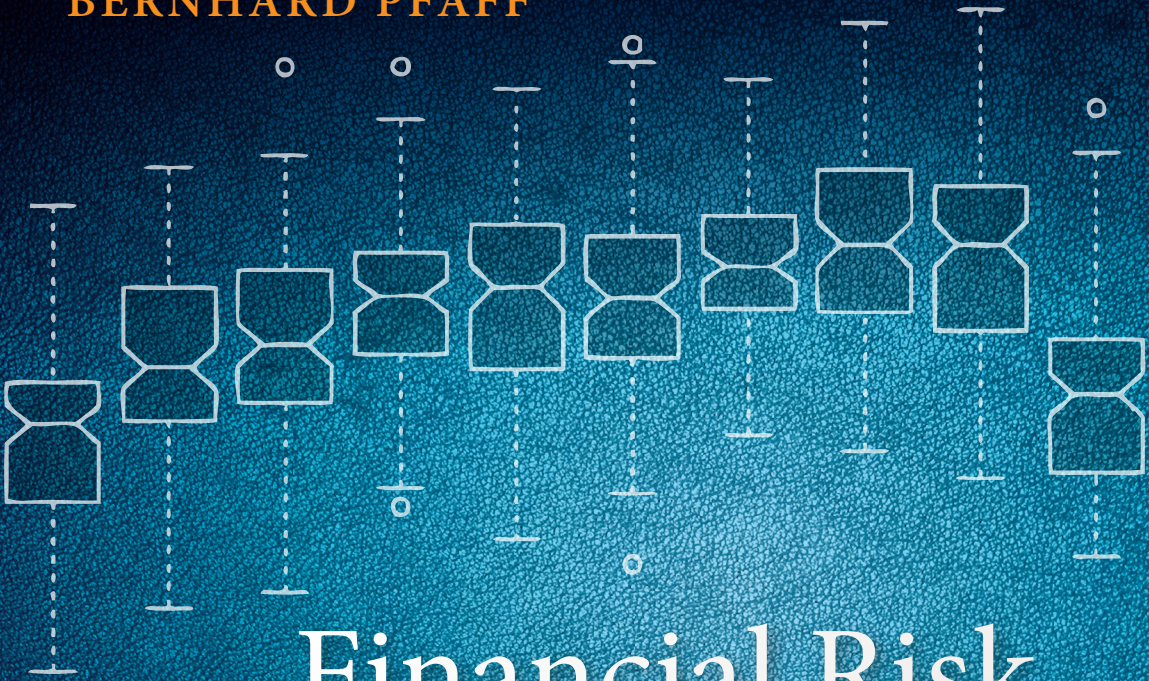


BERNHARD PFAFF



# Financial Risk Modelling and Portfolio Optimization with R

 WILEY

STATISTICS IN PRACTICE



# Financial Risk Modelling and Portfolio Optimization with R

# Statistics in Practice

Series Advisory Editors

**Marian Scott**

*University of Glasgow, UK*

**Stephen Senn**

*CRP-Santé, Luxembourg*

**Wolfgang Jank**

*University of Maryland, USA*

Founding Editor

**Vic Barnett**

*Nottingham Trent University, UK*

---

*Statistics in Practice* is an important international series of texts which provide detailed coverage of statistical concepts, methods and worked case studies in specific fields of investigation and study.

With sound motivation and many worked practical examples, the books show in down-to-earth terms how to select and use an appropriate range of statistical techniques in a particular practical field within each title's special topic area.

The books provide statistical support for professionals and research workers across a range of employment fields and research environments. Subject areas covered include medicine and pharmaceuticals; industry, finance and commerce; public services; the earth and environmental sciences, and so on.

The books also provide support to students studying statistical courses applied to the above areas. The demand for graduates to be equipped for the work environment has led to such courses becoming increasingly prevalent at universities and colleges.

It is our aim to present judiciously chosen and well-written workbooks to meet everyday practical needs. Feedback of views from readers will be most valuable to monitor the success of this aim.

A complete list of titles in this series appears at the end of the volume.

# Financial Risk Modelling and Portfolio Optimization with R

**Bernhard Pfaff**

*Invesco Global Strategies, Germany*



A John Wiley & Sons, Ltd., Publication

This edition first published 2013  
© 2013 John Wiley & Sons, Ltd

*Registered Office*

John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ,  
United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at [www.wiley.com](http://www.wiley.com).

The right of the author to be identified as the author of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book. This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

*Library of Congress Cataloging-in-Publication Data*

Pfaff, Bernhard.

Financial risk modelling and portfolio optimization with R / Bernhard Pfaff.  
p. cm.

Includes bibliographical references and index.

ISBN 978-0-470-97870-2 (cloth)

1. Financial risk—Mathematical models. 2. Portfolio management.

3. R (Computer program language) I. Title.

HG106.P484 2013

332.0285'5133—dc23

2012030904

A catalogue record for this book is available from the British Library.

ISBN: 978-0-470-97870-2

Set in 10/12 pt Times Roman by Aptara Inc., New Delhi, India

# Contents

<b>Preface</b>	<b>xi</b>
<b>List of abbreviations</b>	<b>xiii</b>
<b>Part I MOTIVATION</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
Reference	5
<b>2 A brief course in R</b>	<b>6</b>
2.1 Origin and development	6
2.2 Getting help	7
2.3 Working with R	10
2.4 Classes, methods and functions	12
2.5 The accompanying package FRAPO	20
References	25
<b>3 Financial market data</b>	<b>26</b>
3.1 Stylized facts on financial market returns	26
3.1.1 Stylized facts for univariate series	26
3.1.2 Stylized facts for multivariate series	29
3.2 Implications for risk models	32
References	33
<b>4 Measuring risks</b>	<b>34</b>
4.1 Introduction	34
4.2 Synopsis of risk measures	34
4.3 Portfolio risk concepts	39
References	41
<b>5 Modern portfolio theory</b>	<b>43</b>
5.1 Introduction	43

5.2	Markowitz portfolios	43
5.3	Empirical mean–variance portfolios	47
	References	49
<b>Part II RISK MODELLING</b>		<b>51</b>
<b>6</b>	<b>Suitable distributions for returns</b>	<b>53</b>
6.1	Preliminaries	53
6.2	The generalized hyperbolic distribution	53
6.3	The generalized lambda distribution	56
6.4	Synopsis of R packages for the GHD	62
6.4.1	The package <b>fBasics</b>	62
6.4.2	The package <b>GeneralizedHyperbolic</b>	63
6.4.3	The package <b>ghyp</b>	64
6.4.4	The package <b>QRM</b>	65
6.4.5	The package <b>SkewHyperbolic</b>	66
6.4.6	The package <b>VarianceGamma</b>	67
6.5	Synopsis of R packages for GLD	67
6.5.1	The package <b>Davies</b>	67
6.5.2	The package <b>fBasics</b>	67
6.5.3	The package <b>gld</b>	68
6.5.4	The package <b>lmomco</b>	69
6.6	Applications of the GHD to risk modelling	69
6.6.1	Fitting stock returns to the GHD	69
6.6.2	Risk assessment with the GHD	73
6.6.3	Stylized facts revisited	75
6.7	Applications of the GLD to risk modelling and data analysis	78
6.7.1	VaR for a single stock	78
6.7.2	Shape triangle for FTSE 100 constituents	79
	References	82
<b>7</b>	<b>Extreme value theory</b>	<b>84</b>
7.1	Preliminaries	84
7.2	Extreme value methods and models	85
7.2.1	The block maxima approach	85
7.2.2	$r$ th largest order models	86
7.2.3	The peaks-over-threshold approach	87
7.3	Synopsis of R packages	89
7.3.1	The package <b>evd</b>	89
7.3.2	The package <b>evdbayes</b>	90
7.3.3	The package <b>evir</b>	91



7.3.4	The package <b>fExtremes</b>	93
7.3.5	The packages <b>ismev</b> and <b>extRemes</b>	95
7.3.6	The package <b>POT</b>	96
7.3.7	The package <b>QRM</b>	97
7.3.8	The package <b>Renext</b>	97
7.4	Empirical applications of EVT	98
7.4.1	Section outline	98
7.4.2	Block maxima model for Siemens	99
7.4.3	$r$ block maxima model for BMW	101
7.4.4	POT method for Boeing	105
	References	110
<b>8</b>	<b>Modelling volatility</b>	<b>112</b>
8.1	Preliminaries	112
8.2	The class of ARCH models	112
8.3	Synopsis of R packages	116
8.3.1	The package <b>bayesGARCH</b>	116
8.3.2	The package <b>cggarch</b>	117
8.3.3	The package <b>fGarch</b>	118
8.3.4	The package <b>gogarch</b>	118
8.3.5	The packages <b>rugarch</b> and <b>rmgarch</b>	120
8.3.6	The package <b>tseries</b>	122
8.4	Empirical application of volatility models	123
	References	125
<b>9</b>	<b>Modelling dependence</b>	<b>127</b>
9.1	Overview	127
9.2	Correlation, dependence and distributions	127
9.3	Copulae	130
9.3.1	Motivation	130
9.3.2	Correlations and dependence revisited	131
9.3.3	Classification of copulae	133
9.4	Synopsis of R packages	136
9.4.1	The package <b>BLCOP</b>	136
9.4.2	The packages <b>copula</b> and <b>nacopula</b>	138
9.4.3	The package <b>fCopulae</b>	140
9.4.4	The package <b>gumbel</b>	141
9.4.5	The package <b>QRM</b>	142
9.5	Empirical applications of copulae	142
9.5.1	GARCH–copula model	142
9.5.2	Mixed copula approaches	149
	References	151

<b>Part III</b>	<b>PORTFOLIO OPTIMIZATION APPROACHES</b>	<b>153</b>
<b>10</b>	<b>Robust portfolio optimization</b>	<b>155</b>
10.1	Overview	155
10.2	Robust statistics	156
10.2.1	Motivation	156
10.2.2	Selected robust estimators	157
10.3	Robust optimization	160
10.3.1	Motivation	160
10.3.2	Uncertainty sets and problem formulation	160
10.4	Synopsis of R packages	166
10.4.1	The package <b>covRobust</b>	166
10.4.2	The package <b>fPortfolio</b>	166
10.4.3	The package <b>MASS</b>	167
10.4.4	The package <b>robustbase</b>	168
10.4.5	The package <b>robust</b>	168
10.4.6	The package <b>rrcov</b>	169
10.4.7	The package <b>Rsocp</b>	170
10.5	Empirical applications	171
10.5.1	Portfolio simulation: Robust versus classical statistics	171
10.5.2	Portfolio back-test: Robust versus classical statistics	177
10.5.3	Portfolio back-test: Robust optimization	182
	References	187
<b>11</b>	<b>Diversification reconsidered</b>	<b>189</b>
11.1	Introduction	189
11.2	Most diversified portfolio	190
11.3	Risk contribution constrained portfolios	192
11.4	Optimal tail-dependent portfolios	195
11.5	Synopsis of R packages	197
11.5.1	The packages <b>DEoptim</b> and <b>RcppDE</b>	197
11.5.2	The package <b>FRAPO</b>	199
11.5.3	The package <b>PortfolioAnalytics</b>	201
11.6	Empirical applications	201
11.6.1	Comparison of approaches	201
11.6.2	Optimal tail-dependent portfolio against benchmark	206
11.6.3	Limiting contributions to expected shortfall	211
	References	215
<b>12</b>	<b>Risk-optimal portfolios</b>	<b>217</b>
12.1	Overview	217
12.2	Mean–VaR portfolios	218
12.3	Optimal CVaR portfolios	223
12.4	Optimal draw-down portfolios	227

12.5	Synopsis of R packages	229
12.5.1	The package <b>fPortfolio</b>	229
12.5.2	The package <b>FRAPO</b>	230
12.5.3	Packages for linear programming	232
12.5.4	The package <b>PerformanceAnalytics</b>	236
12.6	Empirical applications	238
12.6.1	Minimum-CVaR versus minimum-variance portfolios	238
12.6.2	Draw-down constrained portfolios	242
12.6.3	Back-test comparison for stock portfolio	247
	References	253
<b>13</b>	<b>Tactical asset allocation</b>	<b>255</b>
13.1	Overview	255
13.2	Survey of selected time series models	256
13.2.1	Univariate time series models	256
13.2.2	Multivariate time series models	262
13.3	Black–Litterman approach	270
13.4	Copula opinion and entropy pooling	273
13.4.1	Introduction	273
13.4.2	The COP model	273
13.4.3	The EP model	274
13.5	Synopsis of R packages	276
13.5.1	The package <b>BLCOP</b>	276
13.5.2	The package <b>dse</b>	278
13.5.3	The package <b>fArma</b>	281
13.5.4	The package <b>forecast</b>	281
13.5.5	The package <b>MSBVAR</b>	283
13.5.6	The package <b>PairTrading</b>	284
13.5.7	The packages <b>urca</b> and <b>vars</b>	285
13.6	Empirical applications	288
13.6.1	Black–Litterman portfolio optimization	288
13.6.2	Copula opinion pooling	295
13.6.3	Protection strategies	299
	References	310
<b>Appendix A</b>	<b>Package overview</b>	<b>314</b>
A.1	Packages in alphabetical order	314
A.2	Packages ordered by topic	317
	References	320
<b>Appendix B</b>	<b>Time series data</b>	<b>324</b>
B.1	Date-time classes	324
B.2	The <b>ts</b> class in the base package stats	327
B.3	Irregular-spaced time series	328

B.4	The package timeSeries	330
B.5	The package zoo	332
B.6	The packages tframe and xts	334
	References	337
<b>Appendix C</b>	<b>Back-testing and reporting of portfolio strategies</b>	<b>338</b>
C.1	R packages for back-testing	338
C.2	R facilities for reporting	339
C.3	Interfacing databases	339
	References	340
<b>Appendix D</b>	<b>Technicalities</b>	<b>342</b>
	<b>Index</b>	<b>343</b>

# Preface

The project for this book began in mid-2010. At that time, financial markets were in distress and far from operating smoothly. The impact of the US real estate crisis could still be felt and the sovereign debt crisis in some European countries was beginning to emerge. Major central banks implemented measures to avoid a collapse of the inter-bank market by providing liquidity. Given the massive financial book and real losses sustained by investors, it was also a time when quantitatively managed funds were in jeopardy and investors questioned the suitability of quantitative methods for protecting their wealth from the severe losses they had made in the past.

Two years later not much has changed, though the debate on whether quantitative techniques *per se* are limited has ceased. Hence, the modelling of financial risks and the adequate allocation of wealth is still as important as it always has been, and these topics have gained in importance driven by experiences since the financial crisis started in the latter part of the previous decade.

The content of the book is aimed at these two topics by acquainting and familiarizing the reader with market risk models and portfolio optimization techniques that have been proposed in the literature. These more recently proposed methods are elucidated by code examples written in the **R** language, a freely available software environment for statistical computing.

This book certainly could not have been written without the public provision of such a superb piece of software as **R** and the numerous package authors who have greatly enriched this software environment. I therefore wish to express my sincere appreciation and thanks to the **R** Core Team members and all the contributors and maintainers of the packages cited and utilized in this book. By the same token, I would like to apologize to those authors whose packages I have not mentioned. This can only be ascribed to my ignorance of their existence. Second, I would like to thank John Wiley & Sons, Ltd. for the opportunity to write on this topic, in particular Ilaria Meliconi who initiated this book project in the first place and Heather Kay and Richard Davies for their careful editorial work. A special thank belongs to Richard Leigh for his meticulous and mindful copy-editing. Needless to say, any errors and omissions are entirely my responsibility. Finally, I owe a debt of profound gratitude

to my beloved wife, Antonia, who while bearing the burden of many hours of solitude during the writing of this book remained a constant source of support.

This book includes an accompanying website. Please visit [www.wiley.com/go/financial\\_risk](http://www.wiley.com/go/financial_risk)

Bernhard Pfaff  
Kronberg im Taunus

# List of abbreviations

2OLS	Two-stage ordinary least-squares
3OLS	Three-stage ordinary least-squares
ACF	Autocorrelation function
ADF	Augmented Dickey–Fuller
AMPL	A modelling language for mathematical programming
ANSI	American National Standards Institute
AP	Active premium
APARCH	Asymmetric power ARCH
API	Application programming interface
ARCH	Autoregressive conditional heteroscedasticity
AvDD	Average draw-down
BL	Black–Litterman
BP	Break point
CDaR	Conditional draw-down at risk
CLI	Command line interface
CLT	Central limit theorem
COM	Component object model
COP	Copula opinion pooling
CPPI	Constant proportion portfolio insurance
CRAN	Comprehensive R Archive Network
CVaR	Conditional value at risk
DBMS	Data Base Management System
DD	Draw-down
DE	Differential evolution
DR	Diversification ratio
EDA	Exploratory data analysis
EGARCH	Exponential GARCH
EM	Expectation maximization
EMA	Exponentially weighted mean
EP	Entropy pooling
ERS	Elliott–Rothenberg–Stock
ES	Expected shortfall
EVT	Extreme value theory
FIML	Full-information maximum likelihood

GARCH	Generalized autoregressive conditional heteroscedasticity
GEV	Generalized extreme value
GHD	Generalized hyperbolic distribution
GIG	Generalized inverse Gaussian
GLD	Generalized lambda distribution
GLPK	GNU Linear Programming Kit
GMPL	GNU MathProg modelling language
GMV	Global minimum variance
GoF	Goodness of fit
GOGARCH	Generalized orthogonal GARCH
GPD	Generalized Pareto distribution
GUI	Graphical user interface
HYP	Hyperbolic
IDE	Integrated development environment
i.i.d.	independent and identically distributed
IR	Information ratio
JDBC	Java-based Data Access Technology
LP	Linear program
MaxDD	Maximum draw-down
MCD	Minimum covariance determinant
MCMC	Markov chain Monte Carlo
MDA	Maximum domain of attraction
mES	Modified expected shortfall
MILP	Mixed Integer Linear Program
ML	Maximum likelihood
MLE	Maximum likelihood estimation
MPS	Mathematical Programming System
MRL	Mean residual life
MSEM	Multiple-structural equation model
mVaR	Modified value at risk
MVE	Minimum volume ellipsoid
NIG	Normal inverse Gaussian
NN	Nearest neighbour
OBPI	Option-based portfolio insurance
ODBC	Open Data Base Connectivity
OGK	Orthogonalized Gnanadesikan–Kettenring
PACF	Partial autocorrelation function
POT	Peaks over threshold
PWM	Probability weighted moments
QMLE	Quasi-maximum likelihood estimation
RDBMS	Relational Data Base Management System
RE	Relative efficiency
RNG	Random number generator
RPC	Remote procedure call



RS	Ramberg–Schmeiser
SDE	Stahel–Donoho estimator
SMA	Simple moving average
SMEM	Structural multiple equation model
SPI	Swiss Performance Index
SVAR	Structural vector-autoregressive model
SVEC	Structural vector-error-correction model
TAA	Tactical asset allocation
TDC	Tail dependence coefficient
TE	Tracking error
VAR	Vector autoregressive model
VaR	Value at risk
VECM	Vector error correction model
WMA	Weighted moving average
XML	Extended Mark-up Language

Unless otherwise stated, the following notation, symbols and variables are used.

#### Notation:

Bold lower case: $\mathbf{y}, \alpha$	Vectors
Upper case: $Y, \Sigma$	Matrices
Greek letters: $\alpha, \beta, \gamma$	Scalars
Greek letters with $\hat{\cdot}$ or $\tilde{\cdot}$	Sample values (estimates or estimators)

#### Symbols and Variables:

$ \cdot $	Absolute value of an expression
$\sim$	Distributed according to
$\otimes$	Kronecker product of two matrices
arg max	Maximum value of an argument
arg min	Minimum value of an argument
$\perp$	Complement of a matrix
$\mathcal{C}, c$	Copula
Cor	Correlation(s) of an expression
Cov	Variance–covariance of an expression
$\mathbb{D}$	Draw-down
det	Determinant of a matrix
E	Expectation operator
$\mathcal{I}$	Information set
$I(d)$	Integrated of order $d$
L	Lag operator

$\mathcal{L}$	(Log)-likelihood function
$\mu$	Expected value
$\mathcal{N}$	Normal distribution
$\omega$	Weight vector
P	Portfolio problem specification
P	Probability expression
$\mathbb{R}$	Set of real numbers
$\Sigma$	Variance–covariance matrix
$\sigma$	Standard deviation
$\sigma^2$	Variance
$\mathcal{U}$	Uncertainty set
Var	Variance of an expression

# **Part I**

## **MOTIVATION**



# 1

## Introduction

The period since the late 1990s has been marked by financial crises – the Asian crisis of 1997, the Russian debt crisis of 1998, the bursting of the dot-com bubble in 2000, the crises following the attack on the World Trade Center in 2001 and the invasion of Iraq in 2003, the sub-prime mortgage crisis of 2007 and European sovereign debt crisis since 2009 being the most prominent. All of these crises had a tremendous impact on the financial markets, in particular an upsurge in observed volatility and a massive destruction of financial wealth. During most of these episodes the stability of the financial system was in jeopardy and the major central banks were more or less obliged to take counter-measures, as were the governments of the relevant countries. Of course, this is not to say that the time prior to the late 1990s was tranquil – in this context we may mention the European Currency Unit crisis in 1992–1993 and the crash on Wall Street in 1987, known as Black Monday. However, it is fair to say that the frequency of occurrence of crises has increased during the last 15 years.

Given this rise in the frequency of crises, the modelling and measurement of financial market risk have gained tremendously in importance and the focus of portfolio allocation has shifted from the  $\mu$  side of the  $(\mu, \sigma)$  medal to its  $\sigma$  side. Hence, it has become necessary to devise and employ methods and techniques that are better able to cope with the empirically observed extreme fluctuations in the financial markets. The hitherto fundamental assumption of independent and identically normally distributed financial market returns is no longer sacrosanct, having been challenged by statistical models and concepts that take the occurrence of extreme events more adequately into account than the Gaussian model assumption does. As will be shown in the following chapters, the more recently proposed methods of and approaches to wealth allocation are not of a revolutionary kind, but can be seen as an evolutionary development: a recombination and application of already existing statistical concepts to solve finance-related problems. Sixty years after Markowitz's seminal paper 'Modern Portfolio Theory', the key  $(\mu, \sigma)$  paradigm must still be considered as the anchor for portfolio optimization. What has been changed by the more recently advocated

approaches, however, is how the riskiness of an asset is assessed and how portfolio diversification, that is, the dependencies between financial instruments, is measured, and the definition of the portfolio's objective *per se*.

The purpose of this book is to acquaint the reader with some of these recently proposed approaches. Given the length of the book this synopsis must be selective, but the topics chosen are intended to cover a broad spectrum. In order to foster the reader's understanding of these advances, all the concepts introduced are elucidated by practical examples. This is accomplished by means of the R language, a free statistical computing environment (see R Development Core Team 2012). Therefore, almost regardless of the reader's computer facilities in terms of hardware and operating system, all the code examples can be replicated at the reader's desk and s/he is encouraged not only to do so, but also to adapt the code examples to her/his own needs. This book is aimed at the quantitatively inclined reader with a background in finance, statistics and mathematics at upper undergraduate/graduate level. The text can also be used as an accompanying source in a computer lab class, where the modelling of financial risks and/or portfolio optimization are of interest.

The book is divided into three parts. The chapters of this first part are primarily intended to provide an overview of the topics covered in later chapters and serve as a motivation for applying techniques beyond those commonly encountered in assessing financial market risks and/or portfolio optimization. Chapter 2 provides a brief course in the R language and presents the **FRAPO** package accompanying the book. For the reader completely unacquainted with R, this chapter cannot replace a more dedicated course of study of the language itself, but it is rather intended to provide a broad overview of R and how to obtain help. Because in the book's examples quite a few R packages will be presented and utilized, a section on the existing classes and methods is included that will ease the reader's comprehension of these two frameworks. In Chapter 3 stylized facts of univariate and multivariate financial market data are presented. The exposition of these empirical characteristics serves as motivation for the methods and models presented in Part II. Definitions used in the measurement of financial market risks at the single-asset and portfolio level are the topic of the Chapter 4. In the final chapter of Part I (Chapter 5), the Markowitz portfolio framework is described and empirical artefacts of the accordingly optimized portfolios are presented. The latter serve as a motivation for the alternative portfolio optimization techniques presented in the Part III.

In Part II, alternatives to the normal distribution assumption for modelling and measuring financial market risks are presented. This part commences with an exposition of the generalized hyperbolic and generalized lambda distributions for modelling returns of financial instruments. In Chapter 7, the extreme value theory is introduced as a means of modelling and capturing severe financial losses. Here, the block-maxima and peaks-over-threshold approaches are described and applied to stock losses. Both Chapters 6 and 7 have the unconditional modelling of financial losses in common. The conditional modelling and measurement of financial market risks is presented in the form of GARCH models – defined in the broader sense – in Chapter 8. Part II concludes with a chapter on copulae as a means of modelling the dependencies between assets.

Part III commences by introducing robust portfolio optimization techniques as a remedy to the outlier sensitivity encountered by plain Markowitz optimization. In Chapter 10 it is shown how robust estimators for the first and second moments can be used as well as portfolio optimization methods that directly facilitate the inclusion of parameter uncertainty. In Chapter 11 the concept of portfolio diversification is reconsidered. In this chapter the portfolio concepts of the most diversified, equal risk contributed and minimum tail-dependent portfolios are described. In Chapter 12 the focus shifts to downside-related risk measures, such as the conditional value at risk and the draw-down of a portfolio. Chapter 13 is devoted to tactical asset allocation (TAA). Aside from the original Black–Litterman approach, the concept of copula opinion pooling and the construction of a wealth protection strategy are described. The latter is a synthesis between the topics presented in Part II and TAA-related portfolio optimization.

In the Appendix all the R packages cited and used are listed by name and topic. Due to alternative means of handling longitudinal data in R, a separate chapter in the Appendix is dedicated to the presentation of the available classes and methods. In Appendix C it is shown how R can be invoked and employed on a regular basis for producing back-tests, utilized for generating or updating reports and/or embedded in an existing IT infrastructure for risk assessment/portfolio rebalancing. Because all of these topics are highly custom-specific, only pointers to the R facilities are provided. A section on the technicalities concludes the book.

The chapters in Parts Two and Three adhere to a common structure. First the methods and/or models are presented from a theoretical viewpoint only. The following section is reserved for the presentation of R packages and the last section in each chapter contains applications of the concepts and methods previously presented. The R code examples provided are written at an intermediate language level and are intended to be digestible and easy to follow. Each code example could certainly be improved in terms of profiling and the accomplishment of certain computations, but at the risk of too cryptic a code design. It is left to the reader as an exercise to adapt and/or improve the examples to her/his own needs and preferences.

All in all, the aim of this book is to enable the reader to go beyond the ordinarily encountered standard tools and techniques and provide some guidance on when to choose among them. Each quantitative model certainly has its strengths and drawbacks and it is still a subjective matter whether the former outweigh the latter when it comes to employing the model in managing financial market risks and/or allocating wealth at hand. That said, it is better to have a larger set of tools available than to be forced to rely on a more restricted set of methods.

## Reference

R Development Core Team 2012 *R: A Language and Environment for Statistical Computing*  
R Foundation for Statistical Computing Vienna, Austria. ISBN 3-900051-07-0.

# 2

## A brief course in R

### 2.1 Origin and development

R is mainly a programming environment for conducting statistical computations and producing high-level graphics (see R Development Core Team 2012). These two areas of application should be interpreted widely, and indeed many tasks that one would not normally directly subsume under these topics can be accomplished with the R language. The website of the R project is <http://www.r-project.org>. The source code of the software is published as free software under the terms of the GNU General Public License (GPL; see <http://www.gnu.org/licenses/gpl.html>).

The R language is a dialect of the S language, which was developed by John Chambers and colleagues at Bell Labs in the mid-1970s.<sup>1</sup> At that time the software was implemented as FORTRAN libraries. A major advancement of the S language took place in 1988, following which the system was rewritten in C and functions for conducting statistical analysis were added. This was version 3 of the S language, referred to as S3 (see Becker *et al.* 1988; Chambers and Hastie 1992). At that stage in the development of S, the R story commences (see Gentleman and Ihaka 1997). In August 1993 Ross Ihaka and Robert Gentleman, both affiliated with the University of Auckland, New Zealand, released a binary copy of R on Statlib, announcing it on the *s-news* mailing list. This first R binary was based on a Scheme interpreter with an S-like syntax (see Ihaka and Gentleman 1996). The name of R traces back to the initials of the first names of Ihaka and Gentleman and is by coincidence a one-letter abbreviation to the language in the same way as S is. The announcement by Ihaka and Gentleman did not go unnoticed and credit is due to Martin Mächler of ETH Zürich, who persistently advocated the release of R under GNU's GPL. This then happened in June 1995. Interest in the language grew by word of mouth, and as a first means of

---

<sup>1</sup> A detailed account of the history of the S language is accessible at <http://www.stat.bell-labs.com/S/history.html>.



communication and coordination a mailing list was established in March 1996 which was then replaced a year later by the electronic mail facilities that still exist today. The growing interest in the project led to the need for a powerful distribution channel for the software. This was accomplished by Kurt Hornik, at that time affiliated to the Technische Universität in Vienna. The master repository for the software (known as the ‘Comprehensive R Archive Network’ or CRAN) is still located in Vienna, albeit now at the Wirtschaftsuniversität, and mirror server are spread all over the globe. In order to keep pace with requested changes by users and the fixing of bugs in a timely manner, a core group of R developers was set up in mid-1997. This established framework and infrastructure is probably the reason why R has since made such tremendous further progress. Users can contribute packages to solve specific problems or tasks and hence advances in statistical methods and/or computations can be swiftly disseminated. A detailed analysis and synopsis of the social organization and development of R is provided by Fox (2009). The next milestone in the history of language was in 1998, when John Chambers introduced a more formal class and method framework for the S language (version 4), which was then adopted in R (see Chambers 1998, 2008). This evolution explains the coexistence of S3- and S4-like structures in the R language, and the user will meet them both in Section 2.4. More recent advancements are the inclusion of support for high-performance computations and a byte code compiler for R. From these humble beginnings, R has become the *lingua franca* for statistical computing.

## 2.2 Getting help

It is beyond the scope of this book to provide the reader with an introduction to the R language itself. Those who are completely new to R are referred to the manual *An Introduction to R*, available on the project’s website under ‘Manuals’. The purpose of this section is rather to provide the reader with some pointers on obtaining help and retrieving the relevant information for solving a problem at hand.

As already indicated in the previous paragraph, the first resort for obtaining help is by reading the R manuals. These manuals cover different aspects of R and the one mentioned above provides a useful introduction to R. The following R manuals are available, and their titles are self-explanatory:

- *An Introduction to R*
- *The R Language Definition*
- *Writing R Extensions*
- *R Data Import/Export*
- *R Installation and Administration*
- *R Internals*
- *The R Reference Index*

These manuals can either be accessed from the project's website or invoked from an R session by typing

```
> help.start()
```

This function will load an HTML index file into the user's web browser and local links to these manuals appear at the top. Note that a link to the 'Frequently Asked Questions' is included, as well as a 'Windows FAQ' if R has been installed under Microsoft Windows.

Incidentally, in addition to these R manuals, many complementary tutorials and related material can be accessed from <http://www.r-project.org/other-docs.html> and an annotated listing of more than 100 books on R is available at <http://www.r-project.org/doc/bib/R-books.html>. The reader is also pointed to the *The R Journal* (formerly *R News*), which is a biannual publication covering the latest developments in R and consists of articles contributed by users.

Let us return to the subject of invoking help within R itself. As shown above, the function `help.start()` as invoked from the R prompt is one of the in-built help facilities that R offers. Other means of accessing help are:

```
> ## Invoking the manual page of help() itself
> help()
> ## Help on how to search in the help system
> help("help.search")
> ## Help on search by partial matching
> help("apropos")
> ## Displaying available demo files
> demo()
> demo(scoping)
> ## Displaying available package vignettes
> ?vignette
> vignette()
> vignette("parallel")
```

The first command will invoke the help page for `help()` itself; its usage is described therein and pointers given to other help facilities. Among these other facilities are `help.search()`, `apropos()`, and `demo()`. If the latter is executed without arguments, the available demonstration files are displayed and `demo(scoping)` then runs the R code for familiarizing the user with the concept of lexical scoping in R, for instance. More advanced help is provided in vignettes associated with packages. The purpose of these documents is to show the user how the functions and facilities of a package can be employed. These documents can be opened in either a PDF reader or a web browser. In the last code line, the vignette contained in the **parallel** package is opened and the user is given a detailed description of how parallel computations can be carried out with R.

A limitation of these help facilities is, that with these functions only local searches are conducted, so that the results returned depend on the R installation itself and the contributed packages installed. To conduct an on-line search the function

`RSiteSearch()` is available which includes searches in the R mailing lists (mailing lists will be covered as another means of getting help in due course).

```
> ## Online search facilities
> ?RSiteSearch
> RSiteSearch("Portfolio")
> ## The CRAN package sos
> ## 1. Installation
> install.package("sos")
> ## 2. Loading
> library(sos)
> ## 3. Getting an overview of the content
> help(package = sos)
> ## 4. Opening the package's vignette
> vignette("sos")
> ## 5. Getting help on findFn
> ?findFn
> ## 6. Searching online for 'Portfolio'
> findFn("Portfolio")
```

A very powerful tool for conducting on-line searches is the `sos` package (see Graves *et al.* 2011). If the reader has not installed this contributed package by now, s/he is recommended to do so. The cornerstone function is `findFn()` by which on-line searches are conducted. In the example above, all relevant entries with respect to the keyword 'Portfolio' are returned into a browser window and the rightmost column contains a description of the entries with a direct web link.

As shown above, `findFn()` can be used for answering questions of the form 'Can this be achieved with R?' or 'Has this already been implemented in R?'. In this respect, given that at the time of this writing almost 3700 packages are available on CRAN (not to speak of R-Forge), the 'task view' concept is beneficial. CRAN packages that fit into a certain category, say 'Finance', are grouped together and each is briefly described by the maintainer of the task view in question. Hence, the burden of searching the archive for a certain package with which a problem or task can be solved has been greatly reduced. Not only do the task views provide a good overview of what is available, but with the CRAN package `ctv` (see Zeileis 2005) the user can choose to install either the complete set of packages in a task view along with their dependencies or just those considered to be core packages. A listing of the task views can be found at <http://cran.r-project.org/web/views/>.

```
> install.packages("ctv")
> library(ctv)
> install.views("Finance")
```

As mentioned above, mailing lists are available, where users can post their problem/question to a wide audience. An overview of those available is provided at <http://www.r-project.org/mail.html>. Probably of most interest are `R-help` and `R-SIG-Finance`. The former is a high-traffic list dedicated to general questions

about R and the latter is focused on finance-related problems. In either case, before submitting to these lists the user should adhere to the posting guidelines., which can be found at <http://www.r-project.org/posting-guide.html>.

This section concludes with an overview of R conferences that have taken place in the past and will most likely come around again in the future.

- useR!: This is an international R user conference and consists of keynote lectures and user-contributed presentations which are grouped together by topic. Finance-related sessions are ordinarily a part of these topics. The conference started in 2004 on a biannual schedule in Vienna, but now takes place every year at a different location. For more information, see the announcement at <http://www.r-project.org>.
- R/Rmetrics: This annual conference started in 2007 and is solely dedicated to finance-related subjects. The conference has recently been organized as a workshop with tutorial sessions in the morning and user presentations in the afternoon. The venue is Meielisalp, Lake Thune, Switzerland and the conference usually takes place in the third week of June. More information is provided at <http://www.rmetrics.org>.
- R in Finance: Akin to the R/Rmetrics workshop, this conference is also solely dedicated to finance-related topics. It is a two-day event held annually during spring in Chicago at the University of Illinois. Optional pre-conference tutorials are given and the main conference consists of keynote speeches and user-contributed presentations.
- DSC: DSC stands for ‘Directions in Statistical Computing’ and, as its name indicates, is targeted at developers of statistical software. As such, the conference is not confined to R itself, though the lion’s share of topics do relate to advances in this language.

## 2.3 Working with R

By default, R is provided with a command line interface (CLI). At first sight, this might be perceived as a limitation and as an antiquated software design. This perception might be intensified for novice users of R. However, the CLI is a very powerful tool that gives the user direct control over calculations. The dilemma is that most probably only experienced users of R with a good command on the language might share this view on working with R, but do you become a proficient R user in the first place? In order to solve this puzzle and ease the new user’s way on this learning path, several graphical user interfaces (GUIs) and/or integrated development environments (IDEs) are available. Incidentally, it is possible to make this rather rich set of eye-catching GUIs and IDEs available because R is provided with a CLI in the first place, and all of them are factored around it.

In this section some of the platform-independent GUIs and IDEs are presented, acknowledging the fact that R is shipped with a GUI on the Microsoft Windows

operating system only. The listing below is in alphabetical order and does not advocate the usage of one GUI/IDE framework over another, for good reasons. Deciding which system to use is a matter of personal preference and taste. Therefore, the reader is invited to inspect each of the GUIs and IDEs presented and then choose whichever is to his liking.

1. Eclipse: Eclipse is a Java-based IDE and was first designed as an IDE for this language. More information about Eclipse are available on the project's website at <http://www.eclipse.org/>. Since then many modules/plug-ins for other languages have been made available. The plug-in for R is called `StatET` and is distributed via <http://www.walware.de/goto/statet>. Instructions for installing and configuring this module into Eclipse can be found on this website. Further on-line guidance is available elsewhere.
2. Emacs/ESS: GNU Emacs is an extensible and customisable text editor, which at its core is an interpreter for the Emacs Lisp language. The project's website is <http://www.gnu.org/software/emacs/>. Derived from this editor are the distributions XEmacs (<http://www.xemacs.org>), where the 'X' indicates the X window system of Unix/Linux platforms, and Aquamacs (<http://aquamacs.org/>) for Mac OS X only. Similar to Eclipse, the connection between this editor and R is established by means of a module, ESS, which stands for Emacs Speaks Statistics. The project's website is <http://ess.r-project.org/> where this Lisp module can be downloaded and installation instructions are available. A strength of ESS is that other statistical packages such as S-PLUS, SAS, Stata, OpenBUGS and JAGS are also supported. A dedicated mailing list for ESS is available in the 'Getting help' section of the website cited above. Users working in Microsoft Windows might be interested in the prepackaged Emacs/ESS version made available by Vincent Goulet: <http://vgoulet.act.ulaval.ca/en/emacs/>.
3. JGR: In contrast to Eclipse and Emacs/ESS, JGR is a GUI rather than an IDE for R. Like Eclipse, it is based on Java, and 'JGR' stands for Java Gui for R. The project's website is <http://rforge.net/JGR>, where installation instructions can be found for Microsoft Windows, Mac OS X and Linux platforms.
4. RStudio: The latest addition to platform-independent GUIs/IDEs is RStudio. The software is hosted at <http://www.rstudio.org/> and is distributed under the AGPLv3 license. A feature of this IDE is that it can be either installed as a desktop application or run on a server, where users can access RStudio via a web browser.
5. Vim: Last, but not least, there is an R plug-in for the Vim editor available. The Vim editor itself has been available for more than twenty years. The software is hosted at <http://www.vim.org/> and is based on the Unix vi editor. The R plug-in is contained in the section 'Scripts'.

Further information about R GUIs and IDEs can be found at [http://www.sciviews.org/\\_rgui/](http://www.sciviews.org/_rgui/), where a synopsis of available GUIs and IDEs is provided, some of which are platform-dependent. This is quite an extensive listing, and software solutions that might not appear as a GUI, such as a web service, are included too. Furthermore, the reader can subscribe to a special interest group mailing list, R-SIG-GUI, by following the instructions at <https://stat.ethz.ch/mailman/listinfo/r-sig-gui>.

## 2.4 Classes, methods and functions

In this section a concise introduction to the two flavours of class and method definitions in R is provided. The first class and method mechanism is referred to as S3 and the second as S4. Because the S4 classes and methods were included in R at a later stage in its development cycle (since version 1.4.0), S3 classes and methods are sometimes also called old-style classes and methods. Detailed accounts of these class and method schemes are provided in Becker *et al.* (1988) and Chambers and Hastie (1992) for S3 and in Chambers (1998, 2008) for S4. In addition to these sources, the manual pages for classes and methods can be inspected by typing `?Classes` and `?Methods`, respectively. The need to familiarize oneself with these class and method concepts is motivated by the fact that nowadays contributed R packages utilize either one or the other concept and in some packages a link between the class and method definitions of either kind is established. It should be noted that there are also R packages in which neither concept of object-oriented programming has been employed at all in the sense of new class and/or method definitions, and such packages can be viewed as collections of functions only.

Before each of the two concepts is discussed, the reader should recall that everything in R is an object, that a class is the definition of an object, and that a method is a function by which a predefined calculation or manipulation is carried out on the object. Furthermore, there are generic functions which have the sole purpose of determining the class of the object and associating the appropriate method to it. If no specific method can be associated to an object, a default method will be called as a fall-back. Generic functions can therefore be viewed as an umbrella under which all available class-specific methods are collected. The difference between S3 and S4 classes/methods lies in how a certain class is associated to an object and how a method is dispatched to it.

Because R is a dialect of the S language, S3 objects, classes and methods have been available since the very beginning of R, almost 20 years ago. The assignment of a class name and the method dispatch in this scheme are rather informal and hence very simple to implement. No formal requirements on the structure of an S3 class objects are necessary, it is just a matter of adding a class attribute to an object. How swiftly such an attribute can be added to an object and/or changed is shown in the following in-line example:

```
> x <- 1:5
> x
```