 R_{Mt} + \epsilon_{2t}. \end{aligned}$$

Then the portfolio return is

$$\begin{aligned} R_{p,t} &= x_1 R_{1t} + x_2 R_{2t} \\ &= x_1(\alpha_1 + \beta_1 R_{Mt} + \epsilon_{1t}) + x_2(\alpha_2 + \beta_2 R_{Mt} + \epsilon_{2t}) \\ &= (x_1\alpha_1 + x_2\alpha_2) + (x_1\beta_1 + x_2\beta_2) R_{Mt} + (x_1\epsilon_{1t} + x_2\epsilon_{2t}) \\ &= \alpha_p + \beta_p R_{Mt} + \epsilon_{p,t}, \end{aligned}$$

where $\alpha_p = x_1\alpha_1 + x_2\alpha_2$, $\beta_p = x_1\beta_1 + x_2\beta_2$, and $\epsilon_{p,t} = x_1\epsilon_{1t} + x_2\epsilon_{2t}$.

SI model and large portfolios

Consider an equally weighted portfolio of N assets, where N is a large number (e.g. $N = 500$) whose returns are described by the SI model. Here, $x_i = 1/N$ for $i = 1, \dots, N$. Then the portfolio return is

$$\begin{aligned} R_{p,t} &= \sum_{i=1}^N x_i R_{it} \\ &= \sum_{i=1}^N x_i (\alpha_i + \beta_i R_{Mt} + \epsilon_{it}) \\ &= \sum_{i=1}^N x_i \alpha_i + \left(\sum_{i=1}^N x_i \beta_i \right) R_{Mt} + \sum_{i=1}^N x_i \epsilon_{it} \\ &= \frac{1}{N} \sum_{i=1}^N \alpha_i + \left(\frac{1}{N} \sum_{i=1}^N \beta_i \right) R_{Mt} + \frac{1}{N} \sum_{i=1}^N \epsilon_{it} \\ &= \bar{\alpha} + \bar{\beta} R_{Mt} + \bar{\epsilon}_t, \end{aligned}$$

where $\bar{\alpha} = \frac{1}{N} \sum_{i=1}^N \alpha_i$, $\bar{\beta} = \frac{1}{N} \sum_{i=1}^N \beta_i$ and $\bar{\epsilon}_t = \frac{1}{N} \sum_{i=1}^N \epsilon_{it}$. Now,

$$\text{var}(\bar{\epsilon}_t) = \text{var} \left(\frac{1}{N} \sum_{i=1}^N \epsilon_{it} \right) = \frac{1}{N^2} \sum_{i=1}^N \text{var}(\epsilon_{it}) = \frac{1}{N} \left(\frac{1}{N} \sum_{i=1}^N \sigma_{\epsilon,i}^2 \right) = \frac{1}{N} \bar{\sigma}^2$$

where $\bar{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N \sigma_{\epsilon,i}^2$ is the average of the asset specific variances. For large N , $\frac{1}{N} \bar{\sigma}^2 \approx 0$ and we have the Law of Large Numbers result

$$\bar{\epsilon}_t = \frac{1}{N} \sum_{i=1}^N \epsilon_{it} \approx E[\epsilon_{it}] = 0.$$

As a result, in a large equally weighted portfolio we have the following:

- $R_{p,t} \approx \bar{\alpha} + \bar{\beta} R_{Mt}$: all non-market asset-specific risk is diversified away and only market risk remains.

- $\text{var}(R_{p,t}) = \bar{\beta}^2 \text{var}(R_{Mt}) \Rightarrow \text{SD}(R_{p,t}) = |\bar{\beta}| \times \text{SD}(R_{Mt})$: portfolio volatility is proportional to market volatility where the factor of proportionality is the absolute value of portfolio beta.
- $R^2 \approx 1$: Approximately 100% of portfolio variance is due to market variance.
- $\bar{\beta} \approx 1$. A large equally weighted portfolio resembles the market portfolio (e.g., as proxied by the S&P 500 index) and so the beta of a well diversified portfolio will be close to the beta of the market portfolio which is one by definition.²

These results help us to understand the type of risk that gets diversified away and the type of risk that remains when we form diversified portfolios. Asset specific risk, which is uncorrelated across assets, gets diversified away whereas market risk, which is common to all assets, does not get diversified away.

- (Relate to average covariance calculation from portfolio theory chapter).
- Related to asset R2 discussed earlier. R2 of an asset shows the portion of risk that cannot be diversified away when forming portfolios.

16.2.4 The SI model in matrix notation

- Need to emphasize that the SI model covariance matrix is always positive definite. This is an important result because it allows for the mean-variance analysis of very large portfolios.

For $i = 1, \dots, N$ assets, stacking (16.1) gives the SI model in matrix notation

$$\begin{pmatrix} R_{1t} \\ \vdots \\ R_{Nt} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix} R_{Mt} + \begin{pmatrix} \epsilon_{1t} \\ \vdots \\ \epsilon_{Nt} \end{pmatrix},$$

or

$$\mathbf{R}_t = \boldsymbol{\alpha} + \boldsymbol{\beta} R_{Mt} + \boldsymbol{\epsilon}_t. \quad (16.22)$$

The unconditional statistical properties of returns (16.8), (16.9), (16.10) and (16.12) can be re-expressed using matrix notation as follows:

$$E[\mathbf{R}_t] = \boldsymbol{\mu} = \boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{\mu}_M, \quad (16.23)$$

$$\text{var}(\mathbf{R}_t) = \boldsymbol{\Sigma} = \sigma_M^2 \boldsymbol{\beta} \boldsymbol{\beta}' + \mathbf{D}, \quad (16.24)$$

$$\mathbf{R}_t \sim iid N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = N(\boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{\mu}_M, \sigma_M^2 \boldsymbol{\beta} \boldsymbol{\beta}' + \mathbf{D}), \quad (16.25)$$

where

² The S&P 500 index is a value weighted index and so is not equal to an equally weighted portfolio of the stocks in the S&P 500 index. However, since most stocks in the S&P 500 index have large market capitalizations the values weights are not too different from equal weights.

$$\mathbf{D} = \text{var}(\epsilon_t) = \begin{pmatrix} \sigma_{\epsilon,1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{\epsilon,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{\epsilon,N}^2 \end{pmatrix} = \text{diag}(\sigma_{\epsilon,1}^2, \sigma_{\epsilon,2}^2, \dots, \sigma_{\epsilon,N}^2). \quad (16.26)$$

The derivation of the SI model covariance matrix (16.24) is

$$\begin{aligned} \text{var}(\mathbf{R}_t) &= \boldsymbol{\Sigma} = \beta \text{var}(R_{Mt})\beta' + \text{var}(\epsilon_t) \\ &= \sigma_M^2 \beta \beta' + \mathbf{D}, \end{aligned}$$

which uses the assumption that the market return R_{Mt} is uncorrelated with all asset specific error terms in ϵ_t .

It is useful to examine the SI covariance matrix (16.24) for a three asset portfolio. In this case, we have

$$\begin{aligned} R_{it} &= \alpha_i + \beta_i R_{Mt} + \epsilon_{it}, \quad i = 1, 2, 3 \\ \sigma_i^2 &= \text{var}(R_{it}) = \beta_i^2 \sigma_M^2 + \sigma_{\epsilon,i}^2 \\ \sigma_{ij} &= \text{cov}(R_{it}, R_{jt}) = \sigma_M^2 \beta_i \beta_j \end{aligned}$$

The 3×3 covariance matrix is

$$\begin{aligned} \boldsymbol{\Sigma} &= \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 \end{pmatrix} \\ &= \begin{pmatrix} \beta_1^2 \sigma_M^2 + \sigma_{\epsilon,1}^2 & \sigma_M^2 \beta_1 \beta_2 & \sigma_M^2 \beta_1 \beta_3 \\ \sigma_M^2 \beta_1 \beta_2 & \beta_2^2 \sigma_M^2 + \sigma_{\epsilon,2}^2 & \sigma_M^2 \beta_2 \beta_3 \\ \sigma_M^2 \beta_1 \beta_3 & \sigma_M^2 \beta_2 \beta_3 & \beta_3^2 \sigma_M^2 + \sigma_{\epsilon,3}^2 \end{pmatrix} \\ &= \sigma_M^2 \begin{pmatrix} \beta_1^2 & \beta_1 \beta_2 & \beta_1 \beta_3 \\ \beta_1 \beta_2 & \beta_2^2 & \beta_2 \beta_3 \\ \beta_1 \beta_3 & \beta_2 \beta_3 & \beta_3^2 \end{pmatrix} + \begin{pmatrix} \sigma_{\epsilon,1}^2 & 0 & 0 \\ 0 & \sigma_{\epsilon,2}^2 & 0 \\ 0 & 0 & \sigma_{\epsilon,3}^2 \end{pmatrix}. \end{aligned}$$

The first matrix shows the return variance and covariance contributions due to the market returns, and the second matrix shows the contributions due to the asset specific errors. Define $\beta = (\beta_1, \beta_2, \beta_3)'$. Then

$$\sigma_M^2 \beta \beta' = \sigma_M^2 \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 \end{pmatrix} = \sigma_M^2 \begin{pmatrix} \beta_1^2 & \beta_1\beta_2 & \beta_1\beta_3 \\ \beta_1\beta_2 & \beta_2^2 & \beta_2\beta_3 \\ \beta_1\beta_3 & \beta_2\beta_3 & \beta_3^2 \end{pmatrix},$$

$$\mathbf{D} = \text{diag}(\sigma_{\varepsilon,1}^2, \sigma_{\varepsilon,2}^2, \sigma_{\varepsilon,3}^2) = \begin{pmatrix} \sigma_{\varepsilon,1}^2 & 0 & 0 \\ 0 & \sigma_{\varepsilon,2}^2 & 0 \\ 0 & 0 & \sigma_{\varepsilon,3}^2 \end{pmatrix},$$

and so

$$\boldsymbol{\Sigma} = \sigma_M^2 \beta \beta' + \mathbf{D}.$$

The matrix form of the SI model (16.23) - (16.25) is useful for portfolio analysis. For example, consider a portfolio with $N \times 1$ weight vector $\mathbf{x} = (x_1, \dots, x_N)'$. Using (16.22), the SI model for the portfolio return $R_{p,t} = \mathbf{x}' R_t$ is

$$\begin{aligned} R_{p,t} &= \mathbf{x}' (\alpha + \beta R_{Mt} + \epsilon_t) \\ &= \mathbf{x}' \alpha + \mathbf{x}' \beta R_{Mt} + \mathbf{x}' \epsilon_t \\ &= \alpha_p + \beta_p R_{Mt} + \epsilon_{p,t}, \end{aligned}$$

where $\alpha_p = \mathbf{x}' \alpha$, $\beta_p = \mathbf{x}' \beta$ and $\epsilon_{p,t} = \mathbf{x}' \epsilon_t$.

16.3 Monte Carlo Simulation of the SI Model

To give a first step reality check for the SI model, consider simulating data from the SI model for a single asset. The steps to create a Monte Carlo simulation from the SI model are:

1. Fix values for the SI model parameters α , β , σ_ϵ , μ_M and σ_M
2. Determine the number of simulated returns, T , to create.
3. Use a computer random number generator to simulate T iid values of $R_{Mt} \sim N(\mu_M, \sigma_M^2)$ and $\epsilon_t \sim N(0, \sigma_\epsilon^2)$, where R_{Mt} is simulated independently from ϵ_t .
4. Create the simulated asset returns $\tilde{R}_t = \alpha + \beta \tilde{R}_M + \tilde{\epsilon}_t$ for $t = 1, \dots, T$

The following example illustrates simulating SI model returns for Boeing.

Example 16.2. Simulating monthly returns from the SI model for Boeing.

Consider simulating monthly returns on Boeing from the SI model where the S&P 500 index is used as the market index. The SI model parameters are calibrated from the actual monthly returns on Boeing and the S&P 500 index using sample statistics, as discussed in the next section, and are given by:

$$\alpha_{BA} = 0.005, \beta_{BA} = 0.98, \sigma_{\epsilon,BA} = 0.08, \mu_M = 0.003, \sigma_M = 0.048.$$

These values are set in R using:

```
alpha.BA = 0.005
beta.BA = 0.98
sd.e.BA = 0.08
mu.sp500 = 0.003
sd.sp500 = 0.048
```

The simulated returns for the S&P 500 index and Boeing are created using:

```
n.sim = nrow(siRetS)
set.seed(123)
sp500.sim = rnorm(n.sim, mu.sp500, sd.sp500)
e.BA.sim = rnorm(n.sim, 0, sd.e.BA)
BA.sim = alpha.BA + beta.BA * sp500.sim + e.BA.sim
BA.sim = xts(BA.sim, index(siRetS))
sp500.sim = xts(sp500.sim, index(siRetS))
colnames(BA.sim) = "BA.sim"
colnames(sp500.sim) = "SP500.sim"
```

The simulate returns are illustrated in Figures 16.5 and 16.6. The time plot and scatterplot of the simulated returns on the S&P 500 index look much like the corresponding plots of the actual returns shows in the top left panels of Figures 16.2 and 16.4, respectively. Hence, the SI model for the S&P 500 index and Boeing passes the first step reality check. In Figure 16.6, the line $\alpha_i + \beta_i R_{Mt} = 0.005 + 0.98R_{Mt}$ is shown in orange. This line represents the predictions of Boeing's return given the S&P 500 return. For example, if $R_{Mt} = 0.05$ then the predicted return for Boeing is $R_{BA} = 0.005 + 0.98 \times (0.05) = 0.054$. The differences between the observed Boeing returns and (blue dots) and the predicted returns (orange line) are the asset specific error terms. These are the random asset specific news components. The standard deviation of these components, $\sigma_{\epsilon,BA} = 0.08$, represents the typical magnitude of these components. Here, we would not be too surprised if the observed Boeing return is 0.08 above or below the orange line. How close the Boeing returns are to the predicted returns is determined by the R^2 value given by (16.15). Here, $\sigma_{BA}^2 = \beta_{BA}^2 \sigma_M^2 + \sigma_{\epsilon,BA}^2 = (0.98)^2 (0.048)^2 + (0.08)^2 = 0.00861$ and so $R^2 = (0.98)^2 (0.048)^2 / 0.00861 = 0.257$. Hence, 25.7% of the variability (risk) of Boeing returns are explained by the variability of the S&P 500 returns. As a result, 74.3% ($1 - R^2 = 0.743$) of the variability of Boeing returns are due to the random asset specific terms.

16.4 Estimation of SI Model

Consider the SI model (16.1) - (16.5). The asset specific parameters to be estimated are α_i , β_i and $\sigma_{\epsilon,i}^2$, ($i = 1, \dots, N$), and the market parameters to be estimated are μ_M and σ_M^2 . These parameters can be estimated using the plug-in principle, linear regression, and maximum likelihood. All methods give essentially the same estimators for the SI model parameters.

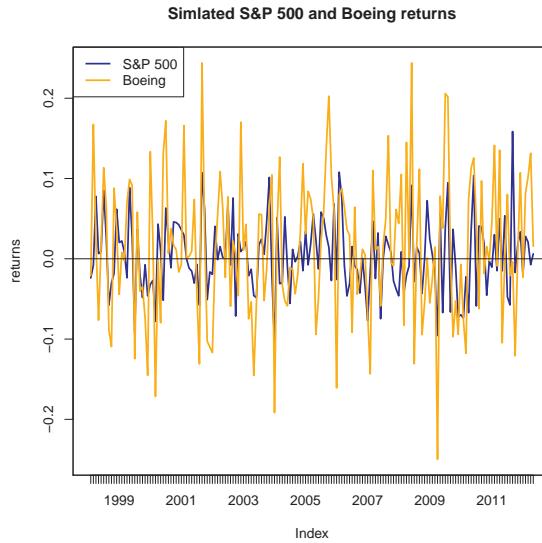


Fig. 16.5 Simulated SI model monthly returns on the S&P 500 index and Boeing.

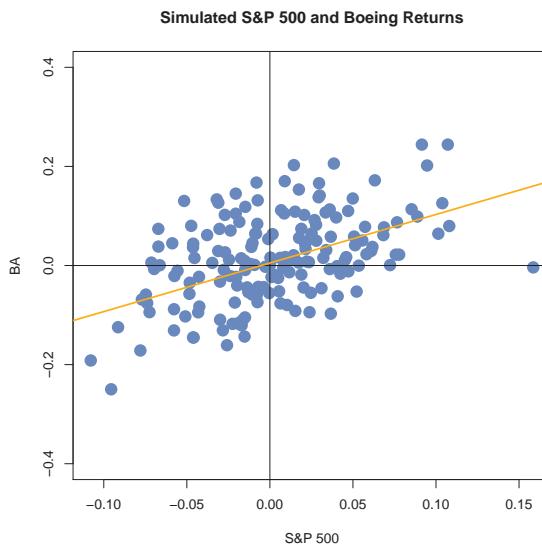


Fig. 16.6 Simulated SI model monthly returns on the S&P 500 index and Boeing. The orange line is the equation $R_{BA,t} = \alpha_{BA} + \beta_{BA}R_{Mt} = 0.005 + 0.98 \times R_{Mt}$.

16.4.1 Plug-in principle estimates

Let $\{(r_{it}, r_{Mt})\}_{t=1}^T$ denote a sample of size T of observed returns on asset i and the market return which are assumed to be generated from the SI model. (16.1) - (16.12). Recall, the plug-in principle says to estimate population model parameters using appropriate sample statistics. For the market parameters, the plug-in principle estimates are the same as the CER model estimates

$$\hat{\mu}_M = \frac{1}{T} \sum_{t=1}^T r_{Mt},$$

$$\hat{\sigma}_M^2 = \frac{1}{T-1} \sum_{t=1}^T (r_{Mt} - \hat{\mu}_M)^2.$$

From (16.13) - (16.6) we see that α_i and β_i are functions of population parameters

$$\alpha_i = \mu_i - \beta_i \mu_M,$$

$$\beta_i = \frac{\text{cov}(R_{it}, R_{Mt})}{\text{var}(R_{Mt})} = \frac{\sigma_{iM}}{\sigma_M^2}.$$

The corresponding plug-in principle estimates are then:

$$\hat{\alpha}_i = \hat{\mu}_i - \hat{\beta}_i \hat{\mu}_M, \quad (16.27)$$

$$\hat{\beta}_i = \frac{\hat{\sigma}_{iM}}{\hat{\sigma}_M^2}, \quad (16.28)$$

where

$$\hat{\mu}_i = \frac{1}{T} \sum_{t=1}^T r_{it},$$

$$\hat{\sigma}_{iM} = \frac{1}{T-1} \sum_{t=1}^T (r_{it} - \hat{\mu}_i)(r_{Mt} - \hat{\mu}_M).$$

Given the plug-in principle estimates $\hat{\alpha}_i$ and $\hat{\beta}_i$, the plug-in principle estimate of ϵ_{it} is

$$\hat{\epsilon}_{it} = r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{Mt}, \quad t = 1, \dots, T. \quad (16.29)$$

Using (16.29), the plug-in principle estimate of $\sigma_{\epsilon,i}^2$ is the sample variance of $\{\hat{\epsilon}_{it}\}_{t=1}^T$ (adjusted for the number of degrees of freedom):

$$\hat{\sigma}_{\epsilon,i}^2 = \frac{1}{T-2} \sum_{t=1}^T \hat{\epsilon}_t^2 = \frac{1}{T-2} \sum_{t=1}^T (r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{Mt})^2. \quad (16.30)$$

Plug-in principle estimates of R^2 based on (16.15) can be computed using

$$\hat{R}^2 = \frac{\hat{\beta}_i^2 \hat{\sigma}_M^2}{\hat{\sigma}_i^2} = 1 - \frac{\hat{\sigma}_{\epsilon,i}^2}{\hat{\sigma}_i^2}. \quad (16.31)$$

Example 16.3. Computing plug-in principle estimators for SI model parameters.

Consider computing the plug-in principle estimates for α_i , β_i and $\sigma_{\epsilon,i}^2$ from the example data using the formulas (16.27), (16.28) and (16.30), respectively. First, extract the sample statistics $\hat{\mu}_i$, $\hat{\sigma}_{iM}$, $\hat{\mu}_M$, and $\hat{\sigma}_M^2$:

```
assetNames = colnames(siRetS)[1:4]
muhat = colMeans(siRetS)
sig2hat = diag(covmatHat)
covAssetsSp500 = covmatHat[assetNames, "SP500"]
```

Next, estimate $\hat{\beta}_i$ using

```
betaHat = covAssetsSp500/sig2hat["SP500"]
betaHat

##      BA      JWN      MSFT      SBUX
## 0.978 1.485 1.303 1.057
```

Here, we see that $\hat{\beta}_{BA}$ and $\hat{\beta}_{SBUX}$ are very close to one and that $\hat{\beta}_{JWN}$ and $\hat{\beta}_{MSFT}$ are slightly bigger than one. Using the estimates of $\hat{\beta}_i$ and the sample statistics $\hat{\mu}_i$ and $\hat{\mu}_M$ the estimates for $\hat{\alpha}_i$ are

```
alphaHat = muhat[assetNames] - betaHat * muhat["SP500"]
alphaHat

##      BA      JWN      MSFT      SBUX
## 0.00516 0.01231 0.00544 0.01785
```

All of the estimates of $\hat{\alpha}_i$ are close to zero. The estimates of $\sigma_{\epsilon,i}^2$ can be computed using:

```
sig2eHat = rep(0, length(assetNames))
names(sig2eHat) = assetNames
for (aName in assetNames) {
  eHat = siRetS[, aName] - alphaHat[aName] - betaHat[aName] * siRetS[, "SP500"]
  sig2eHat[aName] = crossprod(eHat)/(length(eHat) - 2)
}
sig2eHat

##      BA      JWN      MSFT      SBUX
## 0.00581 0.00994 0.00646 0.00941
```

Lastly, the estimates of R^2 can be computed using

```
R2 = 1 - sig2eHat/sig2hat[assetNames]
R2

##      BA      JWN      MSFT      SBUX
## 0.270 0.334 0.373 0.210
```

16.4.2 Least squares estimates

The SI model representation (16.1) shows that returns are a linear function of the market return and an asset specific error term

$$R_{it} = \alpha_i + \beta_i R_{Mt} + \epsilon_{it},$$

here α_i is the intercept and β_i is the slope. *Least squares regression* is a method for estimating α_i and β_i by finding the “best fitting” line to the scatterplot of returns where R_{it} is on the vertical axis and R_{Mt} is on the horizontal axis.

[Insert Figure here]

To see how the method of least squares determines the “best fitting” line, consider the scatterplot of the sample returns on Boeing and the S&P 500 index illustrated in Figure xxx. In the figure, the black line is a fitted line with initial guess $\hat{\alpha}_{BA} = 0$ and $\hat{\beta}_{BA} = 0.5$. The differences between the observed returns (blue dots) and the values on the fitted line are the estimated errors $\hat{\epsilon}_{BA,t} = r_{BA,t} - \hat{\alpha}_{BA} - \hat{\beta}_{BA} R_{Mt} = r_{BA,t} - 0 - 0.5 \times R_{Mt}$. Some estimated errors are big and some are small. The overall fit of the line can be measured using a statistic based on all $t = 1, \dots, T$ of the estimated errors. A natural choice is the sum of the errors $\sum_{t=1}^T \hat{\epsilon}_t$. However, this choice can be misleading due to the canceling out of large positive and negative errors. To avoid this problem, it is better to measure the overall fit using $\sum_{t=1}^T |\hat{\epsilon}_t|$ or $\sum_{t=1}^T \hat{\epsilon}_t^2$. Then the best fitting line can be determined by finding the intercept and slope values that minimize $\sum_{t=1}^T |\hat{\epsilon}_t|$ or $\sum_{t=1}^T \hat{\epsilon}_t^2$.

The method of least squares regression defines the “best fitting” line by finding the intercept and slope values that minimize the sum of squared errors

$$\text{SSE}(\hat{\alpha}_i, \hat{\beta}_i) = \sum_{t=1}^T \hat{\epsilon}_{it}^2 = \sum_{t=1}^T (r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{Mt})^2. \quad (16.32)$$

Because $\text{SSE}(\hat{\alpha}, \hat{\beta})$ is a continuous and differential function of $\hat{\alpha}_i$ and $\hat{\beta}_i$, the minimizing values of $\hat{\alpha}_i$ and $\hat{\beta}_i$ can be determined using simple calculus. The first order conditions for a minimum are:

$$0 = \frac{\partial \text{SSE}(\hat{\alpha}_i, \hat{\beta}_i)}{\partial \hat{\alpha}_i} = -2 \sum_{t=1}^T (r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{Mt}) = -2 \sum_{t=1}^T \hat{\epsilon}_{it}, \quad (16.33)$$

$$0 = \frac{\partial \text{SSE}(\hat{\alpha}_i, \hat{\beta}_i)}{\partial \hat{\beta}_i} = -2 \sum_{t=1}^T (r_{it} - \hat{\alpha}_i - \hat{\beta}_i r_{Mt}) r_{Mt} = -2 \sum_{t=1}^T \hat{\epsilon}_{it} r_{Mt}. \quad (16.34)$$

These are two linear equations in two unknowns which can be re-expressed as

$$\begin{aligned} \hat{\alpha}_i T + \hat{\beta}_i \sum_{t=1}^T r_{Mt} &= \sum_{t=1}^T r_{it}, \\ \hat{\alpha}_i \sum_{t=1}^T r_{Mt} + \hat{\beta}_i \sum_{t=1}^T r_{Mt}^2 &= \sum_{t=1}^T r_{it} r_{Mt}. \end{aligned}$$

Using matrix algebra, we can write these equations as:

$$\begin{pmatrix} T & \sum_{t=1}^T r_{Mt} \\ \sum_{t=1}^T r_{Mt} & \sum_{t=1}^T r_{Mt}^2 \end{pmatrix} \begin{pmatrix} \hat{\alpha}_i \\ \hat{\beta}_i \end{pmatrix} = \begin{pmatrix} \sum_{t=1}^T r_{it} \\ \sum_{t=1}^T r_{it} r_{Mt} \end{pmatrix}, \quad (16.35)$$

which is of the form $\mathbf{Ax} = \mathbf{b}$ with

$$\mathbf{A} = \begin{pmatrix} T & \sum_{t=1}^T r_{Mt} \\ \sum_{t=1}^T r_{Mt} & \sum_{t=1}^T r_{Mt}^2 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \hat{\alpha}_i \\ \hat{\beta}_i \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \sum_{t=1}^T r_{it} \\ \sum_{t=1}^T r_{it} r_{Mt} \end{pmatrix}.$$

Hence, we can determine $\hat{\alpha}_i$ and $\hat{\beta}_i$ by solving $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Now,³

$$\begin{aligned} \mathbf{A}^{-1} &= \frac{1}{\det(\mathbf{A})} \begin{pmatrix} \sum_{t=1}^T r_{Mt}^2 & -\sum_{t=1}^T r_{Mt} \\ -\sum_{t=1}^T r_{Mt} & T \end{pmatrix}, \\ \det(\mathbf{A}) &= T \sum_{t=1}^T r_{Mt}^2 - \left(\sum_{t=1}^T r_{Mt} \right)^2 = T \sum_{t=1}^T (r_{Mt} - \hat{\mu}_M)^2, \\ \hat{\mu}_M &= \frac{1}{T} \sum_{t=1}^T r_{Mt}. \end{aligned}$$

Consequently,

$$\begin{pmatrix} \hat{\alpha}_i \\ \hat{\beta}_i \end{pmatrix} = \frac{1}{T \sum_{t=1}^T (r_{Mt} - \hat{\mu}_M)^2} \begin{pmatrix} \sum_{t=1}^T r_{Mt}^2 & -\sum_{t=1}^T r_{Mt} \\ -\sum_{t=1}^T r_{Mt} & T \end{pmatrix} \begin{pmatrix} \sum_{t=1}^T r_{it} \\ \sum_{t=1}^T r_{it} r_{Mt} \end{pmatrix} \quad (16.36)$$

and so

$$\hat{\alpha}_i = \frac{\sum_{t=1}^T r_{Mt}^2 \sum_{t=1}^T r_{it} - \sum_{t=1}^T r_{Mt}^2 \sum_{t=1}^T r_{it} r_{Mt}}{T \sum_{t=1}^T (r_{Mt} - \hat{\mu}_M)^2}, \quad (16.37)$$

$$\hat{\beta}_i = \frac{T \sum_{t=1}^T r_{it} r_{Mt} - \sum_{t=1}^T r_{Mt} \sum_{t=1}^T r_{it}}{T \sum_{t=1}^T (r_{Mt} - \hat{\mu}_M)^2}. \quad (16.38)$$

After a little bit of algebra (see end-of-chapter exercises) it can be shown that

$$\begin{aligned} \hat{\alpha}_i &= \hat{\mu}_i - \hat{\beta}_i \hat{\mu}_M, \\ \hat{\beta}_i &= \frac{\hat{\sigma}_{iM}}{\hat{\sigma}_M^2}, \end{aligned}$$

which are plug-in estimates for $\hat{\alpha}_i$ and $\hat{\beta}_i$ determined earlier. Hence, the least squares estimates of $\hat{\alpha}_i$ and $\hat{\beta}_i$ are identical to the plug-in estimates.

³ The matrix \mathbf{A} is invertible provided $\det(A) \neq 0$. This requires the sample variance of R_{Mt} to be non-zero.

The solution for the least squares estimates in (16.36) has an elegant representation using matrix algebra. To see this, define the $T \times 1$ vectors $\mathbf{r}_i = (r_{i1}, \dots, r_{iT})'$, $\mathbf{r}_M = (r_{M1}, \dots, r_{MT})'$ and $\mathbf{1} = (1, \dots, 1)'$. Then we can re-write (16.35) as

$$\begin{pmatrix} \mathbf{1}'\mathbf{1} & \mathbf{1}'\mathbf{r}_M \\ \mathbf{1}'\mathbf{r}_M & \mathbf{r}_M'\mathbf{r}_M \end{pmatrix} \begin{pmatrix} \hat{\alpha}_i \\ \hat{\beta}_i \end{pmatrix} = \begin{pmatrix} \mathbf{1}'\mathbf{r}_i \\ \mathbf{r}_M'\mathbf{r}_i \end{pmatrix}$$

or

$$\mathbf{X}'\mathbf{X}\hat{\gamma}_i = \mathbf{X}'\mathbf{r}_i \quad (16.39)$$

where $\mathbf{X} = (\mathbf{1} \ \mathbf{r}_M)$ is a $T \times 2$ matrix and $\hat{\gamma} = (\hat{\alpha}_i, \hat{\beta}_i)'$. Provided $\mathbf{X}'\mathbf{X}$ is invertible, solving (16.39) for $\hat{\gamma}_i$ gives the least squares estimates in matrix form:

$$\hat{\gamma}_i = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{r}_i. \quad (16.40)$$

The matrix form solution (16.40) is especially convenient for computation in R.

The least squares estimates of ϵ_t , $\sigma_{\epsilon,i}^2$ and R^2 are the same as the plug-in estimators (16.29), (16.30) and (16.31), respectively. In the context of least squares estimation, the estimate $\hat{\sigma}_{\epsilon,i} = \sqrt{\hat{\sigma}_{\epsilon,i}^2}$ is called the *standard error of the regression* and measures the typical magnitude of $\hat{\epsilon}_t$ (difference between observed return and fitted regression line).

16.4.3 Simple linear regression in R

- don't do regression examples until statistical theory is discussed
- computing least squares estimates using matrix algebra formulas
- computing least squares estimates using lm()
 - See discussion from my regression chapter in MFTSR
 - describe structure of lm() function, extractor and method functions
- Do analysis of example data

16.4.4 Maximum likelihood estimates

The SI model parameters can also be estimated using the method of maximum likelihood, which was introduced in chapter (GARCH estimation chapter). To construct the likelihood function, we use property () of the SI model that conditional on $R_{Mt} = r_{Mt}$ the distribution of R_{it} is normal with mean $\alpha_i + \beta_i r_{Mt}$ and variance $\sigma_{\epsilon,i}^2$. The pdf of $R_{it}|R_{Mt} = r_{mt}$ is then

$$f(r_{it}|r_{mt}, \theta_i) = (2\pi\sigma_{\epsilon,i}^2)^{-1/2} \exp\left(\frac{-1}{2\sigma_{\epsilon,i}^2} (r_{it} - \alpha_i + \beta_i r_{Mt})^2\right), \quad t = 1, \dots, T,$$

where $\theta_i = (\alpha_i, \beta_i, \sigma_{\epsilon,i}^2)'$. Given a sample $\{(r_{it}, r_{Mt})\}_{t=1}^T = \{\mathbf{r}_i, \mathbf{r}_M\}$ of observed returns on asset i and the market return, which are assumed to be generated from the SI model, the joint density of asset returns given the market returns is

$$\begin{aligned} f(\mathbf{r}_i | \mathbf{r}_m) &= \prod_{t=1}^T (2\pi\sigma_{\epsilon,i}^2)^{-1/2} \exp\left(\frac{-1}{2\sigma_{\epsilon,i}^2} (r_{it} - \alpha_i + \beta_i r_{Mt})^2\right) \\ &= (2\pi\sigma_{\epsilon,i}^2)^{-T/2} \exp\left(\frac{-1}{2\sigma_{\epsilon,i}^2} \sum_{t=1}^T (r_{it} - \alpha_i + \beta_i r_{Mt})^2\right) \\ &= (2\pi\sigma_{\epsilon,i}^2)^{-T/2} \exp\left(\frac{-1}{2\sigma_{\epsilon,i}^2} \text{SSE}(\alpha_i, \beta_i)\right). \end{aligned}$$

where $\text{SSE}(\alpha_i, \beta_i)$ is the sum of squared residuals (16.32) used to determine the least squares estimates. The log-likelihood function for θ_i is then

$$\ln L(\theta_i | \mathbf{r}_i, \mathbf{r}_M) = \frac{-T}{2} \ln(2\pi) - \frac{T}{2} \ln(\sigma_{\epsilon,i}^2) - \frac{1}{2\sigma_{\epsilon,i}^2} \text{SSE}(\alpha_i, \beta_i). \quad (16.41)$$

From (16.41), it can be seen that the values of α_i and β_i that maximize the log-likelihood are the values that minimize $\text{SSE}(\alpha_i, \beta_i)$. Hence, the ML estimates of α_i and β_i are the least squares estimates.

To find the ML estimate for $\sigma_{\epsilon,i}^2$, plug the ML estimates of α_i and β_i into (16.41) giving

$$\ln L(\hat{\alpha}_i, \hat{\beta}_i, \sigma_{\epsilon,i}^2 | \mathbf{r}_i, \mathbf{r}_M) = \frac{-T}{2} \ln(2\pi) - \frac{T}{2} \ln(\sigma_{\epsilon,i}^2) - \frac{1}{2\sigma_{\epsilon,i}^2} \text{SSE}(\hat{\alpha}_i, \hat{\beta}_i).$$

Maximization with respect to $\sigma_{\epsilon,i}^2$ gives the first order condition

$$\frac{\partial \ln L(\hat{\alpha}_i, \hat{\beta}_i, \sigma_{\epsilon,i}^2 | \mathbf{r}_i, \mathbf{r}_M)}{\partial \sigma_{\epsilon,i}^2} = -\frac{T}{2\hat{\sigma}_{\epsilon,i}^2} + \frac{1}{2(\hat{\sigma}_{\epsilon,i}^2)^2} \text{SSE}(\hat{\alpha}_i, \hat{\beta}_i) = 0.$$

Solving for $\hat{\sigma}_{\epsilon,i}^2$ gives the ML estimate for $\sigma_{\epsilon,i}^2$:

$$\hat{\sigma}_{\epsilon,i}^2 = \frac{\text{SSE}(\hat{\alpha}_i, \hat{\beta}_i)}{T} = \frac{1}{T} \sum_{t=1}^T \hat{\epsilon}_t^2,$$

which is plug-in principle estimate (16.30) not adjusted for degrees-of-freedom.

16.5 Statistical Properties of SI Model Estimates

To determine the statistical properties of the plug-in principle/least squares/ML estimators $\hat{\alpha}_i$, $\hat{\beta}_i$ and $\hat{\sigma}_{\epsilon,i}^2$ in the SI model, we treat them as functions of the random variables $\{(R_{i,t}, R_{Mt})\}_{t=1}^T$ where R_t and R_{Mt} are assumed to be generated by the SI model (16.1) - (16.5).

16.5.1 Bias

In the SI model, the estimators $\hat{\alpha}_i$, $\hat{\beta}_i$ and $\hat{\sigma}_{\epsilon,i}^2$ (with degrees-of-freedom adjustment) are unbiased:

$$\begin{aligned} E[\hat{\alpha}_i] &= \alpha_i, \\ E[\hat{\beta}_i] &= \beta_i, \\ E[\hat{\sigma}_{\epsilon,i}^2] &= \sigma_{\epsilon,i}^2. \end{aligned}$$

To show that $\hat{\alpha}_i$ and $\hat{\beta}_i$ are unbiased, it is useful to consider the SI model for asset i in matrix form for $t = 1, \dots, T$:

$$\mathbf{R}_i = \alpha_i \mathbf{1} + \beta_i \mathbf{R}_M + \epsilon_i = \mathbf{X} \gamma_i + \epsilon_i,$$

where $\mathbf{R}_i = (R_{i1}, \dots, R_{iT})'$, $\mathbf{R}_M = (R_{M1}, \dots, R_{MT})'$, $\epsilon_i = (\epsilon_{i1}, \dots, \epsilon_{iT})'$, $\mathbf{1} = (1, \dots, 1)'$, $\mathbf{X} = (\mathbf{1} \ \mathbf{R}_M)$, and $\gamma_i = (\alpha_i, \beta_i)'$. The estimator for γ_i is

$$\hat{\gamma}_i = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{R}_i.$$

Plugging in $\mathbf{R}_i = \mathbf{X} \gamma_i + \epsilon_i$ gives

$$\begin{aligned} \hat{\gamma}_i &= (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' (\mathbf{X} \gamma_i + \epsilon_i) = (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{X} \gamma_i + (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \epsilon_i \\ &= \gamma_i + (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \epsilon_i. \end{aligned}$$

Then

$$\begin{aligned} E[\hat{\gamma}_i] &= \gamma_i + E[(\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \epsilon_i] \\ &= \gamma_i + E[(\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}'] E[\epsilon_i] \text{ (because } \epsilon_{it} \text{ is independent of } R_{Mt} \text{)} \\ &= \gamma_i \text{ (because } E[\epsilon_i] = \mathbf{0}). \end{aligned}$$

The derivation of $E[\hat{\sigma}_{\epsilon,i}^2] = \sigma_{\epsilon,i}^2$ is beyond the scope of this book and can be found in graduate econometrics textbooks such as Hayashi (1980).

16.5.2 Precision

Under the assumptions of the SI model, analytic formulas for estimates of the standard errors for $\hat{\alpha}_i$ and $\hat{\beta}_i$ are given by:

$$\widehat{\text{se}}(\hat{\alpha}_i) \approx \frac{\hat{\sigma}_{\epsilon,i}}{\sqrt{T \cdot \hat{\sigma}_M^2}} \cdot \sqrt{\frac{1}{T} \sum_{t=1}^T r_{Mt}^2}, \quad (16.42)$$

$$\widehat{\text{se}}(\hat{\beta}_i) \approx \frac{\hat{\sigma}_{\epsilon,i}}{\sqrt{T \cdot \hat{\sigma}_M^2}}, \quad (16.43)$$

where “ \approx ” denotes an approximation based on the CLT that gets more accurate the larger the sample size. Remarks:

- $\widehat{\text{se}}(\hat{\alpha}_i)$ and $\widehat{\text{se}}(\hat{\beta}_i)$ are smaller the smaller is $\hat{\sigma}_{\varepsilon,i}$. That is, the closer are returns to the fitted regression line the smaller are the estimation errors in $\hat{\alpha}_i$ and $\hat{\beta}_i$.
- $\widehat{\text{se}}(\hat{\beta}_i)$ is smaller the larger is $\hat{\sigma}_M^2$. That is, the greater the variability in the market return R_{Mt} the smaller is the estimation error in the estimated slope coefficient $\hat{\beta}_i$. This is illustrated in Figure xxx. The left panel shows a data sample from the SI model with a small value of $\hat{\sigma}_M^2$ and the right panel shows a sample with a large value of $\hat{\sigma}_M^2$. The right panel shows that the high variability in R_{Mt} makes it easier to identify the slope of the line.
- Both $\widehat{\text{se}}(\hat{\alpha}_i)$ and $\widehat{\text{se}}(\hat{\beta}_i)$ go to zero as the sample size, T , gets large. Since $\hat{\alpha}_i$ and $\hat{\beta}_i$ are unbiased estimators, this implies that they are also consistent estimators. That is, they converge to the true values α_i and β_i , respectively, as $T \rightarrow \infty$.
- In R, the standard error values (16.42) and (16.43) are computed using the `summary()` function on an “`lm`” object.

There are no easy formulas for the estimated standard errors for $\hat{\sigma}_{\varepsilon,i}^2$, $\hat{\sigma}_{\varepsilon,i}$ and \hat{R}^2 . Estimated standard errors for these estimators, however, can be easily computed using the bootstrap.

Example 16.4. Computing estimated standard errors for $\hat{\alpha}_i$ and $\hat{\beta}_i$ in R.

to be completed



16.5.3 Sampling distribution and confidence intervals.

Using arguments based on the CLT, it can be shown that for large enough T the estimators $\hat{\alpha}_i$ and $\hat{\beta}_i$ are approximately normally distributed:

$$\begin{aligned}\hat{\alpha}_i &\sim N(\alpha_i, \widehat{\text{se}}(\hat{\alpha}_i)^2), \\ \hat{\beta}_i &\sim N(\beta_i, \widehat{\text{se}}(\hat{\beta}_i)^2),\end{aligned}$$

where $\widehat{\text{se}}(\hat{\alpha}_i)$ and $\widehat{\text{se}}(\hat{\beta}_i)$ are given by (16.42) and (16.43), respectively.

16.6 Problems

References

1. Ruppert, D.A.. *Statistics and Data Analysis for Financial Engineering*, Springer-Verlag.
2. Martin, R.D., Scherer, B., and Yollin, G. (2016).

Glossary

Here comes the glossary

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

glossary term Write here the description of the glossary term. Write here the description of the glossary term. Write here the description of the glossary term.

Solutions

Problems of Chapter ??

There two styles for solutions: **Sol** and **Solution**. The **Sol** style is the preferred because it references the corresponding **Prob** problem style.

An examples of **Sol**:

?? The solution of the corresponding problem. The number of the prob/sol can also be referenced in the text: “As explained in the description of problem ??...”

?? Problem Heading

- (a) The solution of first part is revealed here.
- (b) The solution of second part is revealed here.

An example of **Solution**:

Solution 16.1 (Title/Name (optional)). The solution is revealed here.

Index

- acronyms, list of, xxi
- cross lag correlation, 141
- dedication, v
- glossary, 503
- solutions, 505
- symbols, list of, xxi