

Springer Texts in Statistics

David Ruppert
David S. Matteson

Statistics and Data Analysis for Financial Engineering

with R examples

Second Edition



Springer Texts in Statistics

Series Editors:

R. DeVeaux
S.E. Fienberg
I. Olkin

More information about this series at <http://www.springer.com/series/417>

David Ruppert • David S. Matteson

Statistics and Data Analysis for Financial Engineering

with R examples

Second Edition



Springer

David Ruppert
Department of Statistical
Science and School of ORIE
Cornell University
Ithaca, NY, USA

David S. Matteson
Department of Statistical Science
Department of Social Statistics
Cornell University
Ithaca, NY, USA

ISSN 1431-875X
Springer Texts in Statistics
ISBN 978-1-4939-2613-8
DOI 10.1007/978-1-4939-2614-5

ISSN 2197-4136 (electronic)
ISBN 978-1-4939-2614-5 (eBook)

Library of Congress Control Number: 2015935333

Springer New York Heidelberg Dordrecht London
© Springer Science+Business Media New York 2011, 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer Science+Business Media LLC New York is part of Springer Science+Business Media (www.springer.com)

To Susan

David Ruppert

To my grandparents

David S. Matteson

Preface

The first edition of this book has received a very warm reception. A number of instructors have adopted this work as a textbook in their courses. Moreover, both novices and seasoned professionals have been using the book for self-study. The enthusiastic response to the book motivated a new edition. One major change is that there are now two authors. The second edition improves the book in several ways: all known errors have been corrected and changes in R have been addressed. Considerably more R code is now included. The GARCH chapter now uses the `rugarch` package, and in the Bayes chapter we now use JAGS in place of OpenBUGS.

The first edition was designed primarily as a textbook for use in university courses. Although there is an Instructor's Manual with solutions to all exercises and all problems in the R labs, this manual has been available only to instructors. No solutions have been available for readers engaged in self-study. To address this problem, the number of exercises and R lab problems has increased and the solutions to many of them are being placed on the book's web site.

Some data sets in the first edition were in R packages that are no longer available. These data sets are also on the web site. The web site also contains R scripts with the code used in the book.

We would like to thank Peter Dalgaard, Guy Yollin, and Aaron Fox for many helpful suggestions. We also thank numerous readers for pointing out errors in the first edition.

The book's web site is <http://people.orie.cornell.edu/davidr/SDAFE2/index.html>.

Ithaca, NY, USA
Ithaca, NY, USA
January 2015

David Ruppert
David S. Matteson

Preface to the First Edition

I developed this textbook while teaching the course *Statistics for Financial Engineering* to master's students in the financial engineering program at Cornell University. These students have already taken courses in portfolio management, fixed income securities, options, and stochastic calculus, so I concentrate on teaching statistics, data analysis, and the use of R, and I cover most sections of Chaps. 4–12 and 18–20. These chapters alone are more than enough to fill a one-semester course. I do not cover regression (Chaps. 9–11 and 21) or the more advanced time series topics in Chap. 13, since these topics are covered in other courses. In the past, I have not covered cointegration (Chap. 15), but I will in the future. The master's students spend much of the third semester working on projects with investment banks or hedge funds. As a faculty adviser for several projects, I have seen the importance of cointegration.

A number of different courses might be based on this book. A two-semester sequence could cover most of the material. A one-semester course with more emphasis on finance would include Chaps. 16 and 17 on portfolios and the CAPM and omit some of the chapters on statistics, for instance, Chaps. 8, 14, and 20 on copulas, GARCH models, and Bayesian statistics. The book could be used for courses at both the master's and Ph.D. levels.

Readers familiar with my textbook *Statistics and Finance: An Introduction* may wonder how that volume differs from this book. This book is at a somewhat more advanced level and has much broader coverage of topics in statistics compared to the earlier book. As the title of this volume suggests, there is more emphasis on data analysis and this book is intended to be more than just “an introduction.” Chapters 8, 15, and 20 on copulas, cointegration, and Bayesian statistics are new. Except for some figures borrowed from *Statistics and Finance*, in this book R is used exclusively for computations, data analysis, and graphing, whereas the earlier book used SAS and MATLAB. Nearly all of the examples in this book use data sets that are available in R, so readers can reproduce the results. In Chap. 20 on Bayesian statistics,

WinBUGS is used for Markov chain Monte Carlo and is called from R using the `R2WinBUGS` package. There is some overlap between the two books, and, in particular, a substantial amount of the material in Chaps. 2, 3, 9, 11–13, and 16 has been taken from the earlier book. Unlike *Statistics and Finance*, this volume does not cover options pricing and behavioral finance.

The prerequisites for reading this book are knowledge of calculus, vectors, and matrices; probability including stochastic processes; and statistics typical of third- or fourth-year undergraduates in engineering, mathematics, statistics, and related disciplines. There is an appendix that reviews probability and statistics, but it is intended for reference and is certainly not an introduction for readers with little or no prior exposure to these topics. Also, the reader should have some knowledge of computer programming. Some familiarity with the basic ideas of finance is helpful.

This book does not teach R programming, but each chapter has an “R lab” with data analysis and simulations. Students can learn R from these labs and by using R’s help or the manual *An Introduction to R* (available at the CRAN web site and R’s online help) to learn more about the functions used in the labs. Also, the text does indicate which R functions are used in the examples. Occasionally, R code is given to illustrate some process, for example, in Chap. 16 finding the tangency portfolio by quadratic programming. For readers wishing to use R, the bibliographical notes at the end of each chapter mention books that cover R programming and the book’s web site contains examples of the R and WinBUGS code used to produce this book. Students enter my course *Statistics for Financial Engineering* with quite disparate knowledge of R. Some are very accomplished R programmers, while others have no experience with R, although all have experience with some programming language. Students with no previous experience with R generally need assistance from the instructor to get started on the R labs. Readers using this book for self-study should learn R first before attempting the R labs.

Ithaca, NY, USA
July 2010

David Ruppert

Contents

Notation	xxv
1 Introduction	1
1.1 Bibliographic Notes	4
References	4
2 Returns	5
2.1 Introduction	5
2.1.1 Net Returns	5
2.1.2 Gross Returns	6
2.1.3 Log Returns	6
2.1.4 Adjustment for Dividends	7
2.2 The Random Walk Model	8
2.2.1 Random Walks	8
2.2.2 Geometric Random Walks	9
2.2.3 Are Log Prices a Lognormal Geometric Random Walk?	9
2.3 Bibliographic Notes	10
2.4 R Lab	11
2.4.1 Data Analysis	11
2.4.2 Simulations	13
2.4.3 Simulating a Geometric Random Walk	14
2.4.4 Let's Look at McDonald's Stock	15
2.5 Exercises	16
References	18
3 Fixed Income Securities	19
3.1 Introduction	19
3.2 Zero-Coupon Bonds	20
3.2.1 Price and Returns Fluctuate with the Interest Rate	20

3.3	Coupon Bonds	22
3.3.1	A General Formula	23
3.4	Yield to Maturity	23
3.4.1	General Method for Yield to Maturity	25
3.4.2	Spot Rates	25
3.5	Term Structure	26
3.5.1	Introduction: Interest Rates Depend Upon Maturity ..	26
3.5.2	Describing the Term Structure	27
3.6	Continuous Compounding	32
3.7	Continuous Forward Rates	33
3.8	Sensitivity of Price to Yield	35
3.8.1	Duration of a Coupon Bond	35
3.9	Bibliographic Notes	36
3.10	R Lab	37
3.10.1	Computing Yield to Maturity	37
3.10.2	Graphing Yield Curves	38
3.11	Exercises	40
	References	43
4	Exploratory Data Analysis	45
4.1	Introduction	45
4.2	Histograms and Kernel Density Estimation	47
4.3	Order Statistics, the Sample CDF, and Sample Quantiles	52
4.3.1	The Central Limit Theorem for Sample Quantiles	54
4.3.2	Normal Probability Plots	54
4.3.3	Half-Normal Plots	58
4.3.4	Quantile–Quantile Plots	61
4.4	Tests of Normality	64
4.5	Boxplots	65
4.6	Data Transformation	67
4.7	The Geometry of Transformations	71
4.8	Transformation Kernel Density Estimation	75
4.9	Bibliographic Notes	77
4.10	R Lab	77
4.10.1	European Stock Indices	77
4.10.2	McDonald's Prices and Returns	80
4.11	Exercises	81
	References	83
5	Modeling Univariate Distributions	85
5.1	Introduction	85
5.2	Parametric Models and Parsimony	85
5.3	Location, Scale, and Shape Parameters	86

5.4	Skewness, Kurtosis, and Moments	87
5.4.1	The Jarque–Bera Test	91
5.4.2	Moments	92
5.5	Heavy-Tailed Distributions	93
5.5.1	Exponential and Polynomial Tails	93
5.5.2	t -Distributions	94
5.5.3	Mixture Models	96
5.6	Generalized Error Distributions	99
5.7	Creating Skewed from Symmetric Distributions	101
5.8	Quantile-Based Location, Scale, and Shape Parameters	103
5.9	Maximum Likelihood Estimation	104
5.10	Fisher Information and the Central Limit Theorem for the MLE	105
5.11	Likelihood Ratio Tests	107
5.12	AIC and BIC	109
5.13	Validation Data and Cross-Validation	110
5.14	Fitting Distributions by Maximum Likelihood	113
5.15	Profile Likelihood	119
5.16	Robust Estimation	121
5.17	Transformation Kernel Density Estimation with a Parametric Transformation	123
5.18	Bibliographic Notes	126
5.19	R Lab	127
5.19.1	Earnings Data	127
5.19.2	DAX Returns	129
5.19.3	McDonald’s Returns	130
5.20	Exercises	131
	References	134
6	Resampling	137
6.1	Introduction	137
6.2	Bootstrap Estimates of Bias, Standard Deviation, and MSE	139
6.2.1	Bootstrapping the MLE of the t -Distribution	139
6.3	Bootstrap Confidence Intervals	142
6.3.1	Normal Approximation Interval	143
6.3.2	Bootstrap- t Intervals	143
6.3.3	Basic Bootstrap Interval	146
6.3.4	Percentile Confidence Intervals	146
6.4	Bibliographic Notes	150
6.5	R Lab	150
6.5.1	BMW Returns	150
6.5.2	Simulation Study: Bootstrapping the Kurtosis	152
6.6	Exercises	154
	References	156

7 Multivariate Statistical Models	157
7.1 Introduction	157
7.2 Covariance and Correlation Matrices	157
7.3 Linear Functions of Random Variables	159
7.3.1 Two or More Linear Combinations of Random Variables	161
7.3.2 Independence and Variances of Sums	162
7.4 Scatterplot Matrices	162
7.5 The Multivariate Normal Distribution	164
7.6 The Multivariate <i>t</i> -Distribution	165
7.6.1 Using the <i>t</i> -Distribution in Portfolio Analysis	167
7.7 Fitting the Multivariate <i>t</i> -Distribution by Maximum Likelihood	168
7.8 Elliptically Contoured Densities	170
7.9 The Multivariate Skewed <i>t</i> -Distributions	172
7.10 The Fisher Information Matrix	174
7.11 Bootstrapping Multivariate Data	175
7.12 Bibliographic Notes	177
7.13 R Lab	177
7.13.1 Equity Returns	177
7.13.2 Simulating Multivariate <i>t</i> -Distributions	178
7.13.3 Fitting a Bivariate <i>t</i> -Distribution	180
7.14 Exercises	181
References	182
8 Copulas	183
8.1 Introduction	183
8.2 Special Copulas	185
8.3 Gaussian and <i>t</i> -Copulas	186
8.4 Archimedean Copulas	187
8.4.1 Frank Copula	187
8.4.2 Clayton Copula	189
8.4.3 Gumbel Copula	191
8.4.4 Joe Copula	192
8.5 Rank Correlation	193
8.5.1 Kendall's Tau	194
8.5.2 Spearman's Rank Correlation Coefficient	195
8.6 Tail Dependence	196
8.7 Calibrating Copulas	198
8.7.1 Maximum Likelihood	199
8.7.2 Pseudo-Maximum Likelihood	199
8.7.3 Calibrating Meta-Gaussian and Meta- <i>t</i> -Distributions	200
8.8 Bibliographic Notes	207

8.9	R Lab	208
8.9.1	Simulating from Copula Models	208
8.9.2	Fitting Copula Models to Bivariate Return Data	210
8.10	Exercises	213
	References	214
9	Regression: Basics	217
9.1	Introduction	217
9.2	Straight-Line Regression	218
9.2.1	Least-Squares Estimation	218
9.2.2	Variance of $\hat{\beta}_1$	222
9.3	Multiple Linear Regression	223
9.3.1	Standard Errors, t -Values, and p -Values	225
9.4	Analysis of Variance, Sums of Squares, and R^2	227
9.4.1	ANOVA Table	227
9.4.2	Degrees of Freedom (DF)	229
9.4.3	Mean Sums of Squares (MS) and F -Tests	229
9.4.4	Adjusted R^2	231
9.5	Model Selection	231
9.6	Collinearity and Variance Inflation	233
9.7	Partial Residual Plots	240
9.8	Centering the Predictors	242
9.9	Orthogonal Polynomials	243
9.10	Bibliographic Notes	243
9.11	R Lab	243
9.11.1	U.S. Macroeconomic Variables	243
9.12	Exercises	245
	References	248
10	Regression: Troubleshooting	249
10.1	Regression Diagnostics	249
10.1.1	Leverages	251
10.1.2	Residuals	252
10.1.3	Cook's Distance	253
10.2	Checking Model Assumptions	255
10.2.1	Nonnormality	256
10.2.2	Nonconstant Variance	258
10.2.3	Nonlinearity	259
10.3	Bibliographic Notes	262
10.4	R Lab	263
10.4.1	Current Population Survey Data	263
10.5	Exercises	265
	References	268

11 Regression: Advanced Topics	269
11.1 The Theory Behind Linear Regression	269
11.1.1 Maximum Likelihood Estimation for Regression	270
11.2 Nonlinear Regression	271
11.3 Estimating Forward Rates from Zero-Coupon Bond Prices	276
11.4 Transform-Both-Sides Regression	281
11.4.1 How TBS Works	283
11.5 Transforming Only the Response	284
11.6 Binary Regression	286
11.7 Linearizing a Nonlinear Model	291
11.8 Robust Regression	293
11.9 Regression and Best Linear Prediction	295
11.9.1 Best Linear Prediction	295
11.9.2 Prediction Error in Best Linear Prediction	297
11.9.3 Regression Is Empirical Best Linear Prediction	298
11.9.4 Multivariate Linear Prediction	298
11.10 Regression Hedging	298
11.11 Bibliographic Notes	300
11.12 R Lab	300
11.12.1 Nonlinear Regression	300
11.12.2 Response Transformations	302
11.12.3 Binary Regression: Who Owns an Air Conditioner?	303
11.13 Exercises	304
References	305
12 Time Series Models: Basics	307
12.1 Time Series Data	307
12.2 Stationary Processes	307
12.2.1 White Noise	310
12.2.2 Predicting White Noise	311
12.3 Estimating Parameters of a Stationary Process	312
12.3.1 ACF Plots and the Ljung–Box Test	312
12.4 AR(1) Processes	314
12.4.1 Properties of a Stationary AR(1) Process	315
12.4.2 Convergence to the Stationary Distribution	316
12.4.3 Nonstationary AR(1) Processes	317
12.5 Estimation of AR(1) Processes	318
12.5.1 Residuals and Model Checking	318
12.5.2 Maximum Likelihood and Conditional Least-Squares	323
12.6 AR(p) Models	325
12.7 Moving Average (MA) Processes	328
12.7.1 MA(1) Processes	328
12.7.2 General MA Processes	330
12.8 ARMA Processes	331
12.8.1 The Backwards Operator	331

12.8.2	The ARMA Model	332
12.8.3	ARMA(1,1) Processes	332
12.8.4	Estimation of ARMA Parameters	333
12.8.5	The Differencing Operator	333
12.9	ARIMA Processes	334
12.9.1	Drifts in ARIMA Processes	337
12.10	Unit Root Tests	338
12.10.1	How Do Unit Root Tests Work?	341
12.11	Automatic Selection of an ARIMA Model	342
12.12	Forecasting	342
12.12.1	Forecast Errors and Prediction Intervals	344
12.12.2	Computing Forecast Limits by Simulation	346
12.13	Partial Autocorrelation Coefficients	349
12.14	Bibliographic Notes	352
12.15	R Lab	352
12.15.1	T-bill Rates	352
12.15.2	Forecasting	355
12.16	Exercises	356
	References	360
13	Time Series Models: Further Topics	361
13.1	Seasonal ARIMA Models	361
13.1.1	Seasonal and Nonseasonal Differencing	362
13.1.2	Multiplicative ARIMA Models	362
13.2	Box–Cox Transformation for Time Series	365
13.3	Time Series and Regression	367
13.3.1	Residual Correlation and Spurious Regressions	368
13.3.2	Heteroscedasticity and Autocorrelation Consistent (HAC) Standard Errors	373
13.3.3	Linear Regression with ARMA Errors	377
13.4	Multivariate Time Series	380
13.4.1	The Cross-Correlation Function	380
13.4.2	Multivariate White Noise	382
13.4.3	Multivariate ACF Plots and the Multivariate Ljung–Box Test	383
13.4.4	Multivariate ARMA Processes	384
13.4.5	Prediction Using Multivariate AR Models	387
13.5	Long-Memory Processes	389
13.5.1	The Need for Long-Memory Stationary Models	389
13.5.2	Fractional Differencing	390
13.5.3	FARIMA Processes	391
13.6	Bootstrapping Time Series	394
13.7	Bibliographic Notes	395
13.8	R Lab	395
13.8.1	Seasonal ARIMA Models	395
13.8.2	Regression with HAC Standard Errors	396

13.8.3	Regression with ARMA Noise	397
13.8.4	VAR Models	397
13.8.5	Long-Memory Processes	399
13.8.6	Model-Based Bootstrapping of an ARIMA Process	400
13.9	Exercises	401
	References	403
14	GARCH Models	405
14.1	Introduction	405
14.2	Estimating Conditional Means and Variances	406
14.3	ARCH(1) Processes	407
14.4	The AR(1)+ARCH(1) Model	409
14.5	ARCH(p) Models	411
14.6	ARIMA(p_M, d, q_M)+GARCH(p_V, q_V) Models	411
14.6.1	Residuals for ARIMA(p_M, d, q_M)+GARCH(p_V, q_V) Models	412
14.7	GARCH Processes Have Heavy Tails	413
14.8	Fitting ARMA+GARCH Models	413
14.9	GARCH Models as ARMA Models	418
14.10	GARCH(1,1) Processes	419
14.11	APARCH Models	421
14.12	Linear Regression with ARMA+GARCH Errors	424
14.13	Forecasting ARMA+GARCH Processes	426
14.14	Multivariate GARCH Processes	428
14.14.1	Multivariate Conditional Heteroscedasticity	428
14.14.2	Basic Setting	431
14.14.3	Exponentially Weighted Moving Average (EWMA) Model	432
14.14.4	Orthogonal GARCH Models	433
14.14.5	Dynamic Orthogonal Component (DOC) Models	436
14.14.6	Dynamic Conditional Correlation (DCC) Models	439
14.14.7	Model Checking	441
14.15	Bibliographic Notes	443
14.16	R Lab	443
14.16.1	Fitting GARCH Models	443
14.16.2	The GARCH-in-Mean (GARCH-M) Model	445
14.16.3	Fitting Multivariate GARCH Models	445
14.17	Exercises	447
	References	451
15	Cointegration	453
15.1	Introduction	453
15.2	Vector Error Correction Models	455
15.3	Trading Strategies	459
15.4	Bibliographic Notes	460

15.5	R Lab	460
15.5.1	Cointegration Analysis of Midcap Prices	460
15.5.2	Cointegration Analysis of Yields	460
15.5.3	Cointegration Analysis of Daily Stock Prices	461
15.5.4	Simulation	462
15.6	Exercises	462
	References	463
16	Portfolio Selection	465
16.1	Trading Off Expected Return and Risk	465
16.2	One Risky Asset and One Risk-Free Asset	465
16.2.1	Estimating $E(R)$ and σ_R	467
16.3	Two Risky Assets	468
16.3.1	Risk Versus Expected Return	468
16.4	Combining Two Risky Assets with a Risk-Free Asset	469
16.4.1	Tangency Portfolio with Two Risky Assets	469
16.4.2	Combining the Tangency Portfolio with the Risk-Free Asset	471
16.4.3	Effect of ρ_{12}	472
16.5	Selling Short	473
16.6	Risk-Efficient Portfolios with N Risky Assets	474
16.7	Resampling and Efficient Portfolios	479
16.8	Utility	484
16.9	Bibliographic Notes	488
16.10	R Lab	488
16.10.1	Efficient Equity Portfolios	488
16.10.2	Efficient Portfolios with Apple, Exxon-Mobil, Target, and McDonald's Stock	489
16.10.3	Finding the Set of Possible Expected Returns	490
16.11	Exercises	491
	References	493
17	The Capital Asset Pricing Model	495
17.1	Introduction to the CAPM	495
17.2	The Capital Market Line (CML)	496
17.3	Betas and the Security Market Line	499
17.3.1	Examples of Betas	500
17.3.2	Comparison of the CML with the SML	500
17.4	The Security Characteristic Line	501
17.4.1	Reducing Unique Risk by Diversification	503
17.4.2	Are the Assumptions Sensible?	504
17.5	Some More Portfolio Theory	504
17.5.1	Contributions to the Market Portfolio's Risk	505
17.5.2	Derivation of the SML	505
17.6	Estimation of Beta and Testing the CAPM	507

17.6.1	Estimation Using Regression	507
17.6.2	Testing the CAPM	509
17.6.3	Interpretation of Alpha	509
17.7	Using the CAPM in Portfolio Analysis	510
17.8	Bibliographic Notes	510
17.9	R Lab	510
17.9.1	Zero-beta Portfolios	512
17.10	Exercises	512
	References	515
18	Factor Models and Principal Components	517
18.1	Dimension Reduction	517
18.2	Principal Components Analysis	517
18.3	Factor Models	527
18.4	Fitting Factor Models by Time Series Regression	528
18.4.1	Fama and French Three-Factor Model	529
18.4.2	Estimating Expectations and Covariances of Asset Returns	534
18.5	Cross-Sectional Factor Models	538
18.6	Statistical Factor Models	540
18.6.1	Varimax Rotation of the Factors	545
18.7	Bibliographic Notes	546
18.8	R Lab	546
18.8.1	PCA	546
18.8.2	Fitting Factor Models by Time Series Regression	548
18.8.3	Statistical Factor Models	550
18.9	Exercises	551
	References	552
19	Risk Management	553
19.1	The Need for Risk Management	553
19.2	Estimating VaR and ES with One Asset	555
19.2.1	Nonparametric Estimation of VaR and ES	555
19.2.2	Parametric Estimation of VaR and ES	557
19.3	Bootstrap Confidence Intervals for VaR and ES	559
19.4	Estimating VaR and ES Using ARMA+GARCH Models	561
19.5	Estimating VaR and ES for a Portfolio of Assets	563
19.6	Estimation of VaR Assuming Polynomial Tails	565
19.6.1	Estimating the Tail Index	567
19.7	Pareto Distributions	571
19.8	Choosing the Horizon and Confidence Level	571
19.9	VaR and Diversification	573
19.10	Bibliographic Notes	575
19.11	R Lab	575
19.11.1	Univariate VaR and ES	575
19.11.2	VaR Using a Multivariate- <i>t</i> Model	576

19.12 Exercises	577
References	578
20 Bayesian Data Analysis and MCMC	581
20.1 Introduction	581
20.2 Bayes's Theorem	582
20.3 Prior and Posterior Distributions	584
20.4 Conjugate Priors	586
20.5 Central Limit Theorem for the Posterior	592
20.6 Posterior Intervals	593
20.7 Markov Chain Monte Carlo	595
20.7.1 Gibbs Sampling	596
20.7.2 Other Markov Chain Monte Carlo Samplers	597
20.7.3 Analysis of MCMC Output	597
20.7.4 JAGS	598
20.7.5 Monitoring MCMC Convergence and Mixing	602
20.7.6 DIC and p_D for Model Comparisons	609
20.8 Hierarchical Priors	612
20.9 Bayesian Estimation of a Covariance Matrix	618
20.9.1 Estimating a Multivariate Gaussian Covariance Matrix	618
20.9.2 Estimating a Multivariate- t Scale Matrix	620
20.9.3 Non-Wishart Priors for the Covariate Matrix	623
20.10 Stochastic Volatility Models	623
20.11 Fitting GARCH Models with MCMC	626
20.12 Fitting a Factor Model	629
20.13 Sampling a Stationary Process	632
20.14 Bibliographic Notes	635
20.15 R Lab	636
20.15.1 Fitting a t -Distribution by MCMC	636
20.15.2 AR Models	639
20.15.3 MA Models	640
20.15.4 ARMA Models	641
20.16 Exercises	642
References	643
21 Nonparametric Regression and Splines	645
21.1 Introduction	645
21.2 Local Polynomial Regression	648
21.2.1 Lowess and Loess	652
21.3 Linear Smoothers	653
21.3.1 The Smoother Matrix and the Effective Degrees of Freedom	653
21.3.2 AIC, CV, and GCV	654
21.4 Polynomial Splines	654
21.4.1 Linear Splines with One Knot	655

21.4.2	Linear Splines with Many Knots	656
21.4.3	Quadratic Splines	656
21.4.4	p th Degree Splines	657
21.4.5	Other Spline Bases	658
21.5	Penalized Splines	658
21.5.1	Cubic Smoothing Splines	659
21.5.2	Selecting the Amount of Penalization	659
21.6	Bibliographic Notes	664
21.7	R Lab	664
21.7.1	Additive Model for Wages, Education, and Experience	664
21.7.2	An Extended CKLS Model for the Short Rate	665
21.8	Exercises	666
	References	667
A	Facts from Probability, Statistics, and Algebra	669
A.1	Introduction	669
A.2	Probability Distributions	669
A.2.1	Cumulative Distribution Functions	669
A.2.2	Quantiles and Percentiles	670
A.2.3	Symmetry and Modes	670
A.2.4	Support of a Distribution	670
A.3	When Do Expected Values and Variances Exist?	671
A.4	Monotonic Functions	672
A.5	The Minimum, Maximum, Infimum, and Supremum of a Set	672
A.6	Functions of Random Variables	672
A.7	Random Samples	673
A.8	The Binomial Distribution	674
A.9	Some Common Continuous Distributions	674
A.9.1	Uniform Distributions	674
A.9.2	Transformation by the CDF and Inverse CDF	675
A.9.3	Normal Distributions	676
A.9.4	The Lognormal Distribution	676
A.9.5	Exponential and Double-Exponential Distributions	678
A.9.6	Gamma and Inverse-Gamma Distributions	678
A.9.7	Beta Distributions	679
A.9.8	Pareto Distributions	680
A.10	Sampling a Normal Distribution	681
A.10.1	Chi-Squared Distributions	681
A.10.2	F -Distributions	681
A.11	Law of Large Numbers and the Central Limit Theorem for the Sample Mean	682
A.12	Bivariate Distributions	682

A.13	Correlation and Covariance	683
A.13.1	Normal Distributions: Conditional Expectations and Variance	687
A.14	Multivariate Distributions	687
A.14.1	Conditional Densities	688
A.15	Stochastic Processes	688
A.16	Estimation	689
A.16.1	Introduction	689
A.16.2	Standard Errors	689
A.17	Confidence Intervals	690
A.17.1	Confidence Interval for the Mean	690
A.17.2	Confidence Intervals for the Variance and Standard Deviation	692
A.17.3	Confidence Intervals Based on Standard Errors	693
A.18	Hypothesis Testing	693
A.18.1	Hypotheses, Types of Errors, and Rejection Regions ..	693
A.18.2	p -Values	693
A.18.3	Two-Sample t -Tests	694
A.18.4	Statistical Versus Practical Significance	697
A.19	Prediction	697
A.20	Facts About Vectors and Matrices	698
A.21	Roots of Polynomials and Complex Numbers	699
A.22	Bibliographic Notes	700
	References	700
Index		703

Notation

The following conventions are observed as much as possible:

- Lowercase letters, e.g., a and b , are used for nonrandom scalars.
- Lowercase boldface letters, e.g., \mathbf{a} , \mathbf{b} , and $\boldsymbol{\theta}$, are used for nonrandom vectors.
- Uppercase letters, e.g., X and Y , are used for random variables.
- Uppercase bold letters either early in the Roman alphabet or in Greek without a “hat,” e.g., \mathbf{A} , \mathbf{B} , and $\boldsymbol{\Omega}$, are used for nonrandom matrices.
- A hat over a parameter or parameter vector, e.g., $\hat{\theta}$ and $\hat{\boldsymbol{\theta}}$, denotes an estimator of the corresponding parameter or parameter vector.
- \mathbf{I} denotes the identity matrix with dimension appropriate for the context.
- $\text{diag}(d_1, \dots, d_p)$ is a diagonal matrix with diagonal elements d_1, \dots, d_p .
- Greek letters with a “hat” or uppercase bold letters later in the Roman alphabet, e.g., \mathbf{X} , \mathbf{Y} , and $\hat{\boldsymbol{\theta}}$, will be used for random vectors.
- $\log(x)$ is the natural logarithm of x and $\log_{10}(x)$ is the base-10 logarithm.
- $E(X)$ is the expected value of a random variable X .
- $\text{Var}(X)$ and σ_X^2 are used to denote the variance of a random variable X .
- $\text{Cov}(X, Y)$ and σ_{XY} are used to denote the covariance between the random variables X and Y .
- $\text{Corr}(X, Y)$ and ρ_{XY} are used to denote the correlation between the random variables X and Y .
- $\text{COV}(\mathbf{X})$ is the covariance matrix of a random vector \mathbf{X} .
- $\text{CORR}(\mathbf{X})$ is the correlation matrix of a random vector \mathbf{X} .
- A Greek letter denotes a parameter, e.g., θ .
- A boldface Greek letter, e.g., $\boldsymbol{\theta}$, denotes a vector of parameters.
- \mathbb{R} is the set of real numbers and \mathbb{R}^p is the p -dimensional Euclidean space, the set of all real p -dimensional vectors.
- $A \cap B$ and $A \cup B$ are, respectively, the intersection and union of the sets A and B .
- \emptyset is the empty set.

- If A is some statement, then $I\{A\}$ is called the indicator function of A and is equal to 1 if A is true and equal to 0 if A is false.
- If f_1 and f_2 are two functions of a variable x , then

$$f_1(x) \sim f_2(x) \text{ as } x \rightarrow x_0$$

means that

$$\lim_{x \rightarrow x_0} \frac{f_1(x)}{f_2(x)} = 1.$$

Similarly,

$$a_n \sim b_n$$

means that the sequences $\{a_n\}$ and $\{b_n\}$ are such that

$$\frac{a_n}{b_n} \rightarrow 1 \text{ as } n \rightarrow \infty.$$

- Vectors are column vectors and transposed vectors are rows, e.g.,

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

and

$$\mathbf{x}^T = (x_1 \quad \cdots \quad x_n).$$

- $|\mathbf{A}|$ is the determinant of a square matrix \mathbf{A} .
- $\text{tr}(\mathbf{A})$ is the trace (sum of the diagonal elements) of a square matrix \mathbf{A} .
- $f(x) \propto g(x)$ means that $f(x)$ is proportional to $g(x)$, that is, $f(x) = ag(x)$ for some nonzero constant a .
- A word appearing in italic font is being defined or introduced in the text.

Introduction

This book is about the analysis of financial markets data. After this brief introductory chapter, we turn immediately in Chaps. 2 and 3 to the sources of the data, returns on equities and prices and yields on bonds. Chapter 4 develops methods for informal, often graphical, analysis of data. More formal methods based on statistical inference, that is, estimation and testing, are introduced in Chap. 5. The chapters that follow Chap. 5 cover a variety of more advanced statistical techniques: ARIMA models, regression, multivariate models, copulas, GARCH models, factor models, cointegration, Bayesian statistics, and nonparametric regression.

Much of finance is concerned with financial risk. The *return* on an investment is its revenue expressed as a fraction of the initial investment. If one invests at time t_1 in an asset with price P_{t_1} and the price later at time t_2 is P_{t_2} , then the net return for the holding period from t_1 to t_2 is $(P_{t_2} - P_{t_1})/P_{t_1}$. For most assets, future returns cannot be known exactly and therefore are random variables. *Risk* means uncertainty in future returns from an investment, in particular, that the investment could earn less than the expected return and even result in a loss, that is, a negative return. Risk is often measured by the standard deviation of the return, which we also call the volatility. Recently there has been a trend toward measuring risk by value-at-risk (VaR) and expected shortfall (ES). These focus on large losses and are more direct indications of financial risk than the standard deviation of the return. Because risk depends upon the probability distribution of a return, probability and statistics are fundamental tools for finance. Probability is needed for risk calculations, and statistics is needed to estimate parameters such as the standard deviation of a return or to test hypotheses such as the so-called random walk hypothesis which states that future returns are independent of the past.

In financial engineering there are two kinds of probability distributions that can be estimated. Objective probabilities are the true probabilities of events. Risk-neutral or pricing probabilities give model outputs that agree with market prices and reflect the market's beliefs about the probabilities of future events. The statistical techniques in this book can be used to estimate both types of probabilities. Objective probabilities are usually estimated from historical data, whereas risk-neutral probabilities are estimated from the prices of options and other financial instruments.

Finance makes extensive use of probability models, for example, those used to derive the famous Black–Scholes formula. Use of these models raises important questions of a statistical nature such as: Are these models supported by financial markets data? How are the parameters in these models estimated? Can the models be simplified or, conversely, should they be elaborated?

After Chaps. 4–8 develop a foundation in probability, statistics, and exploratory data analysis, Chaps. 12 and 13 look at ARIMA models for time series. Time series are sequences of data sampled over time, so much of the data from financial markets are time series. ARIMA models are stochastic processes, that is, probability models for sequences of random variables. In Chap. 16 we study optimal portfolios of risky assets (e.g., stocks) and of risky assets and risk-free assets (e.g., short-term U.S. Treasury bills). Chapters 9–11 cover one of the most important areas of applied statistics, regression. Chapter 15 introduces cointegration analysis. In Chap. 17 portfolio theory and regression are applied to the CAPM. Chapter 18 introduces factor models, which generalize the CAPM. Chapters 14–21 cover other areas of statistics and finance such as GARCH models of nonconstant volatility, Bayesian statistics, risk management, and nonparametric regression.

Several related themes will be emphasized in this book:

Always look at the data According to a famous philosopher and baseball player, Yogi Berra, “You can see a lot by just looking.” This is certainly true in statistics. The first step in data analysis should be plotting the data in several ways. Graphical analysis is emphasized in Chap. 4 and used throughout the book. Problems such as bad data, outliers, mislabeling of variables, missing data, and an unsuitable model can often be detected by visual inspection. *Bad data* refers to data that are outlying because of errors, e.g., recording errors. Bad data should be corrected when possible and otherwise deleted. Outliers due, for example, to a stock market crash are “good data” and should be retained, though the model may need to be expanded to accommodate them. It is important to detect both bad data and outliers, and to understand which is which, so that appropriate action can be taken.

All models are false Many statisticians are familiar with the observation of George Box that “all models are false but some models are useful.” This fact should be kept in mind whenever one wonders whether a statistical,

economic, or financial model is “true.” Only computer-simulated data have a “true model.” No model can be as complex as the real world, and even if such a model did exist, it would be too complex to be useful.

Bias-variance tradeoff If useful models exist, how do we find them? The answer to this question depends ultimately on the intended uses of the model. One very useful principle is *parsimony* of parameters, which means that we should use only as many parameters as necessary. Complex models with unnecessary parameters increase estimation error and make interpretation of the model more difficult. However, a model that is too simple will not capture important features of the data and will lead to serious biases. Simple models have large biases but small variances of the estimators. Complex models have small biases but large variances. Therefore, model choice involves finding a good tradeoff between bias and variance.

Uncertainty analysis It is essential that the uncertainty due to estimation and modeling errors be quantified. For example, portfolio optimization methods that assume that return means, variances, and correlations are known exactly are suboptimal when these parameters are only estimated (as is always the case). Taking uncertainty into account leads to other techniques for portfolio selection—see Chap. 16. With complex models, uncertainty analysis could be challenging in the past, but no longer is so because of modern statistical techniques such as resampling (Chap. 6) and Bayesian MCMC (Chap. 20).

Financial markets data are not normally distributed Introductory statistics textbooks model continuously distributed data with the normal distribution. This is fine in many domains of application where data are well approximated by a normal distribution. However, in finance, stock returns, changes in interest rates, changes in foreign exchange rates, and other data of interest have many more outliers than would occur under normality. For modeling financial markets data, heavy-tailed distributions such as the t -distributions are much more suitable than normal distributions—see Chap. 5. *Remember:* In finance, the normal distribution is not normal.

Variances are not constant Introductory textbooks also assume constant variability. This is another assumption that is rarely true for financial markets data. For example, the daily return on the market on Black Monday, October 19, 1987, was -23% , that is, the market lost 23% of its value in a single day! A return of this magnitude is virtually impossible under a normal model with a constant variance, and it is still quite unlikely under a t -distribution with constant variance, but much more likely under a t -distribution model with conditional heteroskedasticity, e.g., a GARCH model (Chap. 14).

1.1 Bibliographic Notes

The dictum that “All models are false but some models are useful” is from Box (1976).

References

- Box, G. E. P. (1976) Science and statistics, *Journal of the American Statistical Association*, 71, 791–799.

Returns

2.1 Introduction

The goal of investing is, of course, to make a profit. The revenue from investing, or the loss in the case of negative revenue, depends upon both the change in prices and the amounts of the assets being held. Investors are interested in revenues that are high relative to the size of the initial investments. Returns measure this, because returns on an asset, e.g., a stock, a bond, a portfolio of stocks and bonds, are changes in price expressed as a fraction of the initial price.

2.1.1 Net Returns

Let P_t be the price of an asset at time t . Assuming no dividends, the *net return* over the holding period from time $t - 1$ to time t is

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

The numerator $P_t - P_{t-1}$ is the revenue or profit during the holding period, with a negative profit meaning a loss. The denominator, P_{t-1} , was the initial investment at the start of the holding period. Therefore, the net return can be viewed as the relative revenue or profit rate.

The revenue from holding an asset is

$$\text{revenue} = \text{initial investment} \times \text{net return}.$$

For example, an initial investment of \$10,000 and a net return of 6 % earns a revenue of \$600. Because $P_t \geq 0$,

$$R_t \geq -1, \quad (2.1)$$

so the worst possible return is -1 , that is, a 100 % loss, and occurs if the asset becomes worthless.

2.1.2 Gross Returns

The simple *gross return* is

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

For example, if $P_t = 2$ and $P_{t+1} = 2.1$, then $1 + R_{t+1} = 1.05$, or 105 %, and $R_{t+1} = 0.05$, or 5 %. One's final wealth at time t is one's initial wealth at time $t-1$ times the gross return. Stated differently, if X_0 is the initial at time $t-1$, then $X_0(1 + R_t)$ is one's wealth at time t .

Returns are scale-free, meaning that they do not depend on units (dollars, cents, etc.). Returns are *not* unitless. Their unit is time; they depend on the units of t (hour, day, etc.). In this example, if t is measured in years, then, stated more precisely, the net return is 5 % per year.

The *gross return over the most recent k periods* is the product of the k single-period gross returns (from time $t-k$ to time t):

$$\begin{aligned} 1 + R_t(k) &= \frac{P_t}{P_{t-k}} = \left(\frac{P_t}{P_{t-1}} \right) \left(\frac{P_{t-1}}{P_{t-2}} \right) \cdots \left(\frac{P_{t-k+1}}{P_{t-k}} \right) \\ &= (1 + R_t) \cdots (1 + R_{t-k+1}). \end{aligned}$$

The k -period net return is $R_t(k)$.

2.1.3 Log Returns

Log returns, also called *continuously compounded returns*, are denoted by r_t and defined as

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

where $p_t = \log(P_t)$ is called the *log price*.

Log returns are approximately equal to returns because if x is small, then $\log(1+x) \approx x$, as can be seen in Fig. 2.1, where $\log(1+x)$ is plotted. Notice in that figure that $\log(1+x)$ is very close to x if $|x| < 0.1$, e.g., for returns that are less than 10 %.

For example, a 5 % return equals a 4.88 % log return since $\log(1 + 0.05) = 0.0488$. Also, a -5 % return equals a -5.13 % log return since $\log(1 - 0.05) = -0.0513$. In both cases, $r_t = \log(1 + R_t) \approx R_t$. Also, $\log(1 + 0.01) = 0.00995$ and $\log(1 - 0.01) = -0.01005$, so log returns of ± 1 % are very close to the

corresponding net returns. Since returns are smaller in magnitude over shorter periods, we can expect returns and log returns to be similar for daily returns, less similar for yearly returns, and not necessarily similar for longer periods such as 10 years.

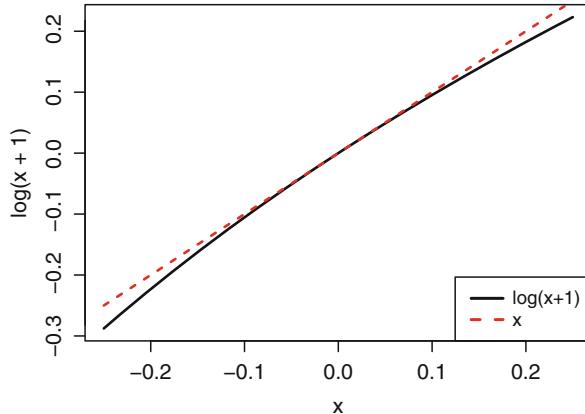


Fig. 2.1. Comparison of functions $\log(1 + x)$ and x .

The return and log return have the same sign. The magnitude of the log return is smaller (larger) than that of the return if they are both positive (negative). The difference between a return and a log return is most pronounced when both are very negative. Returns close to the lower bound of -1 , that is complete losses, correspond to log return close to $-\infty$.

One advantage of using log returns is simplicity of multiperiod returns. A k -period log return is simply the sum of the single-period log returns, rather than the product as for gross returns. To see this, note that the k -period log return is

$$\begin{aligned} r_t(k) &= \log\{1 + R_t(k)\} \\ &= \log\{(1 + R_t) \cdots (1 + R_{t-k+1})\} \\ &= \log(1 + R_t) + \cdots + \log(1 + R_{t-k+1}) \\ &= r_t + r_{t-1} + \cdots + r_{t-k+1}. \end{aligned}$$

2.1.4 Adjustment for Dividends

Many stocks, especially those of mature companies, pay dividends that must be accounted for when computing returns. Similarly, bonds pay interest. If a

dividend (or interest) D_t is paid prior to time t , then the gross return at time t is defined as

$$1 + R_t = \frac{P_t + D_t}{P_{t-1}}, \quad (2.2)$$

and so the net return is $R_t = (P_t + D_t)/P_{t-1} - 1$ and the log return is $r_t = \log(1 + R_t) = \log(P_t + D_t) - \log(P_{t-1})$. Multiple-period gross returns are products of single-period gross returns so that

$$\begin{aligned} 1 + R_t(k) &= \left(\frac{P_t + D_t}{P_{t-1}} \right) \left(\frac{P_{t-1} + D_{t-1}}{P_{t-2}} \right) \cdots \left(\frac{P_{t-k+1} + D_{t-k+1}}{P_{t-k}} \right) \\ &= (1 + R_t)(1 + R_{t-1}) \cdots (1 + R_{t-k+1}), \end{aligned} \quad (2.3)$$

where, for any time s , $D_s = 0$ if there is no dividend between $s - 1$ and s . Similarly, a k -period log return is

$$\begin{aligned} r_t(k) &= \log\{1 + R_t(k)\} = \log(1 + R_t) + \cdots + \log(1 + R_{t-k+1}) \\ &= \log\left(\frac{P_t + D_t}{P_{t-1}}\right) + \cdots + \log\left(\frac{P_{t-k+1} + D_{t-k+1}}{P_{t-k}}\right). \end{aligned}$$

2.2 The Random Walk Model

The *random walk hypothesis* states that the single-period log returns, $r_t = \log(1 + R_t)$, are independent. Because

$$\begin{aligned} 1 + R_t(k) &= (1 + R_t) \cdots (1 + R_{t-k+1}) \\ &= \exp(r_t) \cdots \exp(r_{t-k+1}) \\ &= \exp(r_t + \cdots + r_{t-k+1}), \end{aligned}$$

we have

$$\log\{1 + R_t(k)\} = r_t + \cdots + r_{t-k+1}. \quad (2.4)$$

It is sometimes assumed further that the log returns are $N(\mu, \sigma^2)$ for some constant mean and variance. Since sums of normal random variables are themselves normal, normality of single-period log returns implies normality of multiple-period log returns. Under these assumptions, $\log\{1 + R_t(k)\}$ is $N(k\mu, k\sigma^2)$.

2.2.1 Random Walks

Model (2.4) is an example of a random walk model. Let Z_1, Z_2, \dots be i.i.d. (independent and identically distributed) with mean μ and standard deviation σ . Let S_0 be an arbitrary starting point and

$$S_t = S_0 + Z_1 + \cdots + Z_t, \quad t \geq 1. \quad (2.5)$$

From (2.5), S_t is the position of the random walker after t steps starting at S_0 .

The process S_0, S_1, \dots is called a *random walk* and Z_1, Z_2, \dots are its steps. If the steps are normally distributed, then the process is called a *normal random walk*. The expectation and variance of S_t , conditional given S_0 , are $E(S_t|S_0) = S_0 + \mu t$ and $\text{Var}(S_t|S_0) = \sigma^2 t$. The parameter μ is called the *drift* and determines the general direction of the random walk. The parameter σ is the *volatility* and determines how much the random walk fluctuates about the conditional mean $S_0 + \mu t$. Since the standard deviation of S_t given S_0 is $\sigma\sqrt{t}$, $(S_0 + \mu t) \pm \sigma\sqrt{t}$ gives the mean plus and minus one standard deviation, which, for a normal random walk, gives a range containing 68% probability. The width of this range grows proportionally to \sqrt{t} , as is illustrated in Fig. 2.2, showing that at time $t = 0$ we know far less about where the random walk will be in the distant future compared to where it will be in the immediate future.

2.2.2 Geometric Random Walks

Recall that $\log\{1 + R_t(k)\} = r_t + \dots + r_{t-k+1}$. Therefore,

$$\frac{P_t}{P_{t-k}} = 1 + R_t(k) = \exp(r_t + \dots + r_{t-k+1}), \quad (2.6)$$

so taking $k = t$, we have

$$P_t = P_0 \exp(r_t + r_{t-1} + \dots + r_1). \quad (2.7)$$

We call such a process whose logarithm is a random walk a *geometric random walk* or an *exponential random walk*. If r_1, r_2, \dots are i.i.d. $N(\mu, \sigma^2)$, then P_t is lognormal for all t and the process is called a *lognormal geometric random walk with parameters (μ, σ^2)* . As discussed in Appendix A.9.4, μ is called the log-mean and σ is called the log-standard deviation of the log-normal distribution of $\exp(r_t)$. Also, μ is sometimes called the log-drift of the lognormal geometric random walk.

2.2.3 Are Log Prices a Lognormal Geometric Random Walk?

Much work in mathematical finance assumes that prices follow a lognormal geometric random walk or its continuous-time analog, geometric Brownian motion. So a natural question is whether this assumption is usually true. The quick answer is “no.” The lognormal geometric random walk makes two assumptions: (1) the log returns are normally distributed and (2) the log returns are mutually independent.

In Chaps. 4 and 5, we will investigate the marginal distributions of several series of log returns. The conclusion will be that, though the return density has a bell shape somewhat like that of normal densities, the tails of the log return distributions are generally much heavier than normal tails. Typically, a

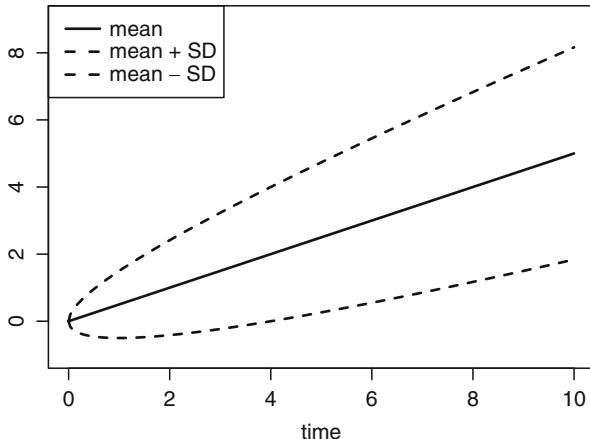


Fig. 2.2. Mean and bounds (mean plus and minus one standard deviation) on a random walk with $S_0 = 0$, $\mu = 0.5$, and $\sigma = 1$. At any given time, the probability of being between the bounds (dashed curves) is 68 % if the distribution of the steps is normal. Since $\mu > 0$, there is an overall positive trend that would be reversed if μ were negative.

t -distribution with a small degrees-of-freedom parameter, say 4–6, is a much better fit than the normal model. However, the log-return distributions do appear to be symmetric, or at least nearly so.

The independence assumption is also violated. First, there is some correlation between returns. The correlations, however, are generally small. More seriously, returns exhibit *volatility clustering*, which means that if we see high volatility in current returns then we can expect this higher volatility to continue, at least for a while. Volatility clustering can be detected by checking for correlations between the *squared* returns.

Before discarding the assumption that the prices of an asset are a lognormal geometric random walk, it is worth remembering Box's dictum that “all models are false, but some models are useful.” This assumption is sometimes useful, e.g., for deriving the famous Black–Scholes formula.

2.3 Bibliographic Notes

The random walk hypothesis is related to the so-called efficient market hypothesis; see Ruppert et al. (2003) for discussion and further references. Bodie et al. (1999) and Sharpe et al. (1995) are good introductions to the random walk hypothesis and market efficiency. A more advanced discussion of the random walk hypothesis is found in Chap. 2 of Campbell et al. (1997) and Lo and MacKinlay (1999). Much empirical evidence about the behavior of

returns is reviewed by Fama (1965, 1970, 1991, 1998). Evidence against the efficient market hypothesis can be found in the field of behavioral finance which uses the study of human behavior to understand market behavior; see Shefrin (2000), Shleifer (2000), and Thaler (1993). One indication of market inefficiency is excess volatility of market prices; see Shiller (1992) or Shiller (2000) for a less technical discussion.

R will be used extensively in what follows. Dalgaard (2008) and Zuur et al. (2009) are good places to start learning R.

2.4 R Lab

2.4.1 Data Analysis

Obtain the data set `Stock_bond.csv` from the book's website and put it in your working directory. Start R¹ and you should see a console window open up. Use `Change Dir` in the “File” menu to change to the working directory. Read the data with the following command:

```
dat = read.csv("Stock_bond.csv", header = TRUE)
```

The data set `Stock_bond.csv` contains daily volumes and adjusted closing (AC) prices of stocks and the S&P 500 (columns B–W) and yields on bonds (columns X–AD) from 2-Jan-1987 to 1-Sep-2006.

This book does not give detailed information about R functions since this information is readily available elsewhere. For example, you can use R's help to obtain more information about the `read.csv()` function by typing “?read.csv” in your R console and then hitting the Enter key. You should also use the manual *An Introduction to R* that is available on R's help file and also on CRAN. Another resource for those starting to learn R is Zuur et al. (2009).

An alternative to typing commands in the console is to start a new script from the “file” menu, put code into the editor, highlight the lines, and then press Ctrl-R to run the code that has been highlighted.² This technique is useful for debugging. You can save the script file and then reuse or modify it.

Once a file is saved, the entire file can be run by “sourcing” it. You can use the “file” menu in R to source a file or use the `source()` function. If the file is in the editor, then it can be run by hitting Ctrl-A to highlight the entire file and then Ctrl-R.

The next lines of code print the names of the variables in the data set, attach the data, and plot the adjusted closing prices of GM and Ford.

¹ You can also run R from Rstudio and, in fact, Rstudio is highly recommended. The authors switched from R to Rstudio while the second edition of this book was being written.

² Or click the “run” button in Rstudio.

```

1 names(dat)
2 attach(dat)
3 par(mfrow = c(1, 2))
4 plot(GM_AC)
5 plot(F_AC)

```

Here and elsewhere in this book, line numbers are often added when listing R code. The line numbers are not part of the code.

By default, as in lines 4 and 5, points are plotted with the character “o”. To plot a line instead, use, for example `plot(GM_AC, type = "l")`. Similarly, `plot(GM_AC, type = "b")` plots both points and a line.

The R function `attach()` puts a database into the R search path. This means that the database is searched by R when evaluating a variable, so objects in the database can be accessed by simply giving their names. If `dat` was not attached, then line 4 would be replaced by `plot(dat$GM_AC)` and similarly for line 5.

The function `par()` specifies plotting parameters and `mfrow=c(n1,n2)` specifies “make a figure, fill by rows, n1 rows and n2 columns.” Thus, the first n1 plots fill the first row and so forth. `mfcol(n1,n2)` fills by columns and so would put the first n2 plots in the first column. As mentioned before, more information about these and other R functions can be obtained from R’s online help or the manual *An Introduction to R*.

Run the code below to find the sample size (`n`), compute GM and Ford returns, and plot GM net returns versus the Ford returns.

```

1 n = dim(dat)[1]
2 GMReturn = GM_AC[-1] / GM_AC[-n] - 1
3 FReturn = F_AC[-1] / F_AC[-n] - 1
4 par(mfrow = c(1, 1))
5 plot(GMReturn,FReturn)

```

On lines 2 and 3, the index `-1` means all indices except the first and similarly `-n` means all indices except the last.

Problem 1 Do the GM and Ford returns seem positively correlated? Do you notice any outlying returns? If “yes,” do outlying GM returns seem to occur with outlying Ford returns?

Problem 2 Compute the log returns for GM and plot the returns versus the log returns. How highly correlated are the two types of returns? (The R function `cor()` computes correlations.)

Problem 3 Repeat Problem 1 with Microsoft (MSFT) and Merck (MRK).

When you exit R, you can “Save workspace image,” which will create an R workspace file in your working directory. Later, you can restart R and load this workspace image into memory by right-clicking on the R workspace file. When R starts, your working directory will be the folder containing the R workspace that was opened. A useful trick when starting a project in a new folder is to put an empty saved workspace into this folder. Double-clicking on the workspace starts R with the folder as the working directory.

2.4.2 Simulations

Hedge funds can earn high profits through the use of leverage, but leverage also creates high risk. The simulations in this section explore the effects of leverage in a simplified setting.

Suppose a hedge fund owns \$1,000,000 of stock and used \$50,000 of its own capital and \$950,000 in borrowed money for the purchase. Suppose that if the value of the stock falls below \$950,000 at the end of any trading day, then the hedge fund will sell all the stock and repay the loan. This will wipe out its \$50,000 investment. The hedge fund is said to be leveraged 20:1 since its position is 20 times the amount of its own capital invested.

Suppose that the daily log returns on the stock have a mean of 0.05/year and a standard deviation of 0.23/year. These can be converted to rates per trading day by dividing by 253 and $\sqrt{253}$, respectively.

Problem 4 *What is the probability that the value of the stock will be below \$950,000 at the close of at least one of the next 45 trading days? To answer this question, run the code below.*

```

1 niter = 1e5           # number of iterations
2 below = rep(0, niter) # set up storage
3 set.seed(2009)
4 for (i in 1:niter)
5 {
6   r = rnorm(45, mean = 0.05/253,
7             sd = 0.23/sqrt(253)) # generate random numbers
8   logPrice = log(1e6) + cumsum(r)
9   minlogP = min(logPrice) # minimum price over next 45 days
10  below[i] = as.numeric(minlogP < log(950000))
11 }
12 mean(below)
```

On line 10, `below[i]` equals 1 if, for the i th simulation, the minimum price over 45 days is less than 950,000. Therefore, on line 12, `mean(below)` is the proportion of simulations where the minimum price is less than 950,000.

If you are unfamiliar with any of the R functions used here, then use R’s help to learn about them; e.g., type `?rnorm` to learn that `rnorm()` generates

normally distributed random numbers. You should study each line of code, understand what it is doing, and convince yourself that the code estimates the probability being requested. Note that anything that follows a pound sign is a comment and is used only to annotate the code.

Suppose the hedge fund will sell the stock for a profit of at least \$100,000 if the value of the stock rises to at least \$1,100,000 at the end of one of the first 100 trading days, sell it for a loss if the value falls below \$950,000 at the end of one of the first 100 trading days, or sell after 100 trading days if the closing price has stayed between \$950,000 and \$1,100,000.

The following questions can be answered by simulations much like the one above. Ignore trading costs and interest when answering these questions.

Problem 5 *What is the probability that the hedge fund will make a profit of at least \$100,000?*

Problem 6 *What is the probability the hedge fund will suffer a loss?*

Problem 7 *What is the expected profit from this trading strategy?*

Problem 8 *What is the expected return? When answering this question, remember that only \$50,000 was invested. Also, the units of return are time, e.g., one can express a return as a daily return or a weekly return. Therefore, one must keep track of how long the hedge fund holds its position before selling.*

2.4.3 Simulating a Geometric Random Walk

In this section you will use simulations to see how stock prices evolve when the log-returns are i.i.d. normal, which implies that the price series is a geometric random walk.

Run the following R code. The `set.seed()` command insures that everyone using this code will have the same random numbers and will obtain the same price series. There are 253 trading days per year, so you are simulating 1 year of daily returns nine times. The price starts at 120.

The code `par(mfrow=c(3,3))` on line 3 opens a graphics window with three rows and three columns and `rnorm()` on line 6 generates normally distributed random numbers.

```

1 set.seed(2012)
2 n = 253
3 par(mfrow=c(3,3))
4 for (i in (1:9))
5 {
6   logr = rnorm(n, 0.05 / 253, 0.2 / sqrt(253))

```

```

7   price = c(120, 120 * exp(cumsum(logr)))
8   plot(price, type = "b")
9 }
```

Problem 9 In this simulation, what are the mean and standard deviation of the log-returns for 1 year?

Problem 10 Discuss how the price series appear to have momentum. Is the appearance of momentum real or an illusion?

Problem 11 Explain what the code `c(120,120*exp(cumsum(logr)))` does.

2.4.4 Let's Look at McDonald's Stock

In this section we will be looking at daily returns on McDonald's stock over the period 2010–2014. To start the lab, run the following commands to get daily adjusted prices over this period:

```

1 data = read.csv('MCD_PriceDaily.csv')
2 head(data)
3 adjPrice = data[, 7]
```

Problem 12 Compute the returns and log returns and plot them against each other. As discussed in Sect. 2.1.3, does it seem reasonable that the two types of daily returns are approximately equal?

Problem 13 Compute the mean and standard deviation for both the returns and the log returns. Comment on the similarities and differences you perceive in the first two moments of each random variable. Does it seem reasonable that they are the same?

Problem 14 Perform a t-test to compare the means of the returns and the log returns. Comment on your findings. Do you reject the null hypothesis that they are the same mean at 5 % significance? Or do you accept it? [Hint: Should you be using an independent samples t-test or a paired-samples t-test?] What are the assumptions behind the t-test? Do you think that they are met in this example? If the assumptions made by the t-test are not met, how would this affect your interpretation of the results of the test?

Problem 15 After looking at return and log return data for McDonald's, are you satisfied that for small values, log returns and returns are interchangeable?

Problem 16 Assume that McDonald's log returns are normally distributed with mean and standard deviation equal to their estimates and that you have been made the following proposition by a friend: If at any point within the next 20 trading days, the price of McDonald's falls below 85 dollars, you will be paid \$100, but if it does not, you have to pay him \$1. The current price of McDonald's is at the end of the sample data, \$93.07. Are you willing to make the bet? (Use 10,000 iterations in your simulation and use the command `set.seed(2015)` to ensure your results are the same as the answer key)

Problem 17 After coming back to your friend with an unwillingness to make the bet, he asks you if you are willing to try a slightly different deal. This time the offer stays the same as before, except he would pay an additional \$25 if the price ever fell below \$84.50. You still only pay him \$1 for losing. Do you now make the bet?

2.5 Exercises

1. Suppose that the daily log returns on a stock are independent and normally distributed with mean 0.001 and standard deviation 0.015. Suppose you buy \$1,000 worth of this stock.
 - (a) What is the probability that after one trading day your investment is worth less than \$990? (**Note:** The R function `pnorm()` will compute a normal CDF, so, for example, `pnorm(0.3, mean = 0.1, sd = 0.2)` is the normal CDF with mean 0.1 and standard deviation 0.2 evaluated at 0.3.)
 - (b) What is the probability that after five trading days your investment is worth less than \$990?
2. The yearly log returns on a stock are normally distributed with mean 0.1 and standard deviation 0.2. The stock is selling at \$100 today. What is the probability that 1 year from now it is selling at \$110 or more?
3. The yearly log returns on a stock are normally distributed with mean 0.08 and standard deviation 0.15. The stock is selling at \$80 today. What is the probability that 2 years from now it is selling at \$90 or more?
4. Suppose the prices of a stock at times 1, 2, and 3 are $P_1 = 95$, $P_2 = 103$, and $P_3 = 98$. Find $r_3(2)$.
5. The prices and dividends of a stock are given in the table below.
 - (a) What is R_2 ?
 - (b) What is $R_4(3)$?
 - (c) What is r_3 ?

t	P_t	D_t
1	52	0.2
2	54	0.2
3	53	0.2
4	59	0.25

6. The prices and dividends of a stock are given in the table below.

- (a) Find $R_3(2)$,
- (b) Find $r_4(3)$.

t	P_t	D_t
1	82	0.1
2	85	0.1
3	83	0.1
4	87	0.125

7. Let r_t be a log return. Suppose that r_1, r_2, \dots are i.i.d. $N(0.06, 0.47)$.

- (a) What is the distribution of $r_t(4) = r_t + r_{t-1} + r_{t-2} + r_{t-3}$?
- (b) What is $P\{r_1(4) < 2\}$?
- (c) What is the covariance between $r_2(1)$ and $r_2(2)$?
- (d) What is the conditional distribution of $r_t(3)$ given $r_{t-2} = 0.6$?
- 8. Suppose that X_1, X_2, \dots is a lognormal geometric random walk with parameters (μ, σ^2) . More specifically, suppose that $X_k = X_0 \exp(r_1 + \dots + r_k)$, where X_0 is a fixed constant and r_1, r_2, \dots are i.i.d. $N(\mu, \sigma^2)$.
 - (a) Find $P(X_2 > 1.3X_0)$.
 - (b) Use (A.4) to find the density of X_1 .
 - (c) Find a formula for the 0.9 quantile of X_k for all k .
 - (d) What is the expected value of X_k^2 for any k ? (Find a formula giving the expected value as a function of k .)
 - (e) Find the variance of X_k for any k .
- 9. Suppose that X_1, X_2, \dots is a lognormal geometric random walk with parameters $\mu = 0.1, \sigma = 0.2$.
 - (a) Find $P(X_3 > 1.2X_0)$.
 - (b) Find the conditional variance of X_k/k given X_0 for any k .
 - (c) Find the minimum number of days before the probability is at least 0.9 of doubling one's money, that is, find the small value of t such that $P(P_t/P_0 \geq 2) \geq 0.9$.
- 10. The daily log returns on a stock are normally distributed with mean 0.0002 and standard deviation 0.03. The stock price is now \$97. What is the probability that it will exceed \$100 after 20 trading days?
- 11. Suppose that daily log-returns are $N(0.0005, 0.012)$. Find the smallest value of t such that $P(P_t/P_0 \geq 2) \geq 0.9$, that is, that after t days the probability the price has doubled is at least 90 %.

References

- Bodie, Z., Kane, A., and Marcus, A. (1999) *Investments*, 4th ed., Irwin/McGraw-Hill, Boston.
- Campbell, J., Lo, A., and MacKinlay, A. (1997) *The Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ.
- Dalgaard, P. (2008) *Introductory Statistics with R*, 2nd ed., Springer.
- Fama, E. (1965) The behavior of stock market prices. *Journal of Business*, **38**, 34–105.
- Fama, E. (1970) Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, **25**, 383–417.
- Fama, E. (1991) Efficient Capital Markets: II. *Journal of Finance*. **46**, 1575–1618.
- Fama, E. (1998) Market efficiency, long-term returns, and behavioral finance. *Journal of Financial Economics*, **49**, 283–306.
- Lo, A. W., and MacKinlay, A. C. (1999) *A Non-Random Walk Down Wall Street*, Princeton University Press, Princeton and Oxford.
- Ruppert, D. (2003) *Statistics and Finance: An Introduction*, Springer, New York.
- Sharpe, W. F., Alexander, G. J., and Bailey, J. V. (1995) *Investments*, 6th ed., Simon and Schuster, Upper Saddle River, NJ.
- Shefrin, H. (2000) *Beyond Greed and Fear: Understanding Behavioral Finance and the Psychology of Investing*, Harvard Business School Press, Boston.
- Shiller, R. (1992) *Market Volatility*, Reprint ed., MIT Press, Cambridge, MA.
- Shiller, R. (2000) *Irrational Exuberance*, Broadway, New York.
- Shleifer, A. (2000) *Inefficient Markets: An Introduction to Behavioral Finance*, Oxford University Press, Oxford.
- Thaler, R. H. (1993) *Advances in Behavioral Finance*, Russell Sage Foundation, New York.
- Zuur, A., Ieno, E., Meesters, E., and Burg, D. (2009) *A Beginner's Guide to R*, Springer, New York.

Fixed Income Securities

3.1 Introduction

Corporations finance their operations by selling stock and bonds. Owning a share of stock means partial ownership of the company. Stockholders share in both the profits and losses of the company. Owning a bond is different. When you buy a bond you are loaning money to the corporation, though bonds, unlike loans, are tradeable. The corporation is obligated to pay back the principal and to pay interest as stipulated by the bond. The bond owner receives a fixed stream of income, unless the corporation defaults on the bond. For this reason, bonds are called “fixed income” securities.

It might appear that bonds are risk-free, almost stodgy, but this is not the case. Many bonds are long-term, e.g., 5, 10, 20, or even 30 years. Even if the corporation stays solvent or if you buy a U.S. Treasury bond, where default is for all intents and purposes impossible, your income from the bond is guaranteed only if you keep the bond to maturity. If you sell the bond before maturity, your return will depend on changes in the price of the bond. Bond prices move in opposite direction to interest rates, so a decrease in interest rates will cause a bond “rally,” where bond prices increase. Long-term bonds are more sensitive to interest-rate changes than short-term bonds. The interest rate on your bond is fixed, but in the market interest rates fluctuate. Therefore, the market value of your bond fluctuates too. For example, if you buy a bond paying 5 % and the rate of interest increases to 6 %, then your bond is inferior to new bonds offering 6 %. Consequently, the price of your bond will decrease. If you sell the bond, you could lose money.

The interest rate of a bond depends on its maturity. For example, on March 28, 2001, the interest rate of Treasury bills¹ was 4.23 % for 3-month bills. The yields on Treasury notes and bonds were 4.41 %, 5.01 %, and 5.46 % for 2-, 10-, and 30-year maturities, respectively. The *term structure* of interest rates describes how rates change with maturity.

3.2 Zero-Coupon Bonds

Zero-coupon bonds, also called *pure discount bonds* and sometimes known as “zeros,” pay no principal or interest until maturity. A “zero” has a *par value* or *face value*, which is the payment made to the bondholder at maturity. The zero sells for less than the par value, which is the reason it is a discount bond.

For example, consider a 20-year zero with a par value of \$1,000 and 6 % interest compounded annually. The market price is the present value of \$1,000 with an annual interest rate of 6 % with annual discounting. That is, the market price is

$$\frac{\$1,000}{(1.06)^{20}} = \$311.80.$$

If the annual interest rate is 6 % but compounded every 6 months, then the price is

$$\frac{\$1,000}{(1.03)^{40}} = \$306.56,$$

and if the annual rate is 6 % compounded continuously, then the price is

$$\frac{\$1,000}{\exp\{(0.06)(20)\}} = \$301.19.$$

3.2.1 Price and Returns Fluctuate with the Interest Rate

For concreteness, assume semiannual compounding. Suppose you bought the zero for \$306.56 and then 6 months later the interest rate increased to 7 %. The market price would now be

$$\frac{\$1,000}{(1.035)^{39}} = \$261.41,$$

so the value of your investment would drop by $(\$306.56 - \$261.41) = \$45.15$. You will still get your \$1,000 if you keep the bond for 20 years, but if you sold it now, you would lose \$45.15. This is a return of

¹ Treasury bills have maturities of 1 year or less, Treasury notes have maturities from 1 to 10 years, and Treasury bonds have maturities from 10 to 30 years.

$$\frac{-45.15}{306.56} = -14.73\%$$

for a half-year, or -29.46% per year. And the interest rate only changed from 6% to 7% ². Notice that the interest rate went up and the bond price went down. This is a general phenomenon. Bond prices always move in the opposite direction of interest rates.

If the interest rate dropped to 5% after 6 months, then your bond would be worth

$$\frac{\$1,000}{(1.025)^{39}} = \$381.74.$$

This would be an annual rate of return of

$$2 \left(\frac{381.74 - 306.56}{306.56} \right) = 49.05\%.$$

If the interest rate remained unchanged at 6% , then the price of the bond would be

$$\frac{\$1,000}{(1.03)^{39}} = \$315.75.$$

The annual rate of return would be

$$2 \left(\frac{315.75 - 306.56}{306.56} \right) = 6\%.$$

Thus, if the interest rate does not change, you can earn a 6% annual rate of return, the same return rate as the interest rate, by selling the bond before maturity. If the interest rate does change, however, the 6% annual rate of return is guaranteed only if you keep the bond until maturity.

General Formula

The price of a zero-coupon bond is given by

$$\text{PRICE} = \text{PAR}(1 + r)^{-T}$$

if T is the time to maturity in years and the annual rate of interest is r with annual compounding. If we assume semiannual compounding, then the price is

$$\text{PRICE} = \text{PAR}(1 + r/2)^{-2T}. \quad (3.1)$$

² Fortunately for investors, a rate change as large as going from 6% to 7% is rare on a 20-year bond.

3.3 Coupon Bonds

Coupon bonds make regular interest payments. Coupon bonds generally sell at or near the par value when issued. At maturity, one receives a principal payment equal to the par value of the bond and the final interest payment.

As an example, consider a 20-year coupon bond with a par value of \$1,000 and 6% annual coupon rate with semiannual coupon payments, so effectively the 6% is compounded semiannually. Each coupon payment will be \$30. Thus, the bondholder receives 40 payments of \$30, one every 6 months plus a principal payment of \$1,000 after 20 years. One can check that the present value of all payments, with discounting at the 6% annual rate (3% semiannual), equals \$1,000:

$$\sum_{t=1}^{40} \frac{30}{(1.03)^t} + \frac{1000}{(1.03)^{40}} = 1000.$$

After 6 months, if the interest rate is unchanged, then the bond (including the first coupon payment, which is now due) is worth

$$\sum_{t=0}^{39} \frac{30}{(1.03)^t} + \frac{1000}{(1.03)^{39}} = (1.03) \left(\sum_{t=1}^{40} \frac{30}{(1.03)^t} + \frac{1000}{(1.03)^{40}} \right) = 1030,$$

which is a semiannually compounded 6% annual return as expected. If the interest rate increases to 7%, then after 6 months the bond (plus the interest due) is only worth

$$\sum_{t=0}^{39} \frac{30}{(1.035)^t} + \frac{1000}{(1.035)^{39}} = (1.035) \left(\sum_{t=1}^{40} \frac{30}{(1.035)^t} + \frac{1000}{(1.035)^{40}} \right) = 924.49.$$

This is an annual return of

$$2 \left(\frac{924.49 - 1000}{1000} \right) = -15.1\%.$$

If the interest rate drops to 5% after 6 months, then the investment is worth

$$\sum_{t=0}^{39} \frac{30}{(1.025)^t} + \frac{1000}{(1.025)^{39}} = (1.025) \left(\sum_{t=1}^{40} \frac{30}{(1.025)^t} + \frac{1000}{(1.025)^{40}} \right) = 1,153.70, \quad (3.2)$$

and the annual return is

$$2 \left(\frac{1153.7 - 1000}{1000} \right) = 30.72\%.$$

3.3.1 A General Formula

Let's derive some useful formulas. If a bond with a par value of PAR matures in T years and makes semiannual coupon payments of C and the yield (rate of interest) is r per half-year, then the value of the bond when it is issued is

$$\begin{aligned} \sum_{t=1}^{2T} \frac{C}{(1+r)^t} + \frac{\text{PAR}}{(1+r)^{2T}} &= \frac{C}{r} \left\{ 1 - (1+r)^{-2T} \right\} + \frac{\text{PAR}}{(1+r)^{2T}} \\ &= \frac{C}{r} + \left\{ \text{PAR} - \frac{C}{r} \right\} (1+r)^{-2T}. \end{aligned} \quad (3.3)$$

Derivation of (3.3)

The summation formula for a finite geometric series is

$$\sum_{i=0}^T r^i = \frac{1 - r^{T+1}}{1 - r}, \quad (3.4)$$

provided that $r \neq 1$. Therefore,

$$\begin{aligned} \sum_{t=1}^{2T} \frac{C}{(1+r)^t} &= \frac{C}{1+r} \sum_{t=0}^{2T-1} \left(\frac{1}{1+r} \right)^t = \frac{C \{ 1 - (1+r)^{-2T} \}}{(1+r) \{ 1 - (1+r)^{-1} \}} \\ &= \frac{C}{r} \{ 1 - (1+r)^{-2T} \}. \end{aligned} \quad (3.5)$$

The remainder of the derivation is straightforward algebra.

3.4 Yield to Maturity

Suppose a bond with $T = 30$ and $C = 40$ is selling for \$1,200, \$200 above par value. If the bond were selling at par value, then the interest rate would be 0.04/half-year ($= 0.08/\text{year}$). The 4 %/half-year rate is called the *coupon rate*.

But the bond is *not* selling at par value. If you purchase the bond at \$1,200, you will make *less* than 8 % per year interest. There are two reasons that the rate of interest is less than 8 %. First, the coupon payments are \$40 or $40/1200 = 3.333\%/\text{half-year}$ (or $6.67\%/\text{year}$) for the \$1,200 investment; 6.67 %/year is called the *current yield*. Second, at maturity you only get back \$1,000, not the entire \$1,200 investment. The current yield of 6.67 %/year, though less than the coupon rate of 8 %/year, overestimates the return since it does not account for this loss of capital.

The *yield to maturity*, often shortened to simply *yield*, is the average rate of return, including the loss (or gain) of capital because the bond was purchased above (or below) par. For this bond, the yield to maturity is the value of r that solves

$$1200 = \frac{40}{r} + \left\{ 1000 - \frac{40}{r} \right\} (1+r)^{-60}. \quad (3.6)$$

The right-hand side of (3.6) is (3.3) with $C = 40$, $T = 30$, and $\text{PAR} = 1000$. It is easy to solve equation (3.6) numerically. The R program in Sect. 3.10.1 does the following:

- computes the bond price for each r value on a grid;
- graphs bond price versus r (this is not necessary, but it is fun to see the graph); and
- interpolates to find the value of r such that the bond value equals \$1,200.

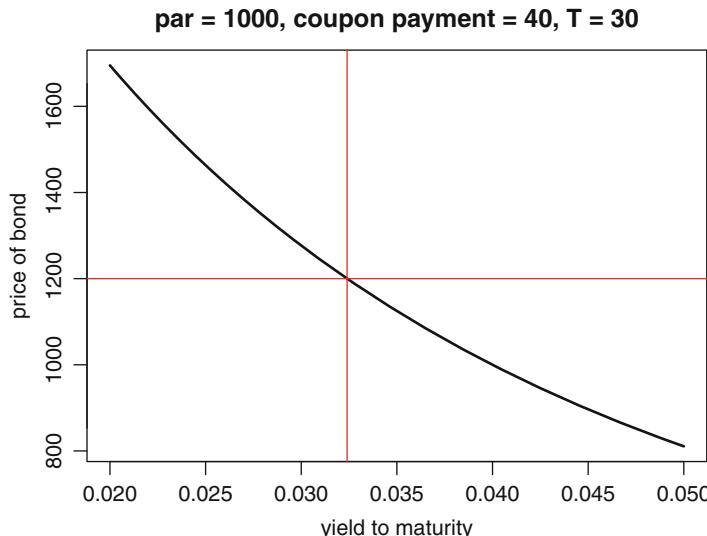


Fig. 3.1. Bond price versus yield to maturity. The horizontal red line is at the bond price of \$1,200. The price/yield curve intersects this line at 0.0324 as indicated by the vertical red line. Therefore, 0.0324 is the bond's yield.

One finds that the yield to maturity is 0.0324, that is, 3.24 %/half-year. Figure 3.1 shows the graph of bond price versus the yield (r) and shows that $r = 0.0324$ maps to a bond price of \$1,200.

The yield to maturity of 0.0324 is less than the current yield of 0.0333, which is less than the coupon rate of $40/1000 = 0.04$. (All three rates are rates per half-year.) Whenever, as in this example, the bond is selling above par

value, then the coupon rate is greater than the current yield because the bond sells above par value, and the current yield is greater than the yield to maturity because the yield to maturity accounts for the loss of capital when at the maturity date you get back only the par value, not the entire investment. In summary,

$$\text{price} > \text{par} \Rightarrow \text{coupon rate} > \text{current yield} > \text{yield to maturity}.$$

Everything is reversed if the bond is selling below par value. For example, if the price of the bond were only \$900, then the yield to maturity would be 0.0448 (as before, this value can be determined by interpolation), the current yield would be $40/900 = 0.0444$, and the coupon rate would still be $40/1000 = 0.04$. In general,

$$\text{price} < \text{par} \Rightarrow \text{coupon rate} < \text{current yield} < \text{yield to maturity}.$$

3.4.1 General Method for Yield to Maturity

The yield to maturity (on a semiannual basis) of a coupon bond is the value of r that solves

$$\text{PRICE} = \frac{C}{r} + \left\{ \text{PAR} - \frac{C}{r} \right\} (1+r)^{-2T}. \quad (3.7)$$

Here PRICE is the market price of the bond, PAR is the par value, C is the semiannual coupon payment, and T is the time to maturity in years and assumed to be a multiple of 1/2.

For a zero-coupon bond, $C = 0$ and (3.7) becomes

$$\text{PRICE} = \text{PAR}(1+r)^{-2T}. \quad (3.8)$$

3.4.2 Spot Rates

The yield to maturity of a zero-coupon bond of maturity n years is called the n -year *spot rate* and is denoted by y_n . One uses the n -year spot rate to discount a payment n years from now, so a payment of \$1 to be made n years from now has a net present value (NPV) of $\$1/(1+y_n)^n$ if y_n is the spot rate per annum or $\$1/(1+y_n)^{2n}$ if y_n is a semiannual rate.

A coupon bond is a bundle of zero-coupon bonds, one for each coupon payment and a final one for the principal payment. The component zeros have different maturity dates and therefore different spot rates. The yield to maturity of the coupon bond is, thus, a complex “average” of the spot rates of the zeros in this bundle.

Example 3.1. Finding the price and yield to maturity of a coupon bond using spot rates

Consider the simple example of 1-year coupon bond with semiannual coupon payments of \$40 and a par value of \$1,000. Suppose that the one-half-year spot rate is 2.5%/half-year and the 1-year spot rate is 3%/half-year. Think of the coupon bond as being composed of two zero-coupon bonds, one with $T = 1/2$ and a par value of \$40 and the second with $T = 1$ and a par value of \$1,040. The price of the bond is the sum of the prices of these two zeros. Applying (3.8) twice to obtain the prices of these zeros and summing, we obtain the price of the zero-coupon bond:

$$\frac{40}{1.025} + \frac{1040}{(1.03)^2} = 1019.32.$$

The yield to maturity on the coupon bond is the value of y that solves

$$\frac{40}{1+y} + \frac{1040}{(1+y)^2} = 1019.32.$$

The solution is $y = 0.0299/\text{half-year}$. Thus, the annual yield to maturity is twice 0.0299, or 5.98%/year. \square

General Formula

In this section we will find a formula that generalizes Example 3.1. Suppose that a coupon bond pays semiannual coupon payments of C , has a par value of PAR, and has T years until maturity. Let y_1, y_2, \dots, y_{2T} be the half-year spot rates for zero-coupon bonds of maturities $1/2, 1, 3/2, \dots, T$ years. Then the yield to maturity (on a half-year basis) of the coupon bond is the value of y that solves

$$\begin{aligned} & \frac{C}{1+y_1} + \frac{C}{(1+y_2)^2} + \cdots + \frac{C}{(1+y_{2T-1})^{2T-1}} + \frac{\text{PAR}+C}{(1+y_n)^{2T}} \\ &= \frac{C}{1+y} + \frac{C}{(1+y)^2} + \cdots + \frac{C}{(1+y)^{2T-1}} + \frac{\text{PAR}+C}{(1+y)^{2T}}. \end{aligned} \quad (3.9)$$

The left-hand side of Eq. (3.9) is the price of the coupon bond, and the yield to maturity is the value of y that makes the right-hand side of (3.9) equal to the price.

Methods for solving (3.9) are explored in the R lab in Sect. 3.10.

3.5 Term Structure

3.5.1 Introduction: Interest Rates Depend Upon Maturity

On January 26, 2001, the 1-year T-bill rate was 4.83% and the 30-year Treasury bond rate was 6.11%. This is typical. Short- and long-term rates usually differ. Often short-term rates are lower than long-term rates. This makes

sense since long-term bonds are riskier, because long-term bond prices fluctuate more with interest-rate changes. However, during periods of very high short-term rates, the short-term rates may be higher than the long-term rates. The reason is that the market believes that rates will return to historic levels and no one will commit to the high interest rate for, say, 20 or 30 years. Figure 3.2 shows weekly values of the 90-day, 10-year, and 30-year Treasury rates from 1970 to 1993, inclusive. Notice that the 90-day rate is more volatile than the longer-term rates and is usually less than them. However, in the early 1980s, when interest rates were very high, the short-term rates were higher than the long-term rates. These data were taken from the Federal Reserve Bank of Chicago's website.

The *term structure* of interest rates is a description of how, *at a given time*, yield to maturity depends on maturity.

3.5.2 Describing the Term Structure

Term structure for all maturities up to n years can be described by any one of the following:

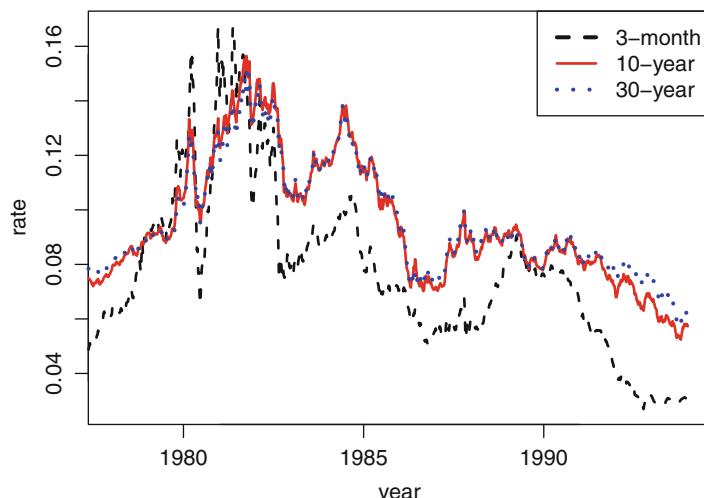


Fig. 3.2. Treasury rates of three maturities. Weekly time series. The data were taken from the website of the Federal Reserve Bank of Chicago.

- prices of zero-coupon bonds of maturities 1-year, 2-years, ..., n -years are denoted here by $P(1), P(2), \dots, P(n)$;
- spot rates (yields of maturity of zero-coupon bonds) of maturities 1-year, 2-years, ..., n -years are denoted by y_1, \dots, y_n ;

- forward rates r_1, \dots, r_n , where r_i is the forward rate that can be locked in now for borrowing in the i th future year ($i = 1$ for next year, and so on).

As discussed in this section, each of the sets $\{P(1), \dots, P(n)\}$, $\{y_1, \dots, y_n\}$, and $\{r_1, \dots, r_n\}$ can be computed from either of the other sets. For example, equation (3.11) ahead gives $\{P(1), \dots, P(n)\}$ in terms of $\{r_1, \dots, r_n\}$, and equations (3.12) and (3.13) ahead give $\{y_1, \dots, y_n\}$ in terms of $\{P(1), \dots, P(n)\}$ or $\{r_1, \dots, r_n\}$, respectively.

Term structure can be described by breaking down the time interval between the present time and the maturity time of a bond into short time segments with a constant interest rate within each segment, but with interest rates varying between segments. For example, a 3-year loan can be considered as three consecutive 1-year loans, or six consecutive half-year loans, and so forth.

Example 3.2. Finding prices from forward rates

As an illustration, suppose that loans have the forward interest rates listed in Table 3.1. Using the forward rates in the table, we see that a par \$1,000 1-year zero would sell for

$$\frac{1000}{1 + r_1} = \frac{1000}{1.06} = \$943.40 = P(1).$$

A par \$1,000 2-year zero would sell for

$$\frac{1000}{(1 + r_1)(1 + r_2)} = \frac{1000}{(1.06)(1.07)} = \$881.68 = P(2),$$

since the rate r_1 is paid the first year and r_2 the following year. Similarly, a par \$1,000 3-year zero would sell for

$$\frac{1000}{(1 + r_1)(1 + r_2)(1 + r_3)} = \frac{1000}{(1.06)(1.07)(1.08)} = 816.37 = P(3).$$

Table 3.1. Forward interest rates used in Examples 3.2 and 3.3

Year (i)	Interest rate (r_i)(%)
1	6
2	7
3	8

□

The general formula for the present value of \$1 paid n periods from now is

$$\frac{1}{(1+r_1)(1+r_2)\cdots(1+r_n)}. \quad (3.10)$$

Here r_i is the *forward interest rate* during the i th period. If the periods are years, then the price of an n -year par \$1,000 zero-coupon bond $P(n)$ is \$1,000 times the discount factor in (3.10); that is,

$$P(n) = \frac{1000}{(1+r_1)\cdots(1+r_n)}. \quad (3.11)$$

Example 3.3. Back to Example 3.2: Finding yields to maturity from prices and from the forward rates

In this example, we first find the yields to maturity from the prices derived in Example 3.2 using the interest rates from Table 3.1. For a 1-year zero, the yield to maturity y_1 solves

$$\frac{1000}{(1+y_1)} = 943.40,$$

which implies that $y_1 = 0.06$. For a 2-year zero, the yield to maturity y_2 solves

$$\frac{1000}{(1+y_2)^2} = 881.68,$$

so that

$$y_2 = \sqrt{\frac{1000}{881.68}} - 1 = 0.0650.$$

For a 3-year zero, the yield to maturity y_3 solves

$$\frac{1000}{(1+y_3)^3} = 816.37,$$

and equals 0.070.

The yields can also be found from the forward rates. First, trivially, $y_1 = r_1 = 0.06$. Next, y_2 is given by

$$y_2 = \sqrt{(1+r_1)(1+r_2)} - 1 = \sqrt{(1.06)(1.07)} - 1 = 0.0650.$$

Also,

$$\begin{aligned} y_3 &= \{(1+r_1)(1+r_2)(1+r_3)\}^{1/3} - 1 \\ &= \{(1.06)(1.07)(1.08)\}^{1/3} - 1 = 0.0700, \end{aligned}$$

or, more precisely, 0.06997. Thus, $(1+y_3)$ is the geometric average of 1.06, 1.07, and 1.08 and very nearly equal to their arithmetic average, which is 1.07.

□

Recall that $P(n)$ is the price of a par \$1,000 n -year zero-coupon bond. The general formulas for the yield to maturity y_n of an n -year zero are

$$y_n = \left\{ \frac{1000}{P(n)} \right\}^{1/n} - 1, \quad (3.12)$$

to calculate the yield from the price, and

$$y_n = \{(1 + r_1) \cdots (1 + r_n)\}^{1/n} - 1 \quad (3.13)$$

to obtain the yield from the forward rate.

Equations (3.12) and (3.13) give the yields to maturity in terms of the bond prices and forward rates, respectively. Also, inverting (3.12) gives the formula

$$P(n) = \frac{1000}{(1 + y_n)^n} \quad (3.14)$$

for $P(n)$ as a function of the yield to maturity.

As mentioned before, interest rates for future years are called *forward rates*. A forward contract is an agreement to buy or sell an asset at some fixed future date at a fixed price. Since r_2, r_3, \dots are rates that can be locked in now for future borrowing, they are forward rates.

The general formulas for determining forward rates from yields to maturity are

$$r_1 = y_1, \quad (3.15)$$

and

$$r_n = \frac{(1 + y_n)^n}{(1 + y_{n-1})^{n-1}} - 1, \quad n = 2, 3, \dots \quad (3.16)$$

Now suppose that we only observed bond prices. Then we can calculate yields to maturity and forward rates using (3.12) and then (3.16).

Table 3.2. Bond prices used in Example 3.4

Maturity	Price
1 Year	\$920
2 Years	\$830
3 Years	\$760

Example 3.4. Finding yields and forward rates from prices

Suppose that one-, two-, and three-year par \$1,000 zeros are priced as given in Table 3.2. Using (3.12), the yields to maturity are

$$\begin{aligned}y_1 &= \frac{1000}{920} - 1 = 0.087, \\y_2 &= \left\{ \frac{1000}{830} \right\}^{1/2} - 1 = 0.0976, \\y_3 &= \left\{ \frac{1000}{760} \right\}^{1/3} - 1 = 0.096.\end{aligned}$$

Then, using (3.15) and (3.16),

$$\begin{aligned}r_1 &= y_1 = 0.087, \\r_2 &= \frac{(1+y_2)^2}{(1+y_1)} - 1 = \frac{(1.0976)^2}{1.0876} - 1 = 0.108, \text{ and} \\r_3 &= \frac{(1+y_3)^3}{(1+y_2)^2} - 1 = \frac{(1.096)^3}{(1.0976)^2} - 1 = 0.092.\end{aligned}$$

□

The formula for finding r_n from the prices of zero-coupon bonds is

$$r_n = \frac{P(n-1)}{P(n)} - 1, \quad (3.17)$$

which can be derived from

$$P(n) = \frac{1000}{(1+r_1)(1+r_2)\cdots(1+r_n)},$$

and

$$P(n-1) = \frac{1000}{(1+r_1)(1+r_2)\cdots(1+r_{n-1})}.$$

To calculate r_1 using (3.17), we need $P(0)$, the price of a 0-year bond, but $P(0)$ is simply the par value.³

Example 3.5. Forward rates from prices

Thus, using (3.17) and the prices in Table 3.2, the forward rates are

$$\begin{aligned}r_1 &= \frac{1000}{920} - 1 = 0.087, \\r_2 &= \frac{920}{830} - 1 = 0.108,\end{aligned}$$

and

$$r_3 = \frac{830}{760} - 1 = 0.092.$$

□

³ Trivially, a bond that must be paid back immediately is worth exactly its par value.

3.6 Continuous Compounding

Now assume continuous compounding with forward rates r_1, \dots, r_n . Using continuously compounded rates simplifies the relationships among the forward rates, the yields to maturity, and the prices of zero-coupon bonds.

If $P(n)$ is the price of a \$1,000 par value n -year zero-coupon bond, then

$$P(n) = \frac{1000}{\exp(r_1 + r_2 + \dots + r_n)}. \quad (3.18)$$

Therefore,

$$\frac{P(n-1)}{P(n)} = \frac{\exp(r_1 + \dots + r_{n-1})}{\exp(r_1 + \dots + r_{n-1})} = \exp(r_n), \quad (3.19)$$

and

$$\log \left\{ \frac{P(n-1)}{P(n)} \right\} = r_n. \quad (3.20)$$

The yield to maturity of an n -year zero-coupon bond solves the equation

$$P(n) = \frac{1000}{\exp(ny_n)},$$

and is easily seen to be

$$y_n = (r_1 + \dots + r_n)/n. \quad (3.21)$$

Therefore, $\{r_1, \dots, r_n\}$ is easily found from $\{y_1, \dots, y_n\}$ by the relationship

$$r_1 = y_n,$$

and

$$r_n = ny_n - (n-1)y_{n-1} \text{ for } n > 1.$$

Example 3.6. Continuously compounded forward rates and yields from prices

Using the prices in Table 3.2, we have $P(1) = 920$, $P(2) = 830$, and $P(3) = 760$. Therefore, using (3.20),

$$r_1 = \log \left\{ \frac{1000}{920} \right\} = 0.083,$$

$$r_2 = \log \left\{ \frac{920}{830} \right\} = 0.103,$$

and

$$r_3 = \log \left\{ \frac{830}{760} \right\} = 0.088.$$

Also, $y_1 = r_1 = 0.083$, $y_2 = (r_1 + r_2)/2 = 0.093$, and $y_3 = (r_1 + r_2 + r_3)/3 = 0.091$. \square

3.7 Continuous Forward Rates

So far, we have assumed that forward interest rates vary from year to year but are constant within each year. This assumption is, of course, unrealistic and was made only to simplify the introduction of forward rates. Forward rates should be modeled as a function varying continuously in time.

To specify the term structure in a realistic way, we assume that there is a function $r(t)$ called the *forward-rate function* such that the current price of a zero-coupon bond of maturity T and with par value equal to 1 is given by

$$D(T) = \exp\left\{-\int_0^T r(t)dt\right\}. \quad (3.22)$$

$D(T)$ is called the discount function and the price of any zero-coupon bond is given by discounting its par value by multiplication with the discount function; that is,

$$P(T) = \text{PAR} \times D(T), \quad (3.23)$$

where $P(T)$ is the price of a zero-coupon bond of maturity T with par value equal to PAR. Also,

$$\log P(T) = \log(\text{PAR}) - \int_0^T r(t)dt,$$

so that

$$-\frac{d}{dT} \log P(T) = r(T) \text{ for all } T. \quad (3.24)$$

Formula (3.22) is a generalization of formula (3.18). To appreciate this, suppose that $r(t)$ is the piecewise constant function

$$r(t) = r_k \text{ for } k-1 < t \leq k.$$

With this piecewise constant r , for any integer T , we have

$$\int_0^T r(t)dt = r_1 + r_2 + \cdots + r_T,$$

so that

$$\exp\left\{-\int_0^T r(t)dt\right\} = \exp\{-(r_1 + \cdots + r_T)\}$$

and therefore (3.18) agrees with (3.22) in this special situation. However, (3.22) is a more general formula since it applies to noninteger T and to arbitrary $r(t)$, not only to piecewise constant functions.

The yield to maturity of a zero-coupon bond with maturity date T is defined to be

$$y_T = \frac{1}{T} \int_0^T r(t) dt. \quad (3.25)$$

Thinking of the right-hand side of (3.25) as the average of $r(t)$ over the interval $0 \leq t \leq T$, we see that (3.25) is the analog of (3.21). From (3.22) and (3.25) it follows that the discount function can be obtained from the yield to maturity by the formula

$$D(T) = \exp\{-Ty_T\}, \quad (3.26)$$

so that the price of a zero-coupon bond maturing at time T is the same as it would be if there were a constant forward interest rate equal to y_T . It follows from (3.26) that

$$y_T = -\log\{D(T)\}/T. \quad (3.27)$$

Example 3.7. Finding continuous yield and discount functions from forward rates

Suppose the forward rate is the linear function $r(t) = 0.03 + 0.0005t$. Find $r(15)$, y_{15} , and $D(15)$.

Answer: $r(15) = 0.03 + (0.0005)(15) = 0.0375$,

$$\begin{aligned} y_{15} &= (15)^{-1} \int_0^{15} (0.03 + 0.0005t) dt \\ &= (15)^{-1} (0.03t + 0.0005t^2/2) \Big|_0^{15} = 0.03375, \end{aligned}$$

and $D(15) = \exp(-15y_{15}) = \exp\{-(15)(0.03375)\} = \exp(-0.5055) = 0.603$. \square

The linear forward rate in Example 3.7 was chosen for simplicity and is not realistic. The Nelson-Siegel and Svensson parametric families of curves introduced in Sect. 11.3 are used in practice to model forward rates and yield curves. The European Community Bank uses the Svensson family. Nonparametric estimation of a forward rate by local polynomial and spline estimation is discussed in Examples 21.1 and 21.3, respectively. The Federal Reserve, the Bank of England, and the Bank of Canada use splines. The European Central Bank uses the Svensson family.

The discount function $D(T)$ and forward-rate function $r(t)$ in formula (3.22) depend on the current time, which is taken to be zero in that formula. However, we could be interested in how the discount function and forward rate function change over time. In that case we define the discount function $D(s, T)$ to be the price at time s of a zero-coupon bond, with a par value of \$1, maturing at time T . Also, if the forward-rate curve at time s is $r(s, t)$, $t \geq s$, then

$$D(s, T) = \exp \left\{ - \int_s^T r(s, t) dt \right\}. \quad (3.28)$$

The yield at time s of a bond maturing at time $T > s$ is

$$y(s, T) = (T - s)^{-1} \int_s^T r(s, u) du.$$

Since $r(t)$ and $D(t)$ in (3.22) are $r(0, t)$ and $D(0, t)$ in our new notation, (3.22) is the special case of (3.28) with $s = 0$. Similarly, y_T is equal to $y(0, T)$ in the new notation. However, for the remainder of this chapter we assume that $s = 0$ and return to the simpler notation of $r(t)$ and $D(t)$.

3.8 Sensitivity of Price to Yield

As we have seen, bonds are risky because bond prices are sensitive to interest rates. This problem is called *interest-rate risk*. This section describes a traditional method of quantifying interest-rate risk.

Using Eq. (3.26), we can approximate how the price of a zero-coupon bond changes if there is a small change in yield. Suppose that y_T changes to $y_T + \delta$, where the change in yield δ is small. Then the change in $D(T)$ is approximately δ times

$$\frac{d}{dy_T} \exp\{-Ty_T\} \approx -T \exp\{-Ty_T\} = -TD(T). \quad (3.29)$$

Therefore, by Eq. (3.23), for a zero-coupon bond of maturity T ,

$$\frac{\text{change bond price}}{\text{bond price}} \approx -T \times \text{change in yield}. \quad (3.30)$$

In this equation “ \approx ” means that the ratio of the right- to left-hand sides converges to 1 as $\delta \rightarrow 0$.

Equation (3.30) is worth examining. The minus sign on the right-hand side shows us something we already knew, that bond prices move in the opposite direction to interest rates. Also, the relative change in the bond price, which is the left-hand side of the equation, is proportional to T , which quantifies the principle that longer-term bonds have higher interest-rate risks than short-term bonds.

3.8.1 Duration of a Coupon Bond

Remember that a coupon bond can be considered a bundle of zero-coupon bonds of various maturities. The *duration* of a coupon bond, which we will denote by DUR, is the weighted average of these maturities with weights in

proportion to the net present value of the cash flows (coupon payments and par value at maturity).

Now assume that all yields change by a constant amount δ , that is, y_T changes to $y_T + \delta$ for all T . This restrictive assumption is needed to define duration. Because of this assumption, Eq. (3.30) applies to each of these cash flows and averaging them with these weights gives us that for a coupon bond,

$$\frac{\text{change bond price}}{\text{bond price}} \approx -\text{DUR} \times \delta. \quad (3.31)$$

The details of the derivation of (3.31) are left as an exercise (Exercise 15). *Duration analysis* uses (3.31) to approximate the effect of a change in yield on bond prices.

We can rewrite (3.31) as

$$\text{DUR} \approx \frac{-1}{\text{price}} \times \frac{\text{change in price}}{\text{change in yield}} \quad (3.32)$$

and use (3.32) as a *definition* of duration. Notice that “bond price” has been replaced by “price.” The reason for this is that (3.32) can define the durations of not only bonds but also of derivative securities whose prices depend on yield, for example, call options on bonds. When this definition is extended to derivatives, duration has nothing to do with maturities of the underlying securities. Instead, duration is solely a measure of sensitivity of price to yield. Tuckman (2002) gives an example of a 10-year coupon bond with a duration of 7.79 years and a call option on this bond with a duration of 120.82 years. These durations show that the call is much riskier than the bond since it is 15.5 ($= 129.82/7.79$) times more sensitive to changes in yield.

Unfortunately, the underlying assumption behind (3.31) that all yields change by the same amount is not realistic, so duration analysis is falling into disfavor and value-at-risk is replacing duration analysis as a method for evaluating interest-rate risk.⁴ Value-at-risk and other risk measures are covered in Chap. 19.

3.9 Bibliographic Notes

Tuckman (2002) is an excellent comprehensive treatment of fixed income securities; it is written at an elementary mathematical level and is highly recommended for readers wishing to learn more about this topic. Bodie, Kane, and Marcus (1999), Sharpe, Alexander, and Bailey (1999), and Campbell, Lo, and MacKinlay (1997) provide good introductions to fixed income securities, with the last-named being at a more advanced level. James and Webber (2000) is an advanced book on interest rate modeling. Jarrow (2002) covers

⁴ See Dowd (1998).

many advanced topics that are not included in this book, including modeling the evolution of term structure, bond trading strategies, options and futures on bonds, and interest-rate derivatives.

3.10 R Lab

3.10.1 Computing Yield to Maturity

The following R function computes the price of a bond given its coupon payment, maturity, yield to maturity, and par value.

```
bondvalue = function(c, T, r, par)
{
  #      Computes bv = bond values (current prices) corresponding
  #      to all values of yield to maturity in the
  #      input vector r
  #
  #      INPUT
  #      c = coupon payment (semiannual)
  #      T = time to maturity (in years)
  #      r = vector of yields to maturity (semiannual rates)
  #      par = par value
  #
  bv = c / r + (par - c / r) * (1 + r)^(-2 * T)
  bv
}
```

The R code that follows computes the price of a bond for 300 semiannual interest rates between 0.02 and 0.05 for a 30-year par \$1,000 bond with coupon payments of \$40. Then interpolation is used to find the yield to maturity if the current price is \$1,200.

```
price = 1200      #    current price of the bond
C = 40            #    coupon payment
T= 30             #    time to maturity
par = 1000         #    par value of the bond

r = seq(0.02, 0.05, length = 300)
value = bondvalue(C, T, r, par)
yield2M = spline(value, r, xout = price) # spline interpolation
```

The final bit of R code below plots price as a function of yield to maturity and graphically interpolates to show the yield to maturity when the price is \$1,200.

```
plot(r, value, xlab = 'yield to maturity', ylab = 'price of bond',
     type = "l", main = "par = 1000, coupon payment = 40,
```

```
T = 30", lwd = 2)
abline(h = 1200)
abline(v = yield2M)
```

Problem 1 Use the plot to estimate graphically the yield to maturity. Does this estimate agree with that from spline interpolation?

As an alternative to interpolation, the yield to maturity can be found using a nonlinear root finder (equation solver) such as `uniroot()`, which is illustrated here:

```
uniroot(function(r) r^2 - .5, c(0.7, 0.8))
```

Problem 2 What does the code

```
uniroot(function(r) r^2 - 0.5, c(0.7, 0.8))
```

do?

Problem 3 Use `uniroot()` to find the yield to maturity of the 30-year par \$1,000 bond with coupon payments of \$40 that is selling at \$1,200.

Problem 4 Find the yield to maturity of a par \$10,000 bond selling at \$9,800 with semiannual coupon payments equal to \$280 and maturing in 8 years.

Problem 5 Use `uniroot()` to find the yield to maturity of the 20-year par \$1,000 bond with semiannual coupon payments of \$35 that is selling at \$1,050.

Problem 6 The yield to maturity is 0.035 on a par \$1,000 bond selling at \$950.10 and maturing in 5 years. What is the coupon payment?

3.10.2 Graphing Yield Curves

R's `fEcofin` package had many interesting financial data sets but is no longer available. The data sets `mk.maturity.csv` and `mk.zero2.csv` used in this example were taken from this package and are now available on this book's webpage. The data set `mk.zero2` has yield curves of U.S. zero coupon bonds recorded monthly at 55 maturities. These maturities are in the data set `mk.maturity`. The following code plots the yield curves on four consecutive months.

```

mk.maturity = read.csv("mk.maturity.csv", header = T)
mk.zero2 = read.csv("mk.zero2.csv", header = T)
plot(mk.maturity[,1], mk.zero2[5,2:56], type = "l",
     xlab = "maturity", ylab = "yield")
lines(mk.maturity[,1], mk.zero2[6,2:56], lty = 2, type = "l")
lines(mk.maturity[,1], mk.zero2[7,2:56], lty = 3, type = "l")
lines(mk.maturity[,1], mk.zero2[8,2:56], lty = 4, type = "l")
legend("bottomright", c("1985-12-01", "1986-01-01",
                        "1986-02-01", "1986-03-01"), lty = 1:4)

```

Run the code above and then, to zoom in on the short end of the curves, rerun the code with maturities restricted to 0 to 3 years; to do that, use `xlim` in the `plot` function.

Problem 7 *Describe how the yield curve changes between December 1, 1985 and March 1, 1986. Describe the behavior of both the short and long ends of the yield curves.*

Problem 8 *Plot the yield curves from December 1, 1986 to March 1, 1987 and describe how the yield curve changes during this period.*

The next set of code estimates the forward rate for 1 month. Line 1 estimates the integrated forward rate, called `intForward`, which is $Ty_T = \int_0^T r(t)dt$ where $r(t)$ is the forward rate. Line 3 interpolates the estimated integrated forward rate onto a grid of 200 points from 0 to 20. This grid is created on line 2.

If a function f is evaluated on a grid, t_1, \dots, t_L , then $\{f(t_\ell) - f(t_{\ell-1})\}/(t_\ell - t_{\ell-1})$ approximates $f'((t_\ell + t_{\ell-1})/2)$ for $\ell = 2, \dots, L$. Line 4 numerically differentiates the integrated forward rate to approximate the forward rate on the grid calculated at Line 5.

```

1 intForward = mk.maturity[, 1] * mk.zero2[6, 2:56]
2 xout = seq(0, 20, length = 200)
3 z1 = spline(mk.maturity[, 1], intForward, xout = xout)
4 forward = diff(z1$y) / diff(z1$x)
5 T_grid = (xout[-1] + xout[-200]) / 2
6 plot(T_grid, forward, type = "l", lwd = 2, ylim = c(0.06, 0.11))

```

Problem 9 *Plot the forward rates on the same dates used before, 1985-12-01, 1986-01-01, 1986-02-01, and 1986-03-01. Describe how the forward rates changed from month to month.*

The approximate forward rates found by numerically differentiating a interpolating spline are “wiggly.” The wiggles can be removed, or at least reduced, by using a penalized spline instead of an interpolating spline. See Chap. 21.

3.11 Exercises

1. Suppose that the forward rate is $r(t) = 0.028 + 0.00042t$.
 - (a) What is the yield to maturity of a bond maturing in 20 years?
 - (b) What is the price of a par \$1,000 zero-coupon bond maturing in 15 years?
2. Suppose that the forward rate is $r(t) = 0.04 + 0.0002t - 0.00003t^2$.
 - (a) What is the yield to maturity of a bond maturing in 8 years?
 - (b) What is the price of a par \$1,000 zero-coupon bond maturing in 5 years?
 - (c) Plot the forward rate and the yield curve. Describe the two curves. Which are convex and which are concave? How do they differ?
 - (d) Suppose you buy a 10-year zero-coupon bond and sell it after 1 year. What will be the return if the forward rate does not change during that year?
3. A coupon bond has a coupon rate of 3% and a current yield of 2.8%.
 - (a) Is the bond selling above or below par? Why or why not?
 - (b) Is the yield to maturity above or below 2.8%? Why or why not?
4. Suppose that the forward rate is $r(t) = 0.032 + 0.001t + 0.0002t^2$.
 - (a) What is the 5-year continuously compounded spot rate?
 - (b) What is the price of a zero-coupon bond that matures in 5 years?
5. The 1/2-, 1-, 1.5-, and 2-year semiannually compounded spot rates are 0.025, 0.028, 0.032, and 0.033, respectively. A par \$1,000 coupon bond matures in 2 years and has semiannual coupon payments of \$35. What is the price of this bond?
6. Verify the following equality:

$$\sum_{t=1}^{2T} \frac{C}{(1+r)^t} + \frac{\text{PAR}}{(1+r)^{2T}} = \frac{C}{r} + \left\{ \text{PAR} - \frac{C}{r} \right\} (1+r)^{-2T}.$$

7. One year ago a par \$1,000 20-year coupon bond with semiannual coupon payments was issued. The annual interest rate (that is, the coupon rate) at that time was 8.5%. Now, a year later, the annual interest rate is 7.6%.
 - (a) What are the coupon payments?
 - (b) What is the bond worth now? Assume that the second coupon payment was just received, so the bondholder receives an additional 38 coupon payments, the next one in 6 months.
 - (c) What would the bond be worth if instead the second payment were just about to be received?
8. A par \$1,000 zero-coupon bond that matures in 5 years sells for \$828. Assume that there is a constant continuously compounded forward rate r .
 - (a) What is r ?
 - (b) Suppose that 1 year later the forward rate r is still constant but has changed to be 0.042. Now what is the price of the bond?

- (c) If you bought the bond for the original price of \$828 and sold it 1 year later for the price computed in part (b), then what is the net return?
9. A coupon bond with a par value of \$1,000 and a 10-year maturity pays semiannual coupons of \$21.
- Suppose the yield for this bond is 4% per year compounded semiannually. What is the price of the bond?
 - Is the bond selling above or below par value? Why?
10. Suppose that a coupon bond with a par value of \$1,000 and a maturity of 7 years is selling for \$1,040. The semiannual coupon payments are \$23.
- Find the yield to maturity of this bond.
 - What is the current yield on this bond?
 - Is the yield to maturity less or greater than the current yield? Why?
11. Suppose that the continuous forward rate is $r(t) = 0.033 + 0.0012t$. What is the current value of a par \$100 zero-coupon bond with a maturity of 15 years?
12. Suppose the continuous forward rate is $r(t) = 0.04 + 0.001t$ when a 8-year zero coupon bond is purchased. Six months later the forward rate is $r(t) = 0.03 + 0.0013t$ and bond is sold. What is the return?
13. Suppose that the continuous forward rate is $r(t) = 0.03 + 0.001t - 0.00021(t - 10)_+$. What is the yield to maturity on a 20-year zero-coupon bond? Here x_+ is the *positive part function* defined by

$$x_+ = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

14. An investor is considering the purchase of zero-coupon bonds with maturities of one, three, or 5 years. Currently the spot rates for 1-, 2-, 3-, 4-, and 5-year zero-coupon bonds are, respectively, 0.031, 0.035, 0.04, 0.042, and 0.043 per year with semiannual compounding. A financial analyst has advised this investor that interest rates will increase during the next year and the analyst expects all spot rates to increase by the amount 0.005, so that the 1-year spot rate will become 0.036, and so forth. The investor plans to sell the bond at the end of 1 year and wants the greatest return for the year. This problem does the bond math to see which maturity, 1, 3, or 5 years, will give the best return under two scenarios: interest rates are unchanged and interest rates increase as forecast by the analyst.
- What are the current prices of 1-, 3-, and 5-year zero-coupon bonds with par values of \$1,000?
 - What will be the prices of these bonds 1 year from now if spot rates remain unchanged?
 - What will be the prices of these bonds 1 year from now if spot rates each increase by 0.005?
 - If the analyst is correct that spot rates will increase by 0.005 in 1 year, which maturity, 1, 3, or 5 years, will give the investor the greatest return when the bond is sold after 1 year? Justify your answer.

- (e) If instead the analyst is incorrect and spot rates remain unchanged, then which maturity, 1, 3, or 5 years, earns the highest return when the bond is sold after 1 year? Justify your answer.
- (f) The analyst also said that if the spot rates remain unchanged, then the bond with the highest spot rate will earn the greatest 1-year return. Is this correct? Why?

(Hint: Be aware that a bond will not have the same maturity in 1 year as it has now, so the spot rate that applies to that bond will change.)

15. Suppose that a bond pays a cash flow C_i at time T_i for $i = 1, \dots, N$. Then the net present value (NPV) of cash flow C_i is

$$\text{NPV}_i = C_i \exp(-T_i y_{T_i}).$$

Define the weights

$$\omega_i = \frac{\text{NPV}_i}{\sum_{j=1}^N \text{NPV}_j}$$

and define the duration of the bond to be

$$\text{DUR} = \sum_{i=1}^N \omega_i T_i,$$

which is the weighted average of the times of the cash flows. Show that

$$\frac{d}{d\delta} \sum_{i=1}^N C_i \exp\{-T_i(y_{T_i} + \delta)\} \Big|_{\delta=0} = -\text{DUR} \sum_{i=1}^N C_i \exp\{-T_i y_{T_i}\}$$

and use this result to verify Eq. (3.31).

16. Assume that the yield curve is $Y_T = 0.04 + 0.001 T$.
- (a) What is the price of a par-\$1,000 zero-coupon bond with a maturity of 10 years?
 - (b) Suppose you buy this bond. If 1 year later the yield curve is $Y_T = 0.042 + 0.001 T$, then what will be the net return on the bond?
17. A coupon bond has a coupon rate of 3% and a current yield of 2.8%.
- (a) Is the bond selling above or below par? Why or why not?
 - (b) Is the yield to maturity above or below 2.8%? Why or why not?
18. Suppose that the forward rate is $r(t) = 0.03 + 0.001t + 0.0002t^2$
- (a) What is the 5-year spot rate?
 - (b) What is the price of a zero-coupon bond that matures in 5 years?
19. The 1/2-, 1-, 1.5-, and 2-year spot rates are 0.025, 0.029, 0.031, and 0.035, respectively. A par \$1,000 coupon bond matures in 2 years and has semi-annual coupon payments of \$35. What is the price of this bond?
20. Par \$1,000 zero-coupon bonds of maturities of 0.5-, 1-, 1.5-, and 2-years are selling at \$980.39, \$957.41, \$923.18, and \$888.489, respectively.
- (a) Find the 0.5-, 1-, 1.5-, and 2-year semiannual spot rates.

- (b) A par \$1,000 coupon bond has a maturity of 2 years. The semiannual coupon payment is \$21. What is the price of this bond?
21. A par \$1,000 bond matures in 4 years and pays semiannual coupon payments of \$25. The price of the bond is \$1,015. What is the semiannual yield to maturity of this bond?
22. A coupon bond matures in 4 years. Its par is \$1,000 and it makes eight coupon payments of \$21, one every one-half year. The continuously compounded forward rate is

$$r(t) = 0.022 + 0.005t - 0.004t^2 + 0.0003t^3.$$

- (a) Find the price of the bond.
 (b) Find the duration of this bond.

References

- Bodie, Z., Kane, A., and Marcus, A. (1999) *Investments*, 4th ed., Irwin/McGraw-Hill, Boston.
- Campbell, J. Y., Lo, A. W., and MacKinlay, A. C. (1997) *Econometrics of Financial Markets*, Princeton University Press, Princeton, NJ.
- Dowd, K. (1998) *Beyond Value at Risk*, Wiley, Chichester.
- James, J., and Webber, N. (2000) *Interest Rate Modeling*, Wiley, Chichester.
- Jarrow, R. (2002) *Modeling Fixed-Income Securities and Interest Rate Options*, 2nd ed., Stanford University Press, Stanford, CA.
- Sharpe, W., Alexander, G., and Bailey, J. (1999) *Investments*, 6th ed., Prentice-Hall, Englewood Cliffs, NJ.
- Tuckman, B. (2002) *Fixed Income Securities*, 2nd ed., Wiley, Hoboken, NJ.

Exploratory Data Analysis

4.1 Introduction

This book is about the statistical analysis of financial markets data such as equity prices, foreign exchange rates, and interest rates. These quantities vary randomly thereby causing financial risk as well as the opportunity for profit. Figures 4.1, 4.2, and 4.3 show, respectively, time series plots of daily log returns on the S&P 500 index, daily changes in the Deutsch Mark (DM) to U.S. dollar exchange rate, and changes in the monthly risk-free return, which is 1/12th the annual risk-free interest rate. A *time series* is a sequence of observations

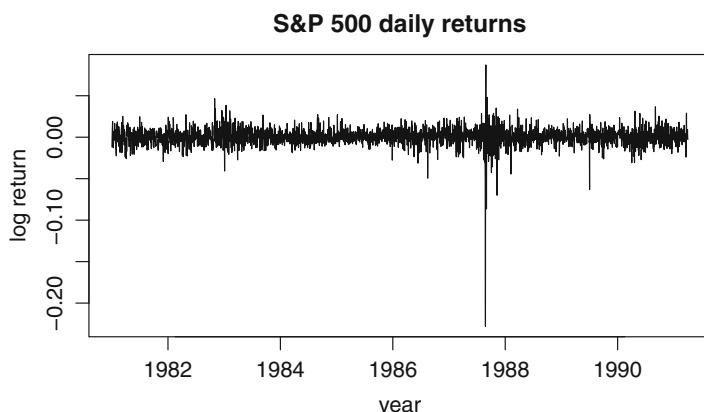


Fig. 4.1. Daily log returns on the S&P 500 index from January 1981 to April 1991. This data set is the variable `r500` in the `SP500` series in the `Ecdat` package in R. Notice the extreme volatility in October 1987.

of some quantity or quantities, e.g., equity prices, taken over time, and a *time series plot* is a plot of a time series in chronological order. Figure 4.1 was produced by the following code:

```
data(SP500, package = "Ecdat")
SPreturn = SP500$r500
n = length(SPreturn)
year_SP = 1981 + (1:n) * (1991.25 - 1981) / n
plot(year_SP, SPreturn, main = "S&P 500 daily returns",
     xlab = "year", type = "l", ylab = "log return")
```

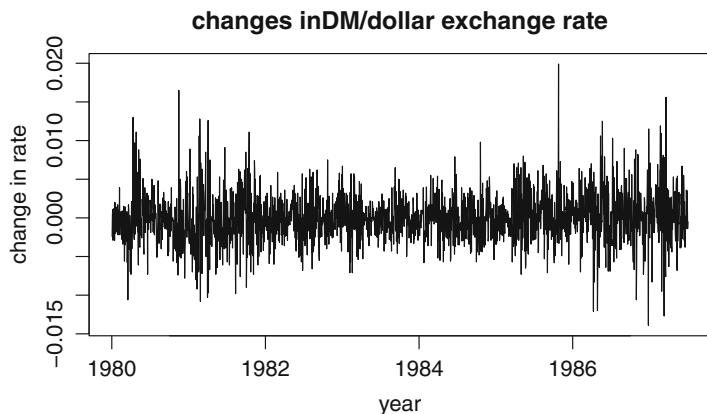


Fig. 4.2. Daily changes in the DM/dollar exchange rate, January 2, 1980, to May 21, 1987. The data come from the Garch series in the Ecdat package in R. The DM/dollar exchange rate is the variable dm.

Despite the large random fluctuations in all three time series, we can see that each series appears *stationary*, meaning that the nature of its random variation is constant over time. In particular, the series fluctuate about means that are constant, or nearly so. We also see *volatility clustering*, because there are periods of higher, and of lower, variation within each series. Volatility clustering does *not* indicate a lack of stationarity but rather can be viewed as a type of dependence in the conditional variance of each series. This point will be discussed in detail in Chap. 14.

Each of these time series will be modeled as a sequence Y_1, Y_2, \dots of random variables, each with a CDF that we will call F .¹ F will vary between series but, because of stationarity, is assumed to be constant within each series. F is also called the marginal distribution function. By the *marginal distribution* of a stationary time series, we mean the distribution of Y_t given no knowledge

¹ See Appendix A.2.1 for definitions of CDF, PDF, and other terms in probability theory.

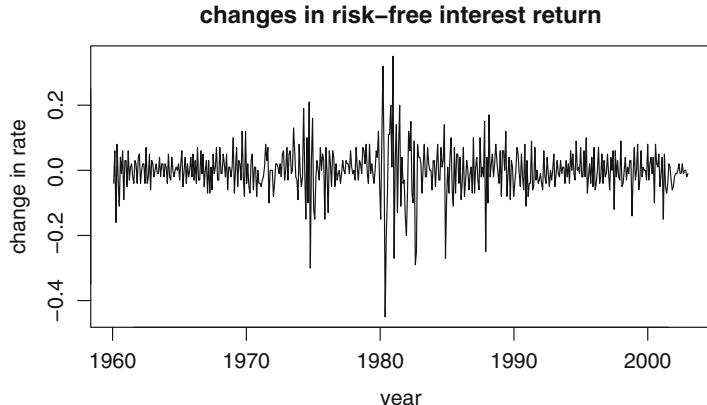


Fig. 4.3. Monthly changes in the risk-free rate, January 1960 to December 2002. The rates are the variable `rf` in the `Capm` series in the `Ecdat` package in R.

of the other observations, that is, no knowledge of Y_s for any $s \neq t$. Thus, when modeling a marginal distribution, we disregard dependencies in the time series. For this reason, a marginal distribution is also called an *unconditional distribution*. Dependencies such as autocorrelation and volatility clustering will be discussed in later chapters.

In this chapter, we explore various methods for modeling and estimating marginal distributions, in particular, graphical methods such as histograms, density estimates, sample quantiles, and probability plots.

4.2 Histograms and Kernel Density Estimation

Assume that the marginal CDF F has a probability density function f . The histogram is a simple and well-known estimator of probability density functions. Panel (a) of Fig. 4.4 is a histogram of the S&P 500 log returns using 30 cells (or bins). There are some outliers in this series, especially a return near -0.23 that occurred on Black Monday, October 19, 1987. Note that a return of this size means that the market lost 23 % of its value in a single day. The outliers are difficult, or perhaps impossible, to see in the histogram, except that they have caused the x -axis to expand. The reason that the outliers are difficult to see is the large sample size. When the sample size is in the thousands, a cell with a small frequency is essentially invisible. Panel (b) of Fig. 4.4 zooms in on the high-probability region. Note that only a few of the 30 cells are in this area.

The histogram is a fairly crude density estimator. A typical histogram looks more like a big city skyline than a density function and its appearance is sensitive to the number and locations of its cells—see Fig. 4.4, where panels (b), (c), and (d) differ only in the number of cells. A much better estimator is

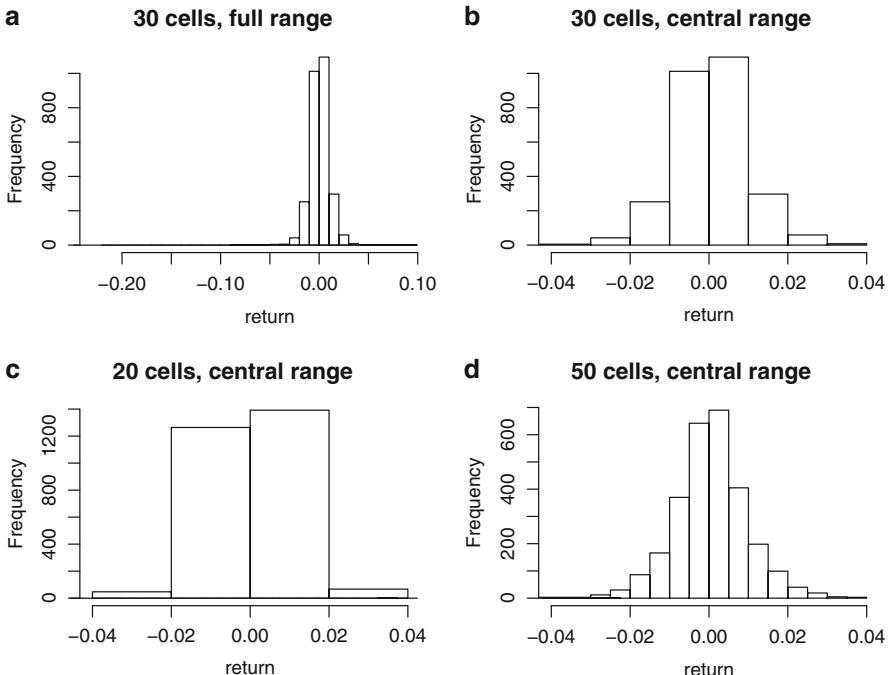


Fig. 4.4. Histograms of the daily log returns on the S&P 500 index from January 1981 to April 1991. This data set is the SP500 series in the Ecdat package in R.

the *kernel density estimator* (KDE). The estimator takes its name from the so-called kernel function, denoted here by K , which is a probability density function that is symmetric about 0. The standard² normal density function is a common choice for K and will be used here. The kernel density estimator based on Y_1, \dots, Y_n is

$$\hat{f}(y) = \frac{1}{nb} \sum_{i=1}^n K\left(\frac{y - Y_i}{b}\right) \quad (4.1)$$

where b , which is called the bandwidth, determines the resolution of the estimator.

Figure 4.5 illustrates the construction of kernel density estimates using a small simulated data set of six observations from a standard normal distribution. The small sample size is needed for visual clarity but, of course, does not lead to an accurate estimate of the underlying normal density. The six data points are shown at the bottom of the figure as short vertical lines called a “rug.” The bandwidth in the top plot is 0.4, and so each of the six dashed lines is $1/6$ times a normal density with standard deviation equal to 0.4 and

² “Standard” means having expectation 0 and variance 1.

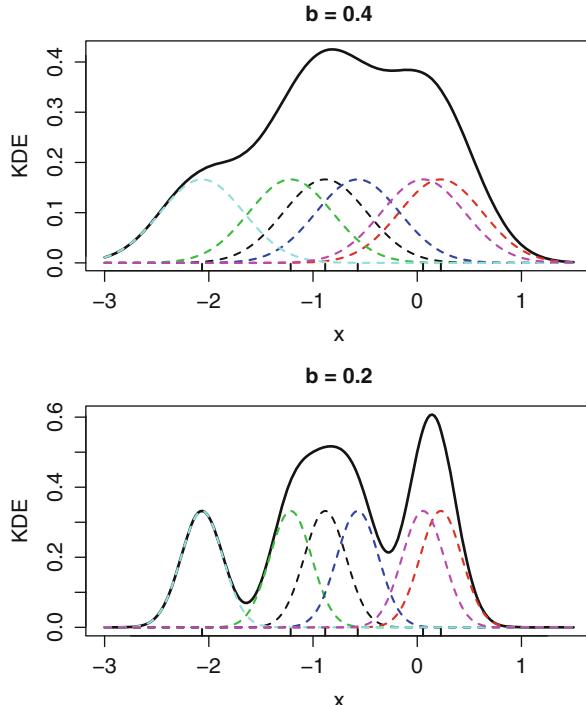


Fig. 4.5. Illustration of kernel density estimates using a sample of size 6 and two bandwidths. The six dashed curves are the kernels centered at the data points, which are indicated by vertical lines at the bottom. The solid curve is the kernel density estimate created by adding together the six kernels. Although the same data are used in the top and bottom panels, the density estimates are different because of the different bandwidths.

centered at one of the data points. The solid curve is the superposition, that is, the sum as in Eq. (4.1), of the six dashed curves and estimates the density of the data.

A small value of b allows the density estimator to detect fine features in the true density, but it also permits a high degree of random variation. This can be seen in the plot in the bottom of Fig. 4.5 where the bandwidth is only half as large as in the plot on the top. Conversely, a large value of b dampens random variation but obscures fine detail in the true density. Stated differently, a small value of b causes the kernel density estimator to have high variance and low bias, and a large value of b results in low variance and high bias.

Choosing b requires one to make a tradeoff between bias and variance. Appropriate values of b depend on both the sample size n and the true density and, of course, the latter is unknown, though it can be estimated. Roughly speaking, nonsmooth or “wiggly” densities require a smaller bandwidth.

Fortunately, a large amount of research has been devoted to automatic selection of b , which, in effect, estimates the roughness of the true density. As a result of this research, modern statistical software can select the bandwidth automatically. However, automatic bandwidth selectors are not foolproof and density estimates should be checked visually and, if necessary, adjusted as described below.

The solid curve in Fig. 4.6 has the default bandwidth from the `density()` function in R. The dashed and dotted curves have the default bandwidth multiplied by $1/3$ and 3 , respectively. The tuning parameter `adjust` in R is the multiplier of the default bandwidth, so that `adjust` is 1 , $1/3$, and 3 in the three curves. The solid curve with `adjust` equal to 1 appears to have a proper amount of smoothness. The dashed curve corresponding to `adjust = 1/3` is wiggly, indicating too much random variability; such a curve is called under-smoothed and overfit. The dotted curve is very smooth but underestimates the peak near 0 , a sign of bias. Such a curve is called oversmoothed or underfit. Here *overfit* means that the density estimate adheres too closely to the data and so is unduly influenced by random variation. Conversely, *underfit* means that the density estimate does not adhere closely enough to the data and misses features in the true density. Stated differently, over- and underfitting means a poor bias-variance tradeoff with an overfitted curve having too much variance and an underfitted curve having too much bias.

Automatic bandwidth selectors are very useful, but there is nothing magical about them, and often one will use an automatic selector as a starting

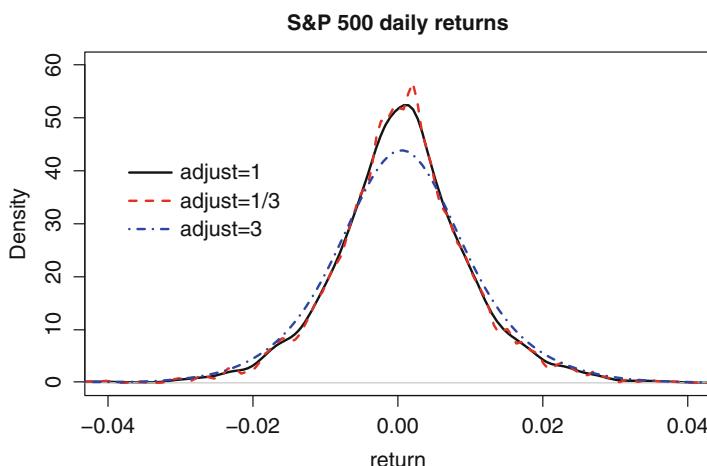


Fig. 4.6. Kernel density estimates of the daily log returns on the S&P 500 index using three bandwidths. Each bandwidth is the default bandwidth times `adjust` and `adjust` is $1/3$, 1 , and 3 . This data set is the `SP500` series in the `Ecdat` package in R. The KDE is plotted only for a limited range of returns to show detail in the middle of the distribution.

point and then “fine-tune” the bandwidth; this is the point of the `adjust` parameter. Generally, `adjust` will be much closer to 1 than the values, 1/3 and 3, used above. The reason for using 1/3 and 3 in Fig. 4.6 was to emphasize the effects of under- and oversmoothing.

Often a kernel density estimate is used to suggest a parametric statistical model. The density estimates in Fig. 4.6 are bell-shaped, suggesting that a normal distribution might be a suitable model. To further investigate the suitability of the normal model, Fig. 4.7 compares the kernel density estimate with `adjust = 1` with normal densities. In panel (a), the normal density has mean and standard deviation equal to the sample mean and standard deviation of the returns. We see that the kernel estimate and the normal density are somewhat dissimilar. The reason is that the outlying returns inflate the sample standard deviation and cause the fitted normal density to be too dispersed in the middle of the data. Panel (b) shows a normal density that is much closer to the kernel estimator. This normal density uses robust estimators which are less sensitive to outliers—the mean is estimated by the sample median and the MAD estimator is used for the standard deviation. The MAD estimator is the median absolute deviation from the median but scaled so that it estimates the standard deviation of a normal population.³ The sample

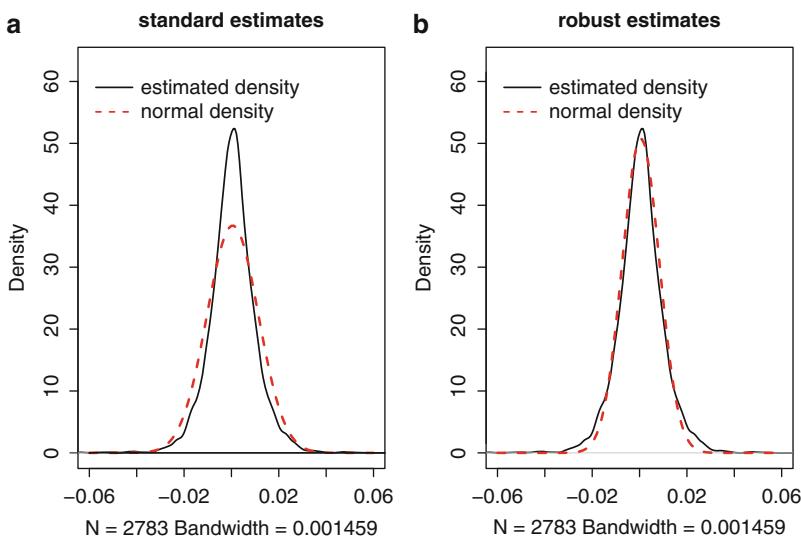


Fig. 4.7. Kernel density estimates (solid) of the daily log returns on the S&P 500 index compared with normal densities (dashed). (a) The normal density uses the sample mean and standard deviation. (b) The normal density uses the sample median and MAD estimate of standard deviation. This data set is the `SP500` series in the `Ecdat` package in R.

³ See Sect. 5.16 for more discussion of robust estimation and the precise definition of MAD.

standard deviation is 0.011, but the MAD is smaller, 0.0079; these values were computed using the R functions `sd()` and `mad()`. Even the normal density in panel (b) shows some deviation from the kernel estimator, and, as we will soon see, the t -distribution provides a better model for the return distribution than does the normal distribution. The need for robust estimators is itself a sign of nonnormality.

We have just seen a problem with using a KDE to suggest a good model for the distribution of the data in a sample—the parameters in the model must be estimated properly. Normal probability plots and, more generally, quantile–quantile plots, which will be discussed in Sects. 4.3.2 and 4.3.4, are better methods for comparing a sample with a theoretical distribution.

Though simple to compute, the KDE has some problems. In particular, it is often too bumpy in the tails. An improvement to the KDE is discussed in Sect. 4.8.

4.3 Order Statistics, the Sample CDF, and Sample Quantiles

Suppose that Y_1, \dots, Y_n is a random sample from a probability distribution with CDF F . In this section we estimate F and its quantiles. The *sample* or *empirical CDF* $F_n(y)$ is defined to be the proportion of the sample that is less than or equal to y . For example, if 10 out of 40 ($= n$) elements of a sample are 3 or less, then $F_n(3) = 0.25$. More generally,

$$F_n(y) = \frac{\sum_{i=1}^n I\{Y_i \leq y\}}{n}, \quad (4.2)$$

where $I\{\cdot\}$ is the indicator function so that $I\{Y_i \leq y\}$ is 1 if $Y_i \leq y$ and is 0 otherwise. Therefore, the sum in the numerator in (4.2) counts the number of Y_i that are less than or equal to y . Figure 4.8 shows F_n for a sample of size 150 from an $N(0, 1)$ distribution. The true CDF (Φ) is shown as well. The sample CDF differs from the true CDF because of random variation. The sample CDF is also called the empirical distribution function, or EDF.

The function `ecdf()` computes a sample CDF. The code to produce Fig. 4.8 is:

```

1 set.seed("991155")
2 edf_norm = ecdf(rnorm(150))
3 pdf("normalcdfplot.pdf", width = 6, height = 5) ## Figure 4.8
4 par(mfrow = c(1, 1))
5 plot(edf_norm, verticals = TRUE, do.p = FALSE, main = "EDF and CDF")
6 tt = seq(from = -3, to = 3, by = 0.01)
7 lines(tt, pnorm(tt), lty = 2, lwd = 2, col = "red")
8 legend(1.5, 0.2, c("EDF", "CDF"), lty = c(1, 2),
9        lwd = c(1.5, 2), col = c("black", "red"))
10 graphics.off()

```

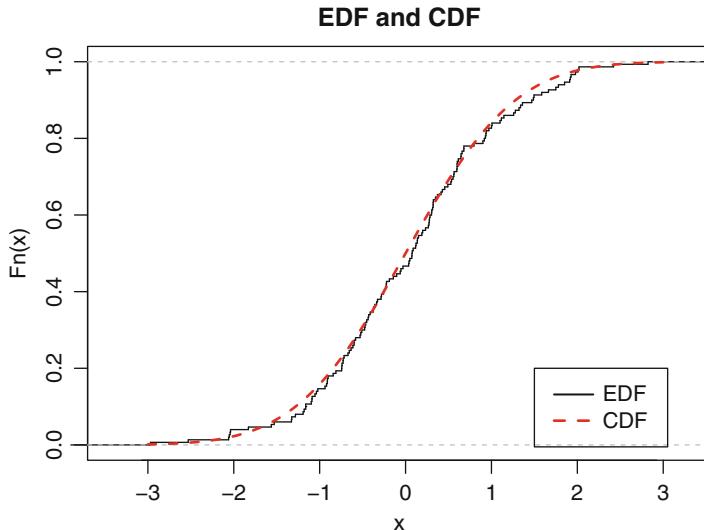


Fig. 4.8. The EDF F_n (solid) and the true CDF (dashed) for a simulated random sample from an $N(0, 1)$ population. The sample size is 150.

The *order statistics* $Y_{(1)}, Y_{(2)}, \dots, Y_{(n)}$ are the values Y_1, \dots, Y_n ordered from smallest to largest. The subscripts of the order statistics are in parentheses to distinguish them from the unordered sample. For example, Y_1 is simply the first observation in the original sample while $Y_{(1)}$ is the smallest observation in that sample. The *sample quantiles* are defined in slightly different ways by different authors, but roughly the q -sample quantile, $0 < q < 1$, is $Y_{(k)}$, where k is qn rounded to an integer. Some authors round up, others round to the nearest integer, and still others interpolate. The function `quantile()` in R has nine different types of sample quantiles, the three used by SASTM, S-PLUSTM, and SPSSTM and MinitabTM, plus six others. With the large sample sizes typical of financial markets data, the different choices lead to nearly identical estimates, but for small samples they can be somewhat different.

The q th quantile is also called the $100q$ th *percentile*. Certain quantiles have special names. The 0.5 sample quantile is the 50th percentile and is usually called the *median*. The 0.25 and 0.75 sample quantiles are called the first and third *quartiles*, and the median is also called the second quartile. The 0.2, 0.4, 0.6, and 0.8 quantiles are the *quintiles* since they divide the data into five equal-size subsets, and the 0.1, 0.2, ..., 0.9 quantiles are the *deciles*.⁴

⁴ Somewhat confusingly, the bottom 10% of the data is also called the first decile and similarly for the second 10%, and so forth. Thus, the first decile could refer to the 10th percentile of the data or to all of the data at or below this percentile. In like fashion, the bottom 20% of the sample is called the first quintile and the second or fifth quantiles are defined analogously.

4.3.1 The Central Limit Theorem for Sample Quantiles

Many estimators have an approximate normal distribution if the sample size is sufficiently large. This is true of sample quantiles by the following central limit theorem.

Result 4.1 Let Y_1, \dots, Y_n be an i.i.d. sample with a CDF F . Suppose that F has a density f that is continuous and positive at $F^{-1}(q)$, $0 < q < 1$. Then for large n , the q th sample quantile is approximately normally distributed with mean equal to the population quantile $F^{-1}(q)$ and variance equal to

$$\frac{q(1-q)}{n [f\{F^{-1}(q)\}]^2}. \quad (4.3)$$

This result is not immediately applicable, for example, for constructing a confidence interval for a population quantile, because $[f\{F^{-1}(q)\}]^2$ is unknown. However, f can be estimated by kernel density estimation (Sect. 4.2) and $F^{-1}(q)$ can be estimated by the q th sample quantile. Alternatively, a confidence interval can be constructed by resampling. Resampling is introduced in Chap. 6.

4.3.2 Normal Probability Plots

Many statistical models assume that a random sample comes from a normal distribution. *Normal probability* plots are used to check this assumption, and, if the normality assumption seems false, to investigate how the distribution of the data differs from a normal distribution. If the normality assumption is true, then the q th sample quantile will be approximately equal to $\mu + \sigma \Phi^{-1}(q)$, which is the population quantile. Therefore, except for sampling variation, a plot of the sample quantiles versus Φ^{-1} will be linear. One version of the normal probability plot is a plot of $Y_{(i)}$ versus $\Phi^{-1}\{(i - 1/2)/n\}$. These are the $(i - 1/2)/n$ sample and population quantiles, respectively. The subtraction of $1/2$ from i in the numerator is used to avoid $\Phi^{-1}(1) = +\infty$ when $i = n$.

Systematic deviation of the plot from a straight line is evidence of non-normality. There are other versions of the normal plot, e.g., a plot of the order statistics versus their expectations under normality, but for large samples these will all be similar, except perhaps in the extreme tails.

Statistical software differs about whether the data are on the x -axis (horizontal axis) and the theoretical quantiles on the y -axis (vertical axis) or vice versa. The `qqnorm()` function in R allows the data to be on either axis depending on the choice of the parameter `datax`. When interpreting a normal plot with a nonlinear pattern, it is essential to know which axis contains the data. In this book, the data will always be plotted on the x -axis and the theoretical quantiles on the y -axis, so in R, `datax = TRUE` was used to construct the plots rather than the default, which is `datax = FALSE`.

If the pattern in a normal plot is nonlinear, then to interpret the pattern one checks where the plot is convex and where it is concave. A convex curve is one such that as one moves from left to right, the slope of the tangent line increases; see Fig. 4.9a. Conversely, if the slope decreases as one moves from left to right, then the curve is concave; see Fig. 4.9b. A convex-concave curve is convex on the left and concave on the right and, similarly, a concave-convex curve is concave on the left and convex on the right; see Fig. 4.9c and d.

A convex, concave, convex-concave, or concave-convex normal plot indicates, respectively, left skewness, right skewness, heavy tails (compared to the normal distribution), or light tails (compared to the normal distribution)—these interpretations require that the sample quantiles are on the horizontal axis and need to be changed if the sample quantiles are plotted on the vertical axis. *Tails* of a distribution are the regions far from the center. Reasonable definitions of the “tails” would be that the left tail is the region from $-\infty$ to $\mu - 2\sigma$ and the right tail is the region from $\mu + 2\sigma$ to $+\infty$, though the choices of $\mu - 2\sigma$ and $\mu + 2\sigma$ are somewhat arbitrary. Here μ and σ are the mean and standard deviation, though they might be replaced by the median and MAD estimator, which are less sensitive to tail weight.

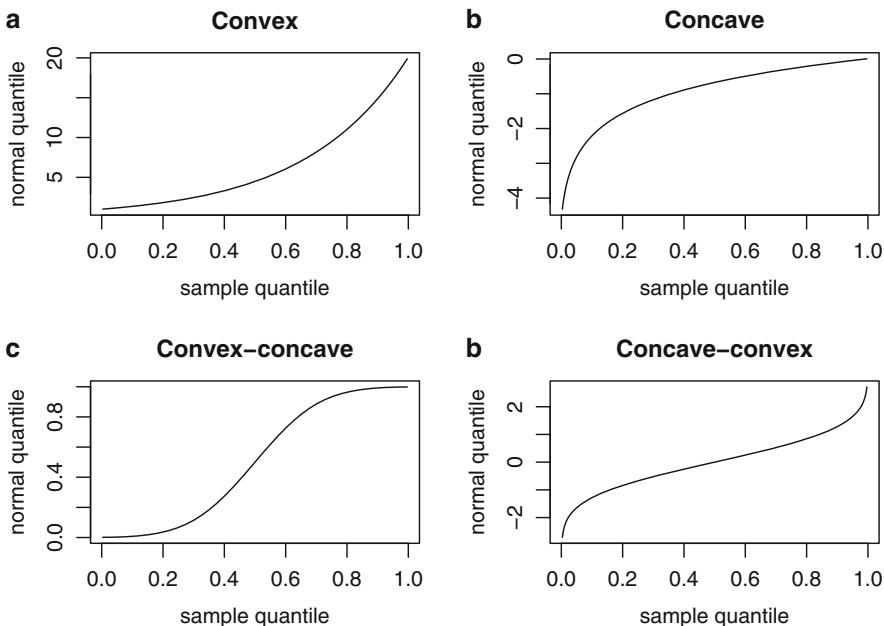


Fig. 4.9. As one moves from (a) to (d), the curves are convex, concave, convex-concave, and concave-convex. Normal plots with these patterns indicate left skewness, right skewness, heavier tails than a normal distribution, and lighter tails than a normal distribution, respectively, assuming that the data are on the x-axis and the normal quantiles on the y-axis, as will always be the case in this textbook.

Figure 4.10 contains normal plots of samples of size 20, 150, and 1000 from a normal distribution. To show the typical amount of random variation in normal plots, two independent samples are shown for each sample size. The plots are only close to linear because of random variation. Even for normally distributed data, some deviation from linearity is to be expected, especially for smaller sample sizes. With larger sample sizes, the only deviations from linearity are in the extreme left and right tails, where the plots are more variable.

Often, a reference line is added to the normal plot to help the viewer determine whether the plot is reasonably linear. One choice for the reference line goes through the pair of first quartiles and the pair of third quartiles; this is what R's `qqline()` function uses. Other possibilities would be a least-squares fit to all of the quantiles or, to avoid the influence of outliers, some subset of the quantiles, e.g., all between the 0.1 and 0.9-quantiles.

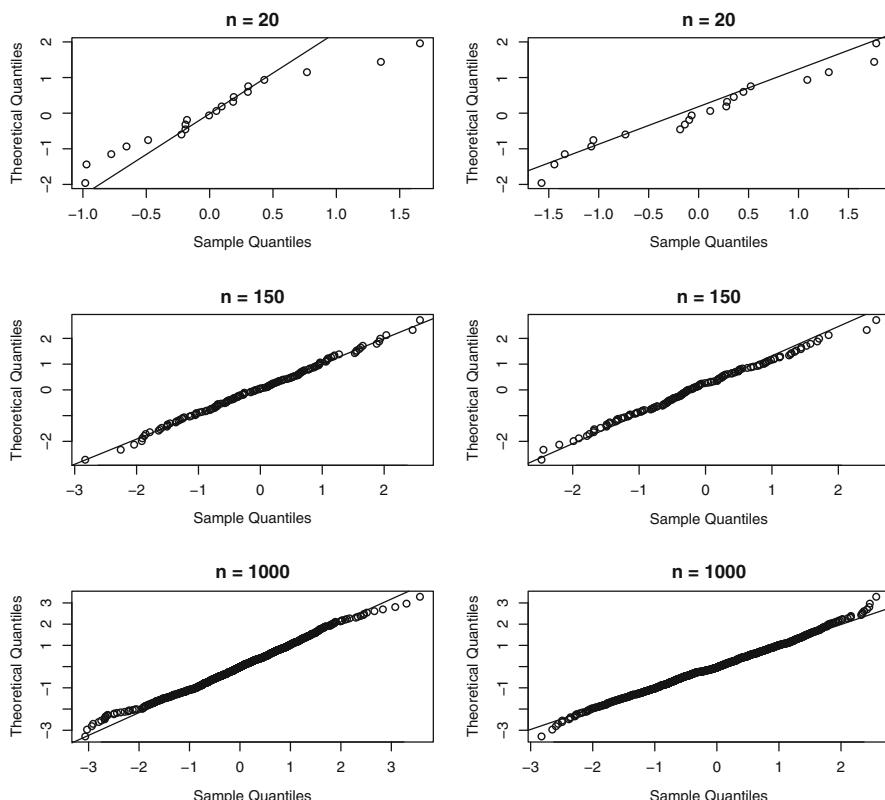


Fig. 4.10. Normal probability plots of random samples of size 20, 150, and 1000 from an $N(0, 1)$ population. The plots were produced by the R function `qqnorm()`. The reference lines pass through the first and third quartiles and were produced by R's `qqline()` function.

Figure 4.11 contains normal probability plots of samples of size 150 from lognormal $(0, \sigma^2)$ distributions,⁵ with the log-standard deviation $\sigma = 1, 1/2$, and $1/5$. The concave shapes in Fig. 4.11 indicate right skewness. The skewness when $\sigma = 1$ is quite strong, and when $\sigma = 1/2$, the skewness is still very noticeable. With σ reduced to $1/5$, the right skewness is much less pronounced and might not be discernable with smaller sample sizes.

Figure 4.12 contains normal plots of samples of size 150 from t -distributions with 4, 10, and 30 degrees of freedom. The first two distributions have heavy tails or, stated differently, are outlier-prone, meaning that the extreme observations on both the left and right sides are significantly more extreme than would be expected for a normal distribution. One can see that the tails are heavier in the sample with 4 degrees of freedom compared to the sample with

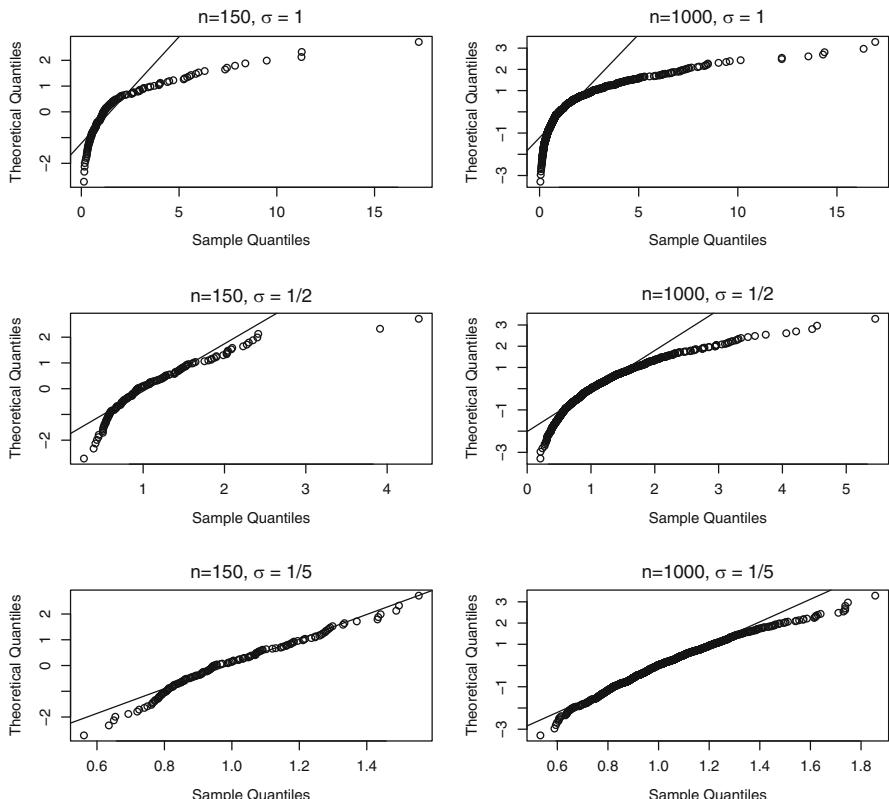


Fig. 4.11. Normal probability plots of random samples of sizes 150 and 1000 from lognormal populations with $\mu = 0$ and $\sigma = 1, 1/2$, or $1/5$. The reference lines pass through the first and third quartiles.

⁵ See Appendix A.9.4 for an introduction to the lognormal distribution and the definition of the log-standard deviation.

10 degrees of freedom, and the tails of the t -distribution with 30 degrees of freedom are not much different from the tails of a normal distribution. It is a general property of the t -distribution that the tails become heavier as the degrees of freedom parameter decreases and the distribution approaches the normal distribution as the degrees of freedom approaches infinity. Any t -distribution is symmetric,⁶ so none of the samples is skewed. Heavy-tailed distributions with little or no skewness are common in finance and, as we will see, the t -distribution is a reasonable model for stock returns and other financial markets data.

Sometimes, a normal plot will not have any of the patterns discussed here but instead will have more complex behavior. An example is shown in Fig. 4.13, which uses a simulated sample from a trimodal density. The alternation of the QQ plot between concavity and convexity indicates complex behavior which should be investigated by a KDE. Here, the KDE reveals the trimodality. Multimodality is somewhat rare in practice and often indicates a mixture of several distinct groups of data.

It is often rather difficult to decide whether a normal plot is close enough to linear to conclude that the data are normally distributed, especially when the sample size is small. For example, even though the plots in Fig. 4.10 are close to linear, there is some nonlinearity. Is this nonlinearity due to nonnormality or just due to random variation? If one did not know that the data were simulated from a normal distribution, then it would be difficult to tell, unless one were very experienced with normal plots. In such situations, a test of normality is very helpful. These tests are discussed in Sect. 4.4.

4.3.3 Half-Normal Plots

The half-normal plot is a variation of the normal plot used for detecting outlying data rather than checking for a normal distribution. For example, suppose one has data Y_1, \dots, Y_n and wants to see whether any of the absolute deviations $|Y_1 - \bar{Y}|, \dots, |Y_n - \bar{Y}|$ from the mean are unusual. In a half-normal plot, these deviation are plotted against the quantiles of $|Z|$, where Z is $N(0, 1)$ distributed. More precisely, a half-normal plot is a scatterplot of the order statistics of the absolute values of the data against $\Phi^{-1}\{(n + i)/(2n + 1)\}$, $i = 1, \dots, n$, where n is the sample size. The function `halfnorm()` in R's `faraway` package creates a half-normal plot and labels the `nlab` most outlying observations, where `nlab` is an argument of this function with a default value of 2.

Example 4.1. DM/dollar exchange rate—Half-normal plot

Figure 4.14 is a half-normal plot of changes in the DM/dollar exchange rate. The plot shows that case #1447 is the most outlying, with case #217

⁶ However, t -distributions have been generalized in at least two different ways to the so-called skewed- t -distributions, which need not be symmetric. See Sect. 5.7.

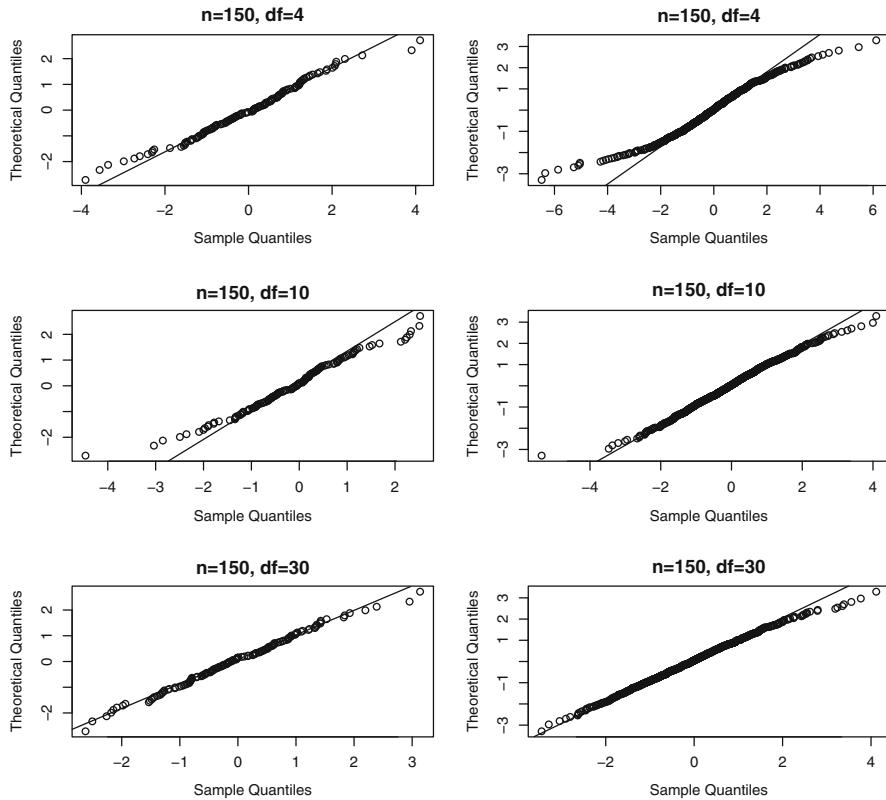


Fig. 4.12. Normal probability plot of a random sample of size 150 and 1000 from a t -distribution with 4, 10, and 30 degrees of freedom. The reference lines pass through the first and third quartiles.

the next most outlying. Only the two most outlying cases are labeled because the default value of `nlab` was used. The code to produce this figure is below.

```

1 data(Garch, package = "Ecdat")
2 diffdm = diff(dm) # Deutsch mar
3 pdf("dm_halfnormal.pdf" ,width = 7, height = 6) # Figure 4.14
4 halfnorm(abs(difffm), main = "changes in DM/dollar exchange rate",
5   ylab = "Sorted data")
6 graphics.off()

```

□

Another application of half-normal plotting can be found in Sect. 10.1.3.

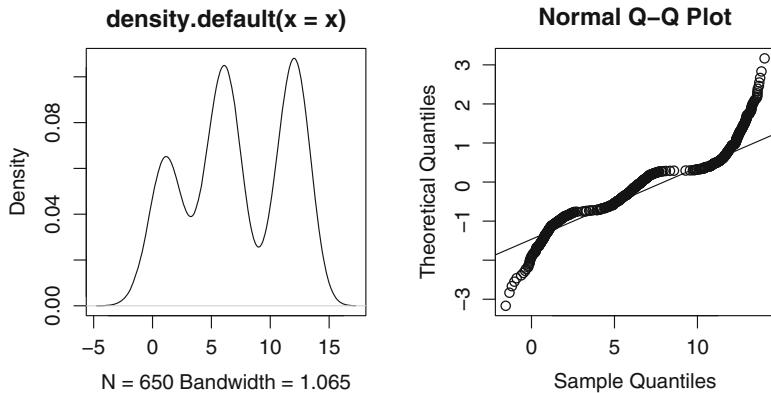


Fig. 4.13. Kernel density estimate (left) and normal plot (right) of a simulated sample from a trimodal density. The reference lines pass through the first and third quartiles. Because of the three modes, the normal plot changes convexity three times, concave to convex to concave to convex, going from left to right.

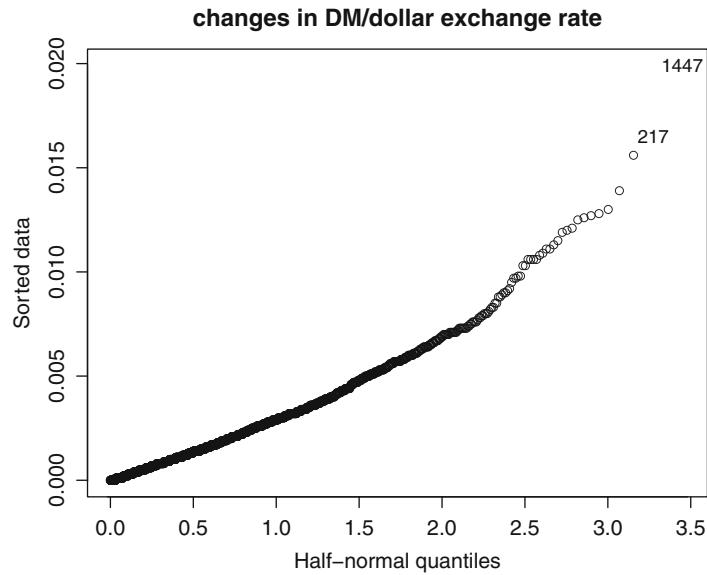


Fig. 4.14. Half-normal plot of changes in DM/dollar exchange rate.

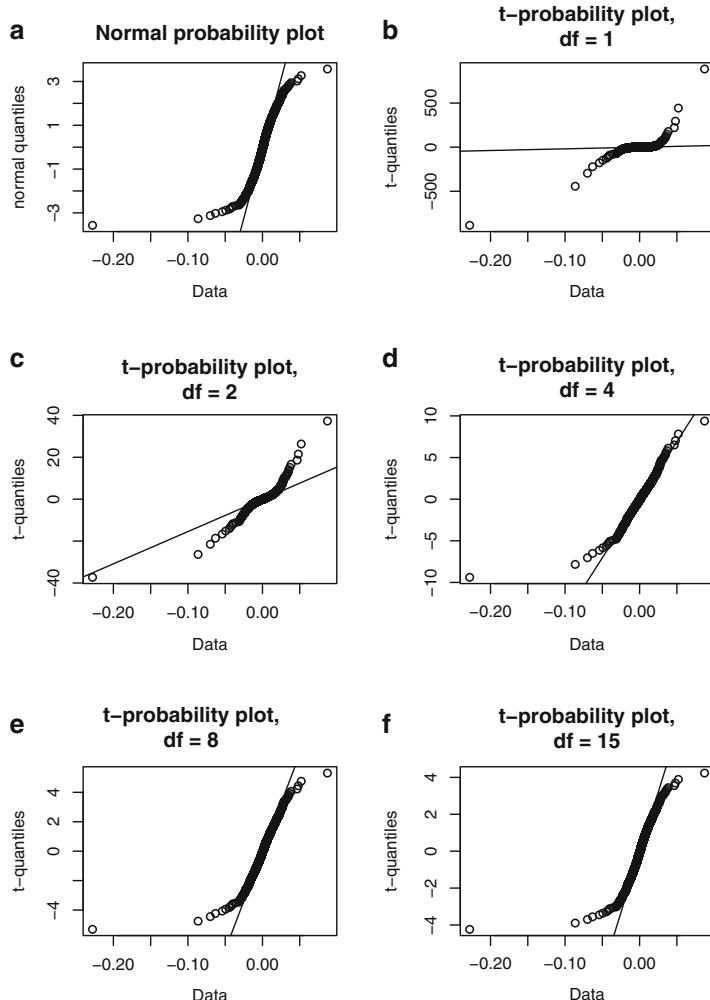


Fig. 4.15. Normal and t probability plots of the daily returns on the S&P 500 index from January 1981 to April 1991. This data set is the `SP500` series in the `Ecdat` package in R. The reference lines pass through the first and third quartiles.

4.3.4 Quantile–Quantile Plots

Normal probability plots are special cases of *quantile-quantile plots*, also known as *QQ plots*. A *QQ plot* is a plot of the quantiles of one sample or distribution against the quantiles of a second sample or distribution.

For example, suppose that we wish to model a sample using the $t_\nu(\mu, \sigma^2)$ distribution defined in Sect. 5.5.2. The parameter ν is called the “degrees of freedom,” or simply “df.” Suppose, initially, that we have a hypothesized value of ν , say $\nu = 6$ to be concrete. Then we plot the sample quantiles

against the quantiles of the $t_6(0, 1)$ distribution. If the data are from a $t_6(\mu, \sigma^2)$ distribution, then, apart from random variation, the plot will be linear with intercept and slope depending on μ and σ .

Figure 4.15 contains a normal plot of the S&P 500 log returns in panel (a) and t -plots with 1, 2, 4, 8, and 15 df in panels (b) through (f). None of the plots looks exactly linear, but the t -plot with 4 df is rather straight through the bulk of the data. There are approximately nine returns in the left tail and four in the right tail that deviate from a line through the remaining data, but these are small numbers compared to the sample size of 2783. Nonetheless, it is worthwhile to keep in mind that the historical data have more extreme outliers than a t -distribution. The t -model with 4 df and mean and standard deviation estimated by maximum likelihood⁷ implies that a daily log return of -0.228 , the return on Black Monday, or less has probability 3.2×10^{-6} . This means approximately 3 such returns every 1,000,000 days or 40,000 years, assuming 250 trading days per year. Thus, the t -model implies that Black Monday was extremely unlikely, and anyone using that model should be mindful that it did happen.

There are two reasons why the t -model does not give a credible probability of a negative return as extreme as on Black Monday. First, the t -model is symmetric, but the return distribution appears to have some skewness in the extreme left tail, which makes extreme negative returns more likely than under the t -model. Second, the t -model assumes constant conditional volatility, but volatility was unusually high in October 1987. GARCH models (Chap. 14) can accommodate this type of volatility clustering and provide more realistic estimates of the probability of an extreme event such as Black Monday.

Quantile–quantile plots are useful not only for comparing a sample with a theoretical model, as above, but also for comparing two samples. If the two samples have the same sizes, then one need only plot their order statistics against each other. Otherwise, one computes the same sets of sample quantiles for each and plots them. This is done automatically with the R command `qqplot()`.

The interpretation of convex, concave, convex-concave, and concave-convex QQ plots is similar to that with QQ plots of theoretical quantiles versus sample quantiles. A concave plot implies that the sample on the x -axis is more right-skewed, or less left-skewed, than the sample on the y -axis. A convex plot implies that the sample on the x -axis is less right-skewed, or more left-skewed, than the sample on the y -axis. A convex-concave (concave-convex) plot implies that the sample on the x -axis is more (less) heavy-tailed than the sample on the y -axis. As before, a straight line, e.g., through the first and third quartiles, is often added for reference.

Figure 4.16 contains sample QQ plots for all three pairs of the three time series, S&P 500 returns, changes in the DM/dollar rate, and changes in the risk-free return, used as examples in this chapter. One sees that the S&P 500

⁷ See Sect. 5.14.

returns have more extreme outliers than the other two series. The changes in DM/dollar and risk-free returns have somewhat similar shapes, but the changes in the risk-free rate have slightly more extreme outliers in the left tail. To avoid any possible confusion, it should be mentioned that the plots in Fig. 4.16 only compare the marginal distributions of the three time series. They tell us nothing about dependencies between the series and, in fact, the three series were observed on different time intervals so correlations between these time series cannot be estimated from these data.

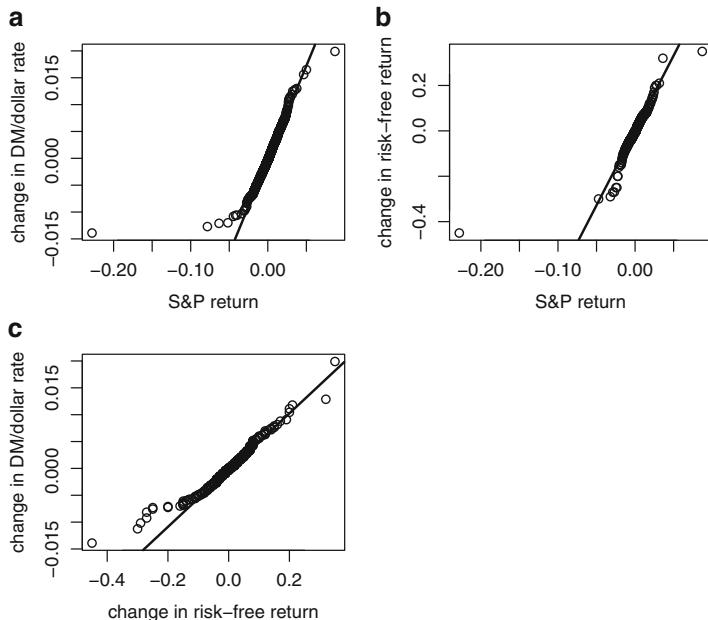


Fig. 4.16. Sample QQ plots. The straight lines pass through the first and third sample quantiles. (a) Change in DM/dollar rate versus S&P return. (b) Change in risk-free rate versus S&P return. (c) Change in DM/dollar rate versus change in risk-free rate.

The code for panel (a) of Fig. 4.16 is below. The code for the other panels is similar and so is omitted.

```

1 qqplot(SPreturn, diffdm, xlab = "S&P return",
2       ylab = "change in DM/dollar rate", main = "(a)")
3 xx = quantile(SPreturn, c(0.25, 0.75))
4 yy = quantile(diffdm, c(0.25, 0.75))
5 slope = (yy[2] - yy[1]) / (xx[2] - xx[1])
6 inter = yy[1] - slope*xx[1]
7 abline(inter, slope, lwd = 2 )

```

4.4 Tests of Normality

When viewing a normal probability plot, it is often difficult to judge whether any deviation from linearity is systematic or instead merely due to sampling variation, so a statistical test of normality is useful. The null hypothesis is that the sample comes from a normal distribution and the alternative is that the sample is from a nonnormal distribution.

The Shapiro–Wilk test of these hypotheses uses something similar to a normal plot. Specifically, the Shapiro–Wilk test is based on the association between sample order statistics $Y_{(i)}$ and the expected normal order statistics which, for large samples, are close to $\Phi^{-1}\{i/(n + 1)\}$, the quantiles of the standard normal distribution. The vector of expected order statistics is multiplied by the inverse of its covariance matrix. Then the correlation between this product and the sample order statistics is used as the test statistic. Correlation and covariance matrices will be discussed in greater detail in Chap. 7. For now, only a few facts will be mentioned. The *covariance* between two random variables X and Y is

$$\text{Cov}(X, Y) = \sigma_{XY} = E\left[\{X - E(X)\}\{Y - E(Y)\}\right],$$

and the *Pearson correlation coefficient* between X and Y is

$$\text{Corr}(X, Y) = \rho_{XY} = \sigma_{XY}/\sigma_X \sigma_Y. \quad (4.4)$$

A correlation equal to 1 indicates a perfect positive linear relationship, where $Y = \beta_0 + \beta_1 X$ with $\beta_1 > 0$. Under normality, the correlation between sample order statistics and the expected normal order statistics should be close to 1 and the null hypothesis of normality is rejected for small values of the correlation coefficient. In R, the Shapiro–Wilk test can be implemented using the `shapiro.test()` function.

The Jarque–Bera test uses the sample skewness and kurtosis coefficients and is discussed in Sect 5.4 where skewness and kurtosis are introduced. Other tests of normality in common use are the Anderson–Darling, Cramér–von Mises, and Kolmogorov–Smirnov tests. These tests compare the sample CDF to the normal CDF with mean equal to \bar{Y} and variance equal to s_Y^2 . The Kolmogorov–Smirnov test statistic is the maximum absolute difference between these two functions, while the Anderson–Darling and Cramér–von Mises tests are based on a weighted integral of the squared difference. The p -values of the Shapiro–Wilk, Anderson–Darling, Cramér–von Mises, and Kolmogorov–Smirnov tests are routinely part of the output of statistical software. A small p -value is interpreted as evidence that the sample is not from a normal distribution.

A recent comparison of eight tests of normality (Yap and Sim 2011) found that the Shapiro–Wilk test was as powerful as its competitors for both short- and long-tailed symmetric alternatives and was the most powerful

test for asymmetric alternatives. The tests in this study were: Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors, Cramér-vo Mises, Anderson-Darling, D'Agostino-Pearson, Jarque-Bera, and chi-squared.

For the S&P 500 returns, the Shapiro-Wilk test rejects the null hypothesis of normality with a p -value less than 2.2×10^{-16} . The Shapiro-Wilk also strongly rejects normality for the changes in DM/dollar rate and for the changes in risk-free return. With large sample sizes, e.g., 2783, 1866, and 515, for the S&P 500 returns, changes in DM/dollar rate, and changes in risk-free return, respectively, it is quite likely that normality will be rejected, since any real population will deviate to some extent from normality and any deviation, no matter how small, will be detected with a large enough sample. When the sample size is large, it is important to look at normal plots to see whether the deviation from normality is of practical importance. For financial time series, the deviation from normality in the tails is often large enough to be important.⁸

4.5 Boxplots

The boxplot is a useful graphical tool for comparing several samples. The appearance of a boxplot depends somewhat on the specific software used. In this section, we will describe boxplots produced by the R function `boxplot()`. The three boxplots in Fig. 4.17 were created by `boxplot()` with default choice of tuning parameters. The “box” in the middle of each plot extends from the first to the third quartile and thus gives the range of the middle half of the data, often called the *interquartile range*, or IQR. The line in the middle of the box is at the median. The “whiskers” are the vertical dashed lines extending from the top and bottom of each box. The whiskers extend to the smallest and largest data points whose distance from the bottom or top of the box is at most 1.5 times the IQR.⁹ The ends of the whiskers are indicated by horizontal lines. All observations beyond the whiskers are plotted with an “o”. The most obvious differences among the three boxplots in Fig. 4.17 are differences in scale, with the monthly risk-free return changes being the most variable and the daily DM/dollar changes being the least variable. It is not surprising that the changes in the risk-free return are most variable, since these are changes over months, not days as with the other series.

These scale differences obscure differences in shape. To remedy this problem, in Fig. 4.18 the three series have been standardized by subtracting the median and then dividing by the MAD. Now, differences in shape are clearer. One can see that the S&P 500 returns have heavier tails because the “o’s are farther from the whiskers. The return of the S&P 500 on Black Monday

⁸ See Chap. 19 for a discussion on how tail weight can greatly affect risk measures such as VaR and expected shortfall.

⁹ The factor 1.5 is the default value of the `range` parameter and can be changed.

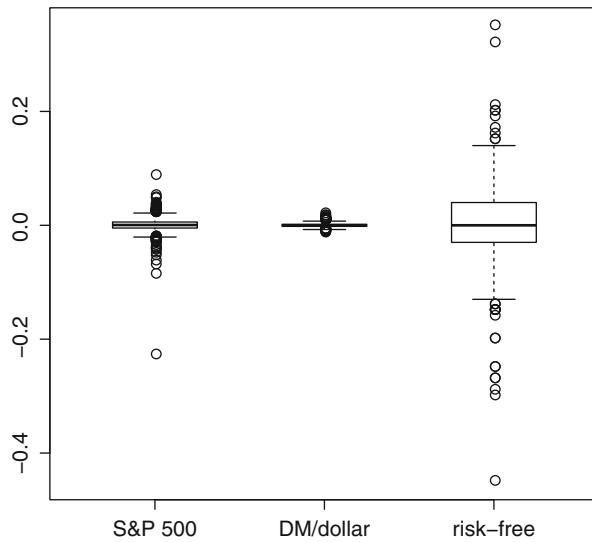


Fig. 4.17. Boxplots of the S&P 500 daily log returns, daily changes in the DM/dollar exchange rate, and monthly changes in the risk-free returns.

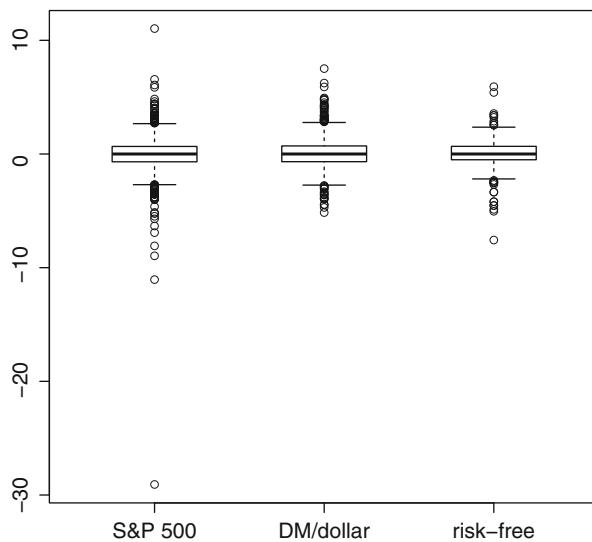


Fig. 4.18. Boxplots of the standardized S&P 500 daily log returns, daily changes in the DM/dollar exchange rate, and monthly changes in the risk-free returns.

is quite detached from the remaining data. Of course, one should be aware of differences in scale, so it is worthwhile to look at boxplots of the variables both without and with standardization.

When comparing several samples, boxplots and QQ plots provide different views of the data. It is best to use both. However, if there are N samples, then the number of QQ plots is $N(N - 1)/2$ or $N(N - 1)$ if, by interchanging axes, one includes two plots for each pair of samples. This number can get out of hand quickly, so, for large values of N , one might use boxplots augmented with a few selected QQ plots.

4.6 Data Transformation

There are a number of reasons why data analysts often work not with the original variables, but rather with transformations of the variables such as logs, square roots, or other power transformations. Many statistical methods work best when the data are normally distributed or at least symmetrically distributed and have a constant variance, and the transformed data will often exhibit less skewness and a more constant variance compared to the original variables, especially if the transformation is selected to induce these features.

A transformation is called *variance stabilizing* if it removes a dependence between the conditional variance and the conditional mean of a variable. For example, if Y is Poisson distributed with a conditional mean depending on X , then its conditional variance is equal to the conditional mean. A transformation h would be variance-stabilizing for Y if the conditional variance of $h(Y)$ did not depend on the conditional mean of $h(Y)$.

The logarithm transformation is probably the most widely used transformation in data analysis, though the square root is a close second. The log stabilizes the variance of a variable whose conditional standard deviation is proportional to its conditional mean. This is illustrated in Fig. 4.19, which plots monthly changes in the risk-free return (top row) and changes in the log of the return (bottom row) against the lagged risk-free return (left column) or year (right column). Notice that the changes in the return are more variable when the lagged return is higher. This behavior is called nonconstant conditional variance or conditional heteroskedasticity. We see in the bottom row that the changes in the log return have relatively constant variability, at least compared to changes in the return.

The log transformation is sometimes embedded into the power transformation family by using the so-called Box–Cox power transformation

$$y^{(\alpha)} = \begin{cases} \frac{y^\alpha - 1}{\alpha}, & \alpha \neq 0 \\ \log(y), & \alpha = 0. \end{cases} \quad (4.5)$$

In (4.5), the subtraction of 1 from y^α and the division by α are not essential, but they make the transformation continuous in α at 0 since

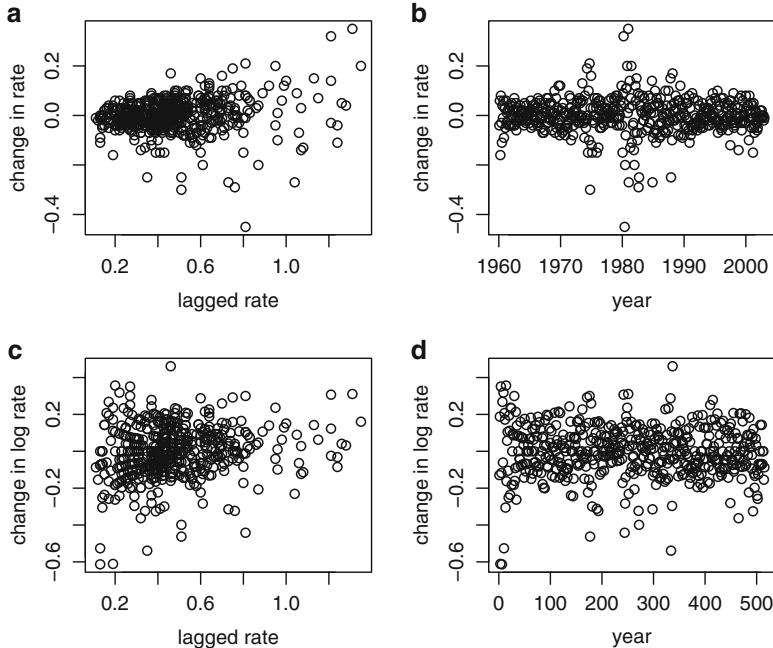


Fig. 4.19. Changes in risk-free rate (top) and changes in the logarithm of the risk-free rate (bottom) plotted against time and against lagged rate. The risk-free returns are the variable `rf` of the `Capm` data set in R's `Ecdat` package. (a) Change in risk-free rate versus change in lagged rate. (b) Change in rate versus year. (c) Change in $\log(\text{rate})$ versus lagged rate. (d) Change in $\log(\text{rate})$ versus year.

$$\lim_{\alpha \rightarrow 0} \frac{y^\alpha - 1}{\alpha} = \log(y).$$

Note that division by α ensures that the transformation is increasing even when $\alpha < 0$. This is convenient though not essential. For the purposes of inducing symmetry and a constant variance, y^α and $y^{(\alpha)}$ work equally well and can be used interchangeably, especially if, when $\alpha < 0$, y^α replaced by $-y^\alpha$ to ensure that the transformation is monotonically increasing for all values of α . The use of a monotonically decreasing, rather than increasing, transformation is inconvenient since decreasing transformations reverse ordering and, for example, transform the p th quantile to the $(1-p)$ th quantile.

It is commonly the case that the response is right-skewed and the conditional response variance is an increasing function of the conditional response mean. In such cases, a concave transformation, e.g., a Box–Cox transformation with $\alpha < 1$, will remove skewness and stabilize the variance. If a Box–Cox transformation with $\alpha < 1$ is used, then the smaller the value of α , the greater the effect of the transformation. One can go too far—if the transformed response is left-skewed or has a conditional variance that is decreasing as a function of the conditional mean, then α has been chosen too small. Instances of this type of overtransformation are given in Examples 4.2, 4.4, and 13.2.

Typically, the value of α that is best for symmetrizing the data is not the same value of α that is best for stabilizing the variance. Then, a compromise is needed so that the transformation is somewhat too weak for one purpose and somewhat too strong for the other. Often, however, the compromise is not severe, and near symmetry and homoskedasticity can both be achieved.

Example 4.2. Gas flows in pipelines

In this example, we will use a data set of daily flows of natural gas in three pipelines. These data are part of a larger data set used in an investigation of the relationships between flows in the pipelines and prices. Figure 4.20 contains histograms of the daily flows. Notice that all three distributions are left-skewed. For left-skewed data, a Box–Cox transformation should use $\alpha > 1$.

Figure 4.21 shows KDEs of the flows in pipeline 1 after a Box–Cox transformation using $\alpha = 1, 2, 3, 4, 5, 6$. One sees that α between 3 and 4 removes most

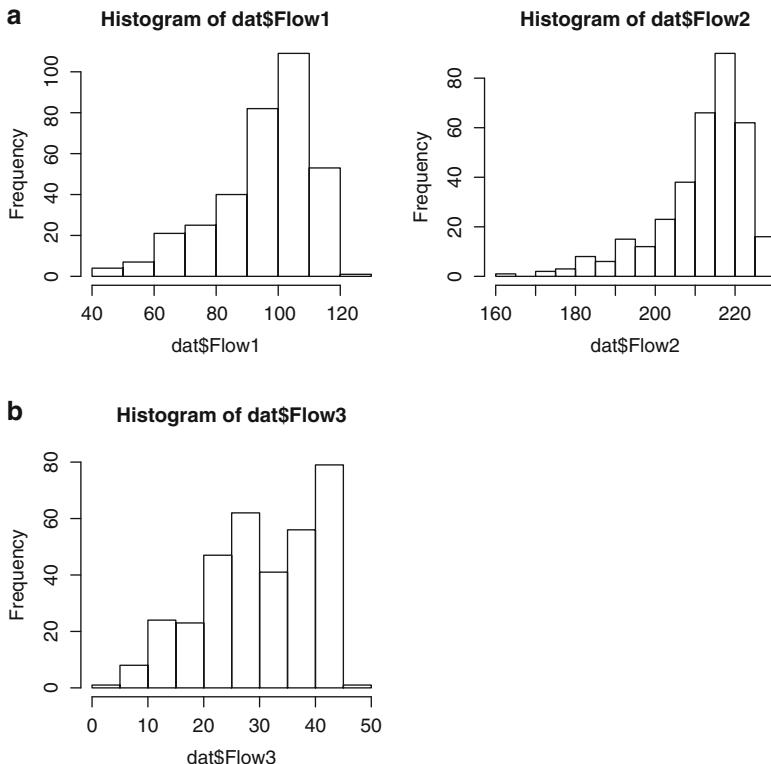


Fig. 4.20. Histograms of daily flows in three pipelines.

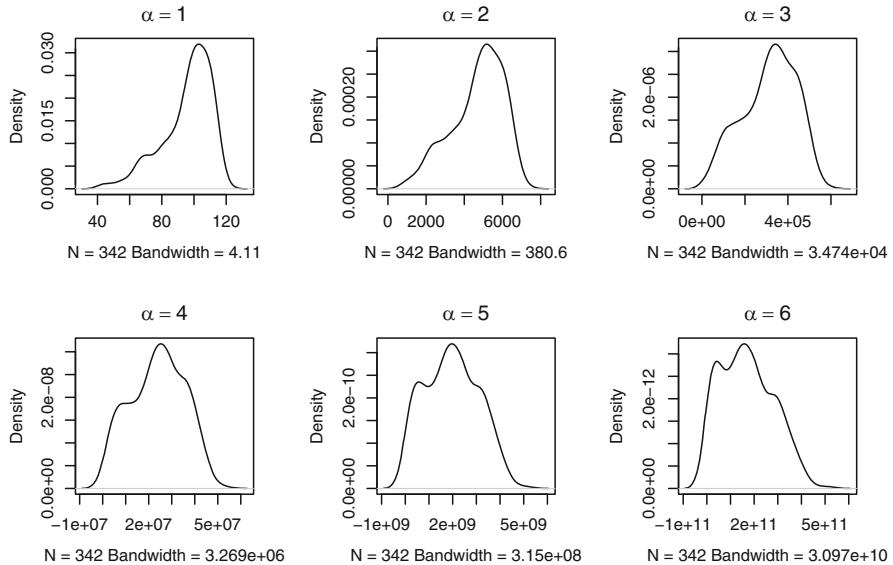


Fig. 4.21. Kernel density estimates for gas flows in pipeline 1 with Box–Cox transformations.

of the left-skewness and $\alpha = 5$ or greater overtransforms to right-skewness. Later, in Example 5.7, we will illustrate an automatic method for selecting α and find that $\alpha = 3.5$ is chosen. \square

Example 4.3. *t*-Tests and transformations

This example shows the deleterious effect of skewness and nonconstant variance on hypothesis testing and how a proper data transformation can remedy this problem. The boxplots on the panel (a) in Fig. 4.22 are of independent samples of size 15 from $\text{lognormal}(1,4)$ (left) and $\text{lognormal}(3,4)$ distributions. Panel (b) shows boxplots of the log-transformed data.

Suppose one wants to test the null hypothesis that the two populations have the same means against a two-sided alternative. The transformed data satisfy the assumptions of the *t*-test that the two populations are normally distributed with the same variance, but of course the original data do not meet these assumptions. Two-sided independent-samples *t*-tests have *p*-values of 0.105 and 0.00467 using the original data and the log-transformed data, respectively. These two *p*-values lead to rather different conclusions, for the first test that the means are not significantly different at the usual $\alpha = 0.05$, and not quite significant even at $\alpha = 0.1$, and for the second test that the difference is highly significant. The first test reaches an incorrect conclusion because its assumptions are not met. \square

The previous example illustrates some general principles to keep in mind. All statistical estimators and tests make certain assumptions about the distribution of the data. One should check these assumptions, and graphical methods are often the most convenient way to diagnose problems. If the assumptions are not met, then one needs to know how sensitive the estimator or test is to violations of the assumptions. If the estimator or test is likely to be seriously degraded by violations of the assumptions, which is called *nonrobustness*, then there are two recourses. The first is to find a new estimator or test that is suitable for the data. The second is to transform the data so that the transformed data satisfy the assumptions of the original test or estimator.

4.7 The Geometry of Transformations

Response transformations induce normality of a distribution and stabilize variances because they can stretch apart data in one region and push observations together in other regions. Figure 4.23 illustrates this behavior. On the horizontal axis is a sample of data from a right-skewed lognormal distribution. The transformation $h(y)$ is the logarithm. The transformed data are plotted on the vertical axis. The dashed lines show the transformation of y to $h(y)$ as one moves from a y -value on the x -axis upward to the curve and then to $h(y)$ on the y -axis. Notice the near symmetry of the transformed data. This symmetry is achieved because the log transformation stretches apart data with small values and shrinks together data with large values. This can be seen by observing the derivative of the log function. The derivative of $\log(y)$ is $1/y$, which is a decreasing function of y . The derivative is, of course, the slope of the tangent line and the tangent lines at $y = 1$ and $y = 5$ are plotted to show the decrease in the derivative as y increases.

Consider an arbitrary increasing transformation, $h(y)$. If x and x' are two nearby data points that are transformed to $h(x)$ and $h(x')$, respectively, then

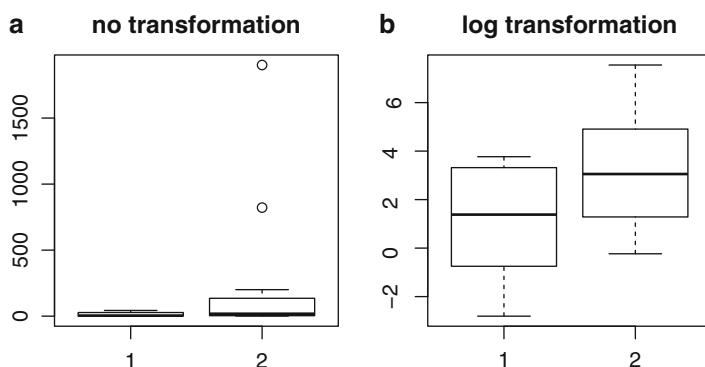


Fig. 4.22. Boxplots of samples from two lognormal distributions without (a) and with (b) log transformation.

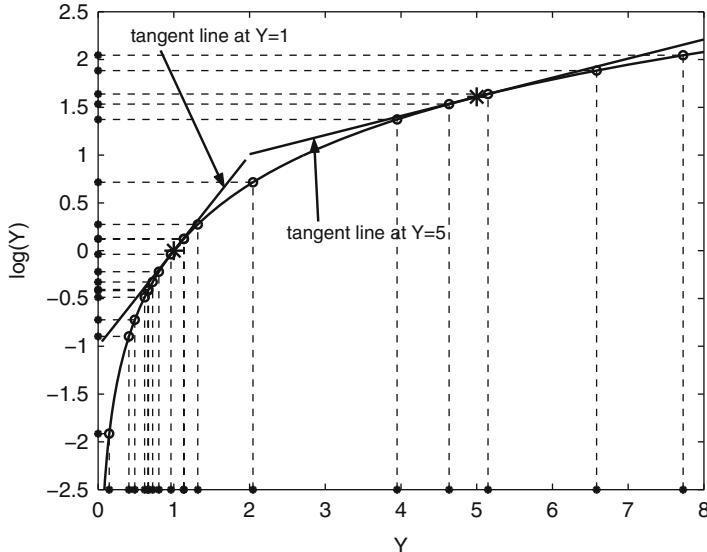


Fig. 4.23. A symmetrizing transformation. The skewed lognormal data on the horizontal axis are transformed to symmetry by the log transformation.

the distance between transformed values is $|h(x) - h(x')| \approx h^{(1)}(x)|x - x'|$. Therefore, $h(x)$ and $h(x')$ are stretched apart where $h^{(1)}$ is large and pushed together where $h^{(1)}$ is small. A function h is called concave if $h^{(1)}(y)$ is a decreasing function of y . As can be seen in Fig. 4.23, concave transformations can remove right skewness.

Concave transformations can also stabilize the variance when the untransformed data are such that small observations are less variable than large observations. This is illustrated in Fig. 4.24. There are two groups of responses, one with a mean of 1 and a relatively small variance and another with a mean of 5 and a relatively large variance. If the expected value of the response Y_i , conditional on \mathbf{X}_i , followed a regression model $m(\mathbf{X}_i; \boldsymbol{\beta})$, then two groups like these would occur if there were two possible values of \mathbf{X}_i , one with a small value of $m(\mathbf{X}_i; \boldsymbol{\beta})$ and the other with a large value. Because of the concavity of the transformation h , the variance of the group with a mean of 5 is reduced by transformation. After the transformation, the groups have nearly the same variance, as can be seen by observing the scatter of the two groups on the y -axis.

The strength of a transformation can be measured by how much its derivative changes over some interval, say a to b . More precisely, for $a < b$, the strength of an increasing transformation h is the derivative ratio $h'(b)/h'(a)$. If the transformation is concave, then the derivative ratio is less than 1 and the smaller the ratio the stronger the concavity. Conversely, if the transformation is convex, then the derivative ratio is greater than 1 and the larger the ratio,

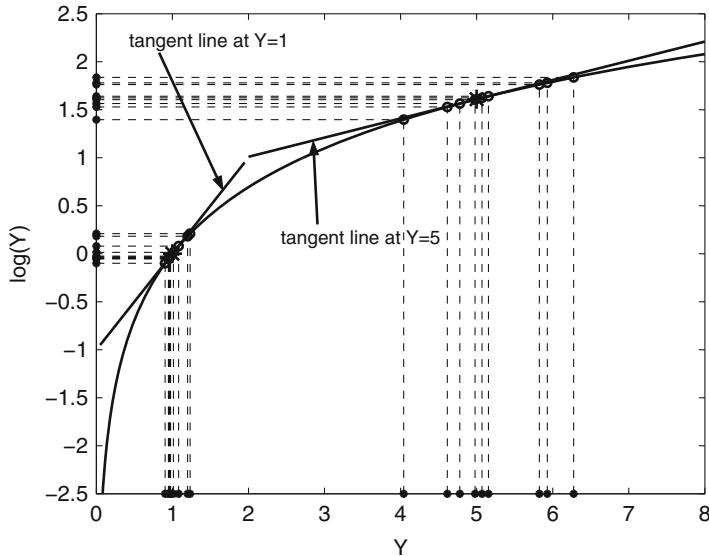


Fig. 4.24. A variance-stabilizing transformation.

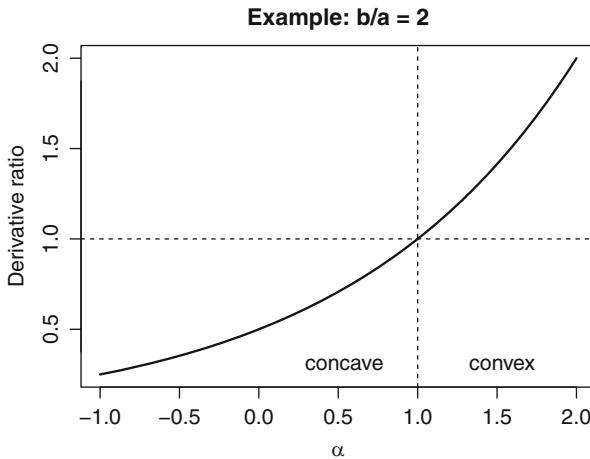


Fig. 4.25. Derivative ratio for Box–Cox transformations.

the greater the convexity. For a Box–Cox transformation, the derivative ratio is $(b/a)^{\alpha-1}$ and so depends on a and b only through the ratio b/a . Figure 4.25 shows the derivative ratio of Box–Cox transformations when $b/a = 2$. One can see that the Box–Cox transformation is concave when $\alpha < 1$, with the concavity becoming stronger as α decreases. Similarly, the transformation is convex for $\alpha > 1$, with increasing convexity as α increases.

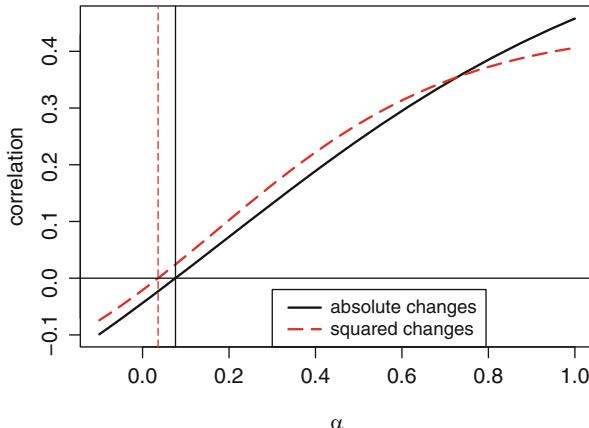


Fig. 4.26. Correlations between the lagged risk-free returns and absolute (solid) and squared (dashed) changes in the Box–Cox transformed returns. A zero correlation indicates a constant conditional variance. Zero correlations are achieved with the transformation parameter α equal to 0.036 and 0.076 for the absolute and squared changes, respectively, as indicated by the vertical lines. If $\alpha \approx 0$, then the data are conditionally homoskedastic, or at least nearly so.

Example 4.4. Risk-free returns—Strength of the Box–Cox transformation for variance stabilization

In this example, we return to the changes in the risk-free interest rates. In Fig. 4.19, it was seen that there is noticeable conditional heteroskedasticity in the changes in the untransformed rate but little or no heteroskedasticity in the changes in the logarithms of the rate. We will see that for a Box–Cox transformation intermediate in strength between the identity transformation ($\alpha = 1$) and the log transformation ($\alpha = 0$), some but not all of the heteroskedasticity is removed, and that a transformation with $\alpha < 0$ is too strong for this application so that a new type of heteroskedasticity is induced.

The strength of a Box–Cox transformation for this example is illustrated in Fig. 4.26. In that figure, the correlations between the lagged risk-free interest returns, r_{t-1} , and absolute and squared changes, $|r_t^{(\alpha)} - r_{t-1}^{(\alpha)}|$ and $\{r_t^{(\alpha)} - r_{t-1}^{(\alpha)}\}^2$, in the transformed rate are plotted against α . The two correlations are similar, especially when they are near zero. Any deviations of the correlations from zero indicate conditional heteroskedasticity where the standard deviation of the change in the transformed rate depends on the previous value of the rate. We see that the correlations decrease as α decreases from 1 so that the concavity of the transformation increases. The correlations are equal to zero when α is very close to 0, that is, the log transformation. If α is much below 0, then the transformation is too strong and the overtransformation induces a negative correlation, which indicates that the conditional standard deviation is a decreasing function of the lagged rate. \square

4.8 Transformation Kernel Density Estimation

The kernel density estimator (KDE) discussed in Sect. 4.2 is popular because of its simplicity and because it is available on most software platforms. However, the KDE has some drawbacks. One disadvantage of the KDE is that it undersmooths densities with long tails. For example, the solid curve in Fig. 4.27 is a KDE of annual earnings in 1988–1989 for 1109 individuals. The data are in the `Earnings` data set in R’s `Ecdat` package. The long right tail of the density estimate exhibits bumps, which seem due solely to random variation in the data, not to bumps in the true density. The problem is that there is no single bandwidth that works well both in the center of the data and in the right tail. The automatic bandwidth selector chose a bandwidth that is a compromise, undersmoothing in the tails and perhaps oversmoothing in the center. The latter problem can cause the height of the density at the mode(s) to be underestimated.

A better density estimate can be obtained by the *transformation kernel density estimator* (TKDE). The idea is to transform the data so that the density of the transformed data is easier to estimate by the KDE. For the earnings data, the square roots of the earnings are closer to being symmetric and have a shorter right tail than the original data; see Fig. 4.28, which compares histograms of the original data and the data transformed by the square root. The KDE should work well for the square roots of the earnings.

Of course, we are interested in the density of the earnings, not the density of their square roots. However, it is easy to convert an estimate of the latter to

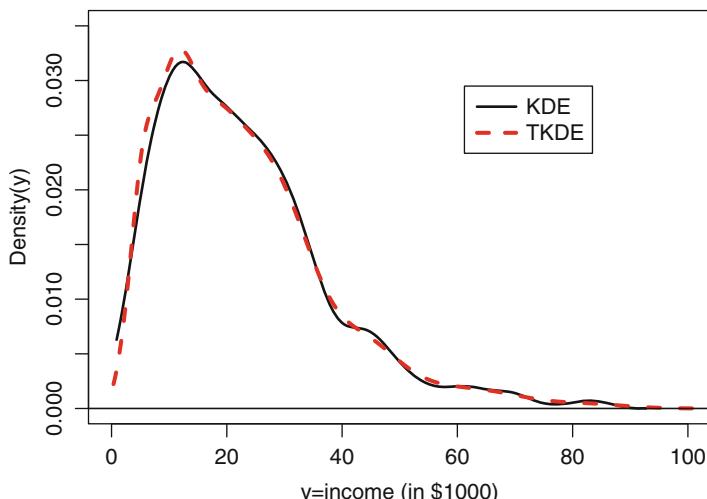


Fig. 4.27. Kernel density and transformation kernel density estimates of annual earnings in 1988–1989 expressed in thousands of 1982 dollars. These data are the same as in Fig. 4.28.

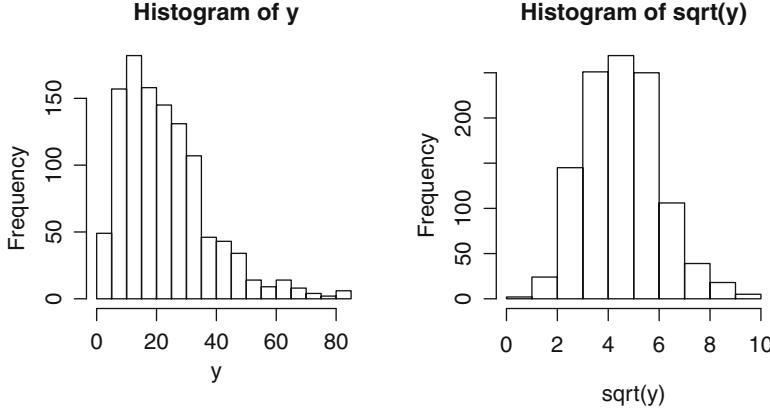


Fig. 4.28. Histograms of earnings (y) and the square roots of earnings. The data are from the `Earnings` data set in R's `Ecdat` package and use only age group `g1`.

one of the former. To do that, one uses the change-of-variables formula (A.4). For convenience, we repeat the result here—if $X = g(Y)$, where g is monotonic and f_X and f_Y are the densities of X and Y , respectively, then

$$f_Y(y) = f_X\{g(y)\} |g'(y)|. \quad (4.6)$$

For example, if $x = g(y) = \sqrt{y}$, then $g'(y) = y^{-1/2}/2$ and

$$f_Y(y) = \{f_X(\sqrt{y})y^{-1/2}\}/2.$$

Putting $y = g^{-1}(x)$ into Eq. (4.6), we obtain

$$f_Y\{g^{-1}(x)\} = f_X(x) |g'\{g^{-1}(x)\}|. \quad (4.7)$$

Equation (4.7) suggests a convenient method for computing the TKDE:

1. start with data Y_1, \dots, Y_n ;
2. transform the data to $X_1 = g(Y_1), \dots, X_n = g(Y_n)$;
3. let \hat{f}_X be the usual KDE calculated on a grid x_1, \dots, x_m using X_1, \dots, X_n ;
4. plot the pairs $[g^{-1}(x_j), \hat{f}_X(x_j) |g'\{g^{-1}(x_j)\}|]$, $j = 1, \dots, m$.

The red dashed curve in Fig. 4.27 is a plot of the TKDE of the earnings data using the square-root transformation. Notice the smoother right tail, the faster decrease to 0 at the left boundary, and the somewhat sharper peak at the mode compared to the KDE (solid curve).

When using a TKDE, it is important to choose a good transformation. For positive, right-skewed variables such as the earnings data, a concave transformation is needed. A power transformation, y^α , for some $\alpha < 1$ is a common choice. Although there are automatic methods for choosing α (see Sect. 4.9), trial-and-error is often good enough.

4.9 Bibliographic Notes

Exploratory data analysis was popularized by Tukey (1977). Hoaglin, Mosteller, and Tukey (1983,1985) are collections of early articles on exploratory data analysis, data transformations, and robust estimation. Kleiber and Zeileis (2008) is an introduction to econometric modeling with R and covers exploratory data analysis as well as material in latter chapters of this book including regression and time series analysis. The R package **AER** accompanies Kleiber and Zeileis's book.

The central limit theorem for sample quantiles is stated precisely and proved in textbooks on asymptotic theory such as Serfling (1980); Lehmann (1999), and van der Vaart (1998).

Silverman (1986) is an early book on nonparametric density estimation and is still well worth reading. Scott (1992) covers both univariate and multivariate density estimation. Wand and Jones (1995) has an excellent treatment of kernel density estimation as well as nonparametric regression, which we cover in Chap. 21. Wand and Jones cover more recent developments such as transformation kernel density estimation. An alternative to the TKDE is variable-bandwidth KDE; see Sect. 2.10 of Wand and Jones (1995) as well as Abramson (1982) and Jones (1990).

Atkinson (1985) and Carroll and Ruppert (1988) are good sources of information about data transformations.

Wand, Marron, and Ruppert (1991) is an introduction to the TKDE and discusses methods for automatic selection of the transformation to minimize the expected squared error of the estimator. Applications of TKDE to losses can be found in Bolance, Guillén, and Nielsen (2003).

4.10 R Lab

4.10.1 European Stock Indices

This lab uses four European stock indices in R's **EuStockMarkets** database. Run the following code to access the database, learn its mode and class, and plot the four time series. The `plot()` function will produce a plot tailored to the class of the object on which it is acting. Here four time series plots are produced because the class of **EuStockMarkets** is **mts**, multivariate time series.

```
data(EuStockMarkets)
mode(EuStockMarkets)
class(EuStockMarkets)
plot(EuStockMarkets)
```

If you right-click on the plot, a menu for printing or saving will open. There are alternative methods for printing graphs. For example,

```
pdf("EuStocks.pdf", width = 6, height = 5)
plot(EuStockMarkets)
graphics.off()
```

will send a pdf file to the working directory and the `width` and `height` parameters allow one to control the size and aspect ratio of the plot.

Problem 1 Write a brief description of the time series plots of the four indices. Do the series look stationary? Do the fluctuations in the series seem to be of constant size? If not, describe how the volatility fluctuates.

Next, run the following R code to compute and plot the log returns on the indices.

```
logR = diff(log(EuStockMarkets))
plot(logR)
```

Problem 2 Write a brief description of the time series plots of the four series of log returns. Do the series look stationary? Do the fluctuations in the series seem to be of constant size? If not, describe how the volatility fluctuates.

In R, data can be stored as a data frame, which does not assume that the data are in time order and would be appropriate, for example, with cross-sectional data. To appreciate how `plot()` works on a data frame rather than on a multivariate time series, run the following code. You will be plotting the same data as before, but they will be plotted in a different way.

```
plot(as.data.frame(logR))
```

Run the code that follows to create normal plots of the four indices and to test each for normality using the Shapiro–Wilk test. You should understand what each line of code does.

```
par(mfrow=c(2, 2))
for(i in colnames(logR))
{
  qqnorm(logR[, i], datax = T, main = i)
  qqline(logR[, i], datax = T)
  print(shapiro.test(logR[, i]))
}
```

Problem 3 Briefly describe the shape of each of the four normal plots and state whether the marginal distribution of each series is skewed or symmetric and whether its tails appear normal. If the tails do not appear normal, do they appear heavier or lighter than normal? What conclusions can be made from the Shapiro–Wilk tests? Include the plots with your work.

The next set of R code creates t -plots with 1, 4, 6, 10, 20, and 30 degrees of freedom and all four indices. However, for the remainder of this lab, only the DAX index will be analyzed. Notice how the reference line is created by the `abline()` function, which adds lines to a plot, and the `lm()` function, which fits a line to the quantiles. The `lm()` function is discussed in Chap. 9.

```

1 n=dim(logR)[1]
2 q_grid = (1:n) / (n + 1)
3 df_grid = c(1, 4, 6, 10, 20, 30)
4 index.names = dimnames(logR)[[2]]
5 for(i in 1:4)
6 {
7   # dev.new()
8   par(mfrow = c(3, 2))
9   for(df in df_grid)
10  {
11     qqplot(logR[,i], qt(q_grid,df),
12            main = paste(index.names[i], ", df = ", df) )
13     abline(lm(qt(c(0.25, 0.75), df = df) ~
14               quantile(logR[,i], c(0.25, 0.75))))
15   }
16 }
```

If you are running R from Rstudio, then line 7 should be left as it is. If you are working directly in R, then remove the “#” in this line to open a new window for each plot.

Problem 4 What does the code `q.grid = (1:n) / (n + 1)` do? What does `qt(q.grid, df = df[j])` do? What does `paste` do?

Problem 5 For the DAX index, state which choice of the degrees of freedom parameter gives the best-fitting t -distribution and explain why.

Run the next set of code to create a kernel density estimate and two parametric density estimates, t with `df` degrees of freedom and normal, for the DAX index. Here `df` equals 5, but you should vary `df` so that the t density agrees as closely as possible with the kernel density estimate.

At lines 5–6, a robust estimator of the standard deviation of the t -distribution is calculated using the `mad()` function. The default value of the argument `constant` is 1.4826, which is calibrated to the normal distribution since $1/\Phi^{-1}(3/4) = 1.4826$. To calibrate to the t -distribution, the normal quantile is replaced by the corresponding t -quantile and multiplied by `df/(df - 2)` to convert from the scale parameter to the standard deviation.

```

1 library("fGarch")
2 x=seq(-0.1, 0.1, by = 0.001)
```

```

3 par(mfrow = c(1, 1))
4 df = 5
5 mad_t = mad(logR[, 1],
6   constant = sqrt(df / (df - 2)) / qt(0.75, df))
7 plot(density(logR[, 1]), lwd = 2, ylim = c(0, 60))
8 lines(x, dstd(x, mean = mean(logR[, 1]), sd = mad_t, nu = df),
9   lty = 5, lwd = 2, col = "red")
10 lines(x, dnorm(x, mean = mean(logR[, 1]), sd = sd(logR[, 1])),
11   lty = 3, lwd = 4, col = "blue")
12 legend("topleft", c("KDE", paste("t: df = ", df), "normal"),
13   lwd = c(2, 2, 4), lty = c(1, 5, 3),
14   col = c("black", "red", "blue"))

```

To examine the left and right tails, plot the density estimate two more times, once zooming in on the left tail and then zooming in on the right tail. You can do this by using the `xlim` parameter of the `plot()` function and changing `ylim` appropriately. You can also use the `adjust` parameter in `density()` to smooth the tail estimate more than is done with the default value of `adjust`.

Problem 6 Do either of the parametric models provide a reasonably good fit to the first index? Explain.

Problem 7 Which bandwidth selector is used as the default by `density`? What is the default kernel?

Problem 8 For the CAC index, state which choice of the degrees of freedom parameter gives the best-fitting t-distribution and explain why.

4.10.2 McDonald's Prices and Returns

This section analyzes daily stock prices and returns of the McDonald's Corporation (MCD) over the period Jan-4-10 to Sep-5-14. The data set is in the file `MCD_PriceDail.csv`. Run the following commands to load the data and plot the adjusted closing prices:

```

data = read.csv('MCD_PriceDaily.csv')
head(data)
adjPrice = data[, 7]
plot(adjPrice, type = "l", lwd = 2)

```

Problem 9 Does the price series appear stationary? Explain your answer.

Problem 10 Transform the prices into log returns and call that series `LogRet`. Create a time series plot of `LogRet` and discuss whether or not this series appears stationary.

The following code produces a histogram of the McDonald's log returns. The histogram will have 80 evenly spaced bins, and the argument `freq = FALSE` specifies the density scale.

```
hist(LogRet, 80, freq = FALSE)
```

Also, make a QQ plot of `LogRet`.

Problem 11 *Discuss any features you see in the histogram and QQ plot, and, specifically, address the following questions: Do the log returns appear to be normally distributed? If not, in what ways do they appear non-normal? Are the log returns symmetrically distributed? If not, how are they skewed? Do the log returns seem heavy tailed compared to a normal distribution? How do the left and right tails compare; is one tail heavier than the other?*

4.11 Exercises

1. This problem uses the data set `ford.csv` on the book's web site. The data were taken from the `ford.s` data set in R's `fEcofin` package. This package is no longer on CRAN. This data set contains 2000 daily Ford returns from January 2, 1984, to December 31, 1991.
 - (a) Find the sample mean, sample median, and standard deviation of the Ford returns.
 - (b) Create a normal plot of the Ford returns. Do the returns look normally distributed? If not, how do they differ from being normally distributed?
 - (c) Test for normality using the Shapiro–Wilk test? What is the p -value? Can you reject the null hypothesis of a normal distribution at 0.01?
 - (d) Create several t -plots of the Ford returns using a number of choices of the degrees of freedom parameter (`df`). What value of `df` gives a plot that is as linear as possible? The returns include the return on Black Monday, October 19, 1987. Discuss whether or not to ignore that return when looking for the best choices of `df`.
 - (e) Find the standard error of the sample median using formula (4.3) with the sample median as the estimate of $F^{-1}(0.5)$ and a KDE to estimate f . Is the standard error of the sample median larger or smaller than the standard error of the sample mean?
2. Column seven of the data set `RecentFord.csv` on the book's web site contains Ford daily closing prices, adjusted for splits and dividends, for the years 2009–2013. Repeat Problem 1 using these more recent returns. One of returns is approximately -0.175 . For part (d), use that return in place of Black Monday. (Black Monday, of course, is not in this data set.) On what date did this return occur? Search the Internet for news about Ford that day. Why did the Ford price drop so precipitously that day?

3. This problem uses the **Garch** data set in R's **Ecdat** package.
- Using a solid curve, plot a kernel density estimate of the first differences of the variable **dy**, which is the U.S. dollar/Japanese yen exchange rate. Using a dashed curve, superimpose a normal density with the same mean and standard deviation as the sample. Do the two estimated densities look similar? Describe how they differ.
 - Repeat part (a), but with the mean and standard deviation equal to the median and MAD. Do the two densities appear more or less similar compared to the two densities in part (a)?
4. Suppose in a normal plot that the sample quantiles are plotted on the vertical axis, rather than on the horizontal axis as in this book.
- What is the interpretation of a convex pattern?
 - What is the interpretation of a concave pattern?
 - What is the interpretation of a convex-concave pattern?
 - What is the interpretation of a concave-convex pattern?
5. Let **diffbp** be the changes (that is, differences) in the variable **bp**, the U.S. dollar to British pound exchange rate, which is in the **Garch** data set of R's **Ecdat** package.
- Create a 3×2 matrix of normal plots of **diffbp** and in each plot add a reference line that goes through the p - and $(1 - p)$ -quantiles, where $p = 0.25, 0.1, 0.05, 0.025, 0.01$, and 0.0025 , respectively, for the six plots. Create a second set of six normal plots using n simulated $N(0, 1)$ random variables, where n is the number of changes in **bp** plotted in the first figure. Discuss how the reference lines change with the value of p and how the set of six different reference lines can help detect nonnormality.
 - Create a third set of six normal plots using changes in the logarithm of **bp**. Do the changes in $\log(\text{bp})$ look closer to being normally distributed than the changes in **bp**?
6. Use the following fact about the standard normal cumulative distribution function $\Phi(\cdot)$:
- $$\Phi^{-1}(0.025) = -1.96.$$
- What value is $\Phi^{-1}(0.975)$? Why?
 - What is the 0.975-quantile of the normal distribution with mean -1 and variance 2?
7. Suppose that Y_1, \dots, Y_n are *i.i.d.* with a uniform distribution on the interval $(0, 1)$, with density function f and distribution function F defined as
- $$f(x) = \begin{cases} 1 & \text{if } x \in (0, 1), \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad F(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ x & \text{if } x \in (0, 1), \\ 1 & \text{if } x \geq 1. \end{cases}$$

Use Result 4.1 to conclude which sample quantile q will have the smallest variance?

References

- Abramson, I. (1982) On bandwidth variation in kernel estimates—a square root law. *Annals of Statistics*, **9**, 168–176.
- Atkinson, A. C. (1985) *Plots, transformations, and regression: An introduction to graphical methods of diagnostic regression analysis*, Clarendon Press, Oxford.
- Bolance, C., Guillén, M., and Nielsen, J. P. (2003) Kernel density estimation of actuarial loss functions. *Insurance: Mathematics and Economics*, **32**, 19–36.
- Carroll, R. J., and Ruppert, D. (1988) *Transformation and Weighting in Regression*, Chapman & Hall, New York.
- Hoaglin, D. C., Mosteller, F., and Tukey, J. W., Eds. (1983) *Understanding Robust and Exploratory Data Analysis*, Wiley, New York.
- Hoaglin, D. C., Mosteller, F., and Tukey, J. W., Eds. (1985) *Exploring Data Tables, Trends, and Shapes*, Wiley, New York.
- Jones, M. C. (1990) Variable kernel density estimates and variable kernel density estimates. *Australian Journal of Statistics*, **32**, 361–371. (Note: The title is intended to be ironic and is not a misprint.)
- Kleiber, C., and Zeileis, A. (2008) *Applied Econometrics with R*, Springer, New York.
- Lehmann, E. L. (1999) *Elements of Large-Sample Theory*, Springer-Verlag, New York.
- Scott, D. W. (1992) *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley-Interscience, New York.
- Serfling, R. J. (1980) *Approximation Theorems of Mathematical Statistics*, Wiley, New York.
- Silverman, B. W. (1986) *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, London.
- Tukey, J. W. (1977) *Exploratory Data Analysis*, Addison-Wesley, Reading, MA.
- van der Vaart, A. W. (1998) *Asymptotic Statistics*, Cambridge University Press, Cambridge.
- Wand, M. P., and Jones, M. C. (1995) *Kernel Smoothing*, Chapman & Hall, London.
- Wand, M. P., Marron, J. S., and Ruppert, D. (1991) Transformations in density estimation, *Journal of the American Statistical Association*, **86**, 343–366.
- Yap, B. W., and Sim, C. H. (2011) Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, **81**, 2141–2155.

Modeling Univariate Distributions

5.1 Introduction

As seen in Chap. 4, usually the marginal distributions of financial time series are not well fit by normal distributions. Fortunately, there are a number of suitable alternative models, such as t -distributions, generalized error distributions, and skewed versions of t - and generalized error distributions. All of these will be introduced in this chapter. Typically, the parameters in these distributions are estimated by maximum likelihood. Sections 5.9 and 5.14 provide an introduction to the maximum likelihood estimator (MLE), and Sect. 5.18 provides references for further study on this topic.

Software for maximum likelihood is readily available for standard models, and a reader interested only in data analysis and modeling often need not be greatly concerned with the technical details of maximum likelihood. However, when performing a statistical analysis, it is always worthwhile to understand the underlying theory, at least at a conceptual level, since doing so can prevent misapplications. Moreover, when using a nonstandard model, often there is no software available for automatic computation of the MLE and one needs to understand enough theory to write a program to compute the MLE.

5.2 Parametric Models and Parsimony

In a parametric statistical model, the distribution of the data is completely specified except for a finite number of unknown parameters. For example, assume that Y_1, \dots, Y_n are i.i.d. from a t -distribution¹ with mean μ , variance

¹ The reader who is unfamiliar with t -distributions should look ahead to Sect. 5.5.2.

σ^2 , and degrees of freedom ν . Then this is a parametric model provided that, as is usually the case, one or more of μ , σ^2 , and ν are unknown.

A model should have only as many parameters as needed to capture the important features of the data. Each unknown parameter is another quantity to estimate and another source of estimation error. Estimation error, among other things, increases the uncertainty when one forecasts future observations. On the other hand, a statistical model must have enough parameters to adequately describe the behavior of the data. A model with too few parameters can create biases because the model does not fit the data well.

A statistical model with little bias, but without excess parameters, is called *parsimonious* and achieves a good tradeoff between bias and variance. Finding one or a few parsimonious models is an important part of data analysis.

5.3 Location, Scale, and Shape Parameters

Parameters are often classified as location, scale, or shape parameters depending upon which properties of a distribution they determine. A *location parameter* is a parameter that shifts a distribution to the right or left without changing the distribution's shape or variability. Scale parameters quantify dispersion. A parameter is a *scale parameter* for a univariate sample if the parameter is increased by the amount $|a|$ when the data are multiplied by a . Thus, if $\sigma(X)$ is a scale parameter for a random variable X , then $\sigma(aX) = |a|\sigma(X)$. A scale parameter is a constant multiple of the standard deviation provided that the latter is finite. Many examples of location and scale parameters can be found in the following sections. If λ is a scale parameter, then λ^{-1} is called an inverse-scale parameter. Since scale parameters quantify dispersion, inverse-scale parameters quantify precision.

If $f(y)$ is any fixed density, then $f(y - \mu)$ is a family of distributions with location parameter μ ; $\theta^{-1}f(y/\theta)$, $\theta > 0$, is a family of distributions with a scale parameter θ ; and $\theta^{-1}f\{\theta^{-1}(y - \mu)\}$ is a family of distributions with location parameter μ and scale parameter θ . These facts can be derived by noting that if Y has density $f(y)$ and $\theta > 0$, then, by Result A.1, $Y + \mu$ has density $f(y - \mu)$, θY has density $\theta^{-1}f(\theta^{-1}y)$, and $\theta Y + \mu$ has density $\theta^{-1}f\{\theta^{-1}(y - \mu)\}$.

A *shape* parameter is defined as any parameter that is not changed by location and scale changes. More precisely, for any $f(y)$, μ , and $\theta > 0$, the value of a shape parameter for the density $f(y)$ will equal the value of that shape parameter for $\theta^{-1}f\{\theta^{-1}(y - \mu)\}$. The degrees-of-freedom parameters of t -distributions and the log-standard deviations of lognormal distributions are shape parameters. Other shape parameters will be encountered later in this chapter. Shape parameters are often used to specify the skewness or tail weight of a distribution.

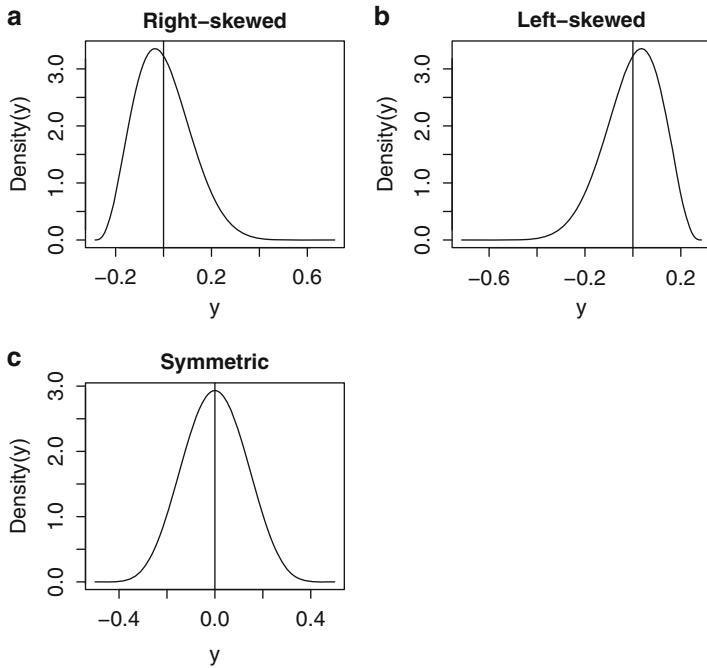


Fig. 5.1. Skewed and symmetric densities. In each case, the mean is zero and is indicated by a vertical line. The distributions in panels (a)–(c) are beta(4,10), beta(10,4), and beta(7,7), respectively. The R function `dbeta()` was used to calculate these densities.

5.4 Skewness, Kurtosis, and Moments

Skewness and kurtosis help characterize the shape of a probability distribution. *Skewness* measures the degree of asymmetry, with symmetry implying zero skewness, positive skewness indicating a relatively long right tail compared to the left tail, and negative skewness indicating the opposite. Figure 5.1 shows three densities, all with an expectation equal to 0. The densities are right-skewed, left-skewed, and symmetric about 0, respectively, in panels (a)–(c).

Kurtosis indicates the extent to which probability is concentrated in the center and especially the tails of the distribution rather than in the “shoulders,” which are the regions between the center and the tails.

In Sect. 4.3.2, the left tail was defined as the region from $-\infty$ to $\mu - 2\sigma$ and the right tail as the region from $\mu + 2\sigma$ to $+\infty$. Here μ and σ could be the mean and standard deviation or the median and MAD. Admittedly, these definitions are somewhat arbitrary. Reasonable definitions of *center* and *shoulder* would be that the center is the region from $\mu - \sigma$ to $\mu + \sigma$, the left

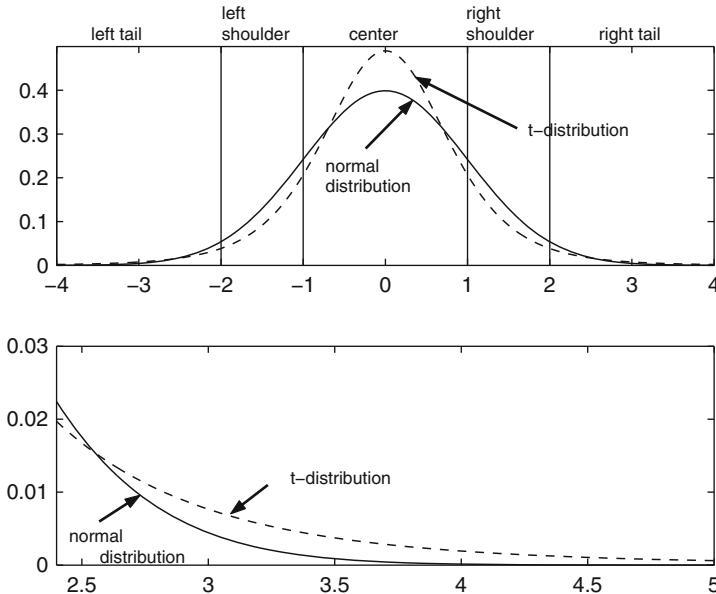


Fig. 5.2. Comparison of a normal density and a t-density with 5 degrees of freedom. Both densities have mean 0 and standard deviation 1. The upper plot also indicates the locations of the center, shoulders, and tail regions. The lower plot zooms in on the right tail region.

shoulder is from $\mu - 2\sigma$ to $\mu - \sigma$, and the right shoulder is from $\mu + \sigma$ to $\mu + 2\sigma$. See the upper plot in Fig. 5.2. Because skewness and kurtosis measure shape, they do not depend on the values of location and scale parameters.

The skewness of a random variable Y is

$$\text{Sk} = E \left\{ \frac{Y - E(Y)}{\sigma} \right\}^3 = \frac{E\{Y - E(Y)\}^3}{\sigma^3}.$$

To appreciate the meaning of the skewness, it is helpful to look at an example; the binomial distribution is convenient for that purpose. The skewness of the $\text{Binomial}(n, p)$ distribution is

$$\text{Sk}(n, p) = \frac{1 - 2p}{\sqrt{np(1-p)}}, \quad 0 < p < 1.$$

Figure 5.3 shows the binomial probability distribution and its skewness for $n = 10$ and four values of p . Notice that

1. the skewness is positive if $p < 0.5$, negative if $p > 0.5$, and 0 if $p = 0.5$;
2. the absolute skewness becomes larger as p moves closer to either 0 or 1 with n fixed;
3. the absolute skewness decreases to 0 as n increases to ∞ with p fixed;

Positive skewness is also called right skewness and negative skewness is called left skewness. A distribution is *symmetric* about a point θ if $P(Y > \theta + y) = P(Y < \theta - y)$ for all $y > 0$. In this case, θ is a location parameter and equals $E(Y)$, provided that $E(Y)$ exists. The skewness of any symmetric distribution is 0. Property 3 is not surprising in light of the central limit theorem. We know that the binomial distribution converges to the symmetric normal distribution as $n \rightarrow \infty$ with p fixed and not equal to 0 or 1.

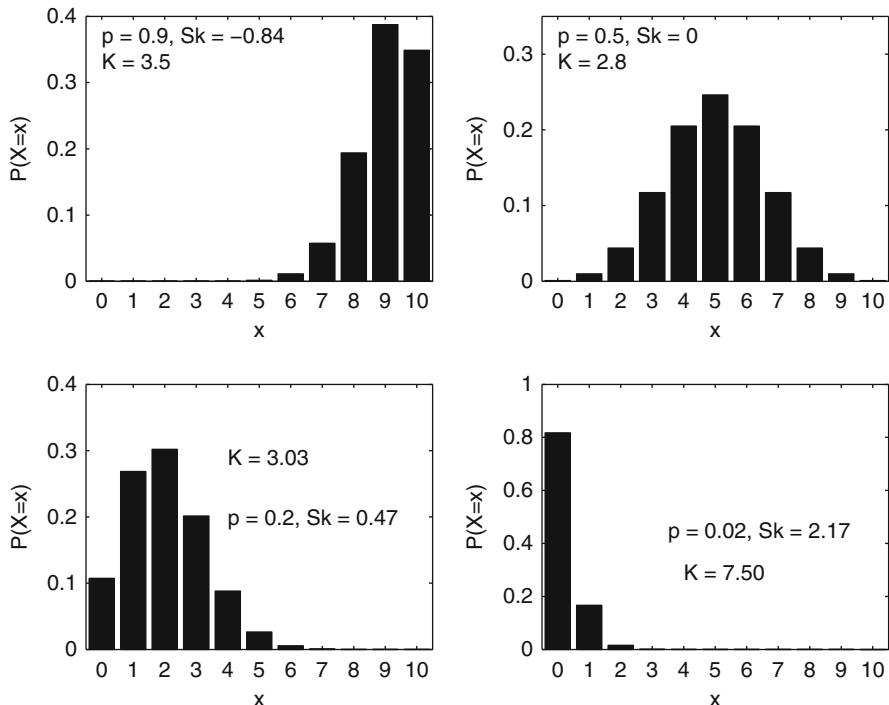


Fig. 5.3. Several binomial probability distributions with $n = 10$ and their skewness determined by the shape parameter p . Sk = skewness coefficient and K = kurtosis coefficient. The top left plot has left-skewness ($Sk = -0.84$). The top right plot has no skewness ($Sk = 0$). The bottom left plot has moderate right-skewness ($Sk = 0.47$). The bottom-left plot has strong right skewness ($Sk = 2.17$).

The kurtosis of a random variable Y is

$$\text{Kur} = E \left\{ \frac{Y - E(Y)}{\sigma} \right\}^4 = \frac{E\{Y - E(Y)\}^4}{\sigma^4}.$$

The kurtosis of a normal random variable is 3. The smallest possible value of the kurtosis is 1 and is achieved by any random variable taking exactly two

distinct values, each with probability $1/2$. The kurtosis of a $\text{Binomial}(n, p)$ distribution is

$$\text{Kur}^{\text{Bin}}(n, p) = 3 + \frac{1 - 6p(1 - p)}{np(1 - p)}.$$

Notice that $\text{Kur}^{\text{Bin}}(n, p) \rightarrow 3$, the value at the normal distribution, as $n \rightarrow \infty$ with p fixed, which is another sign of the central limit theorem at work. Figure 5.3 also gives the kurtosis of the distributions in that figure. $\text{Kur}^{\text{Bin}}(n, p)$ equals 1, the minimum value of kurtosis, when $n = 1$ and $p = 1/2$.

It is difficult to interpret the kurtosis of an asymmetric distribution because, for such distributions, kurtosis may measure both asymmetry and tail weight, so the binomial is not a particularly good example for understanding kurtosis. For that purpose we will look instead at t -distributions because they are symmetric. Figure 5.2 compares a normal density with the t_5 -density rescaled to have variance equal to 1. Both have a mean of 0 and a standard deviation of 1. The mean and standard deviation are location and scale parameters, respectively, and do not affect kurtosis. The parameter ν of the t -distribution is a shape parameter. The kurtosis of a t_ν -distribution is finite if $\nu > 4$ and then the kurtosis is

$$\text{Kur}^t(\nu) = 3 + \frac{6}{\nu - 4}. \quad (5.1)$$

For example, the kurtosis is 9 for a t_5 -distribution. Since the densities in Fig. 5.2 have the same mean and standard deviation, they also have the same tail, center, and shoulder regions, at least according to our somewhat arbitrary definitions of these regions, and these regions are indicated on the top plot. The bottom plot zooms in on the right tail. Notice that the t_5 -density has more probability in the tails and center than the $N(0, 1)$ density. This behavior of t_5 is typical of symmetric distributions with high kurtosis.

Every normal distribution has a skewness coefficient of 0 and a kurtosis of 3. The skewness and kurtosis must be the same for all normal distributions, because the normal distribution has only location and scale parameters, no shape parameters. The kurtosis of 3 agrees with formula (5.1) since a normal distribution is a t -distribution with $\nu = \infty$. The “excess kurtosis” of a distribution is $(\text{Kur} - 3)$ and measures the deviation of that distribution’s kurtosis from the kurtosis of a normal distribution. From (5.1) we see that the excess kurtosis of a t_ν -distribution is $6/(\nu - 4)$.

An exponential distribution² has a skewness equal to 2 and a kurtosis of 9. A double-exponential distribution has skewness 0 and kurtosis 6. Since the exponential distribution has only a scale parameter and the double-exponential has only a location and a scale parameter, their skewness and kurtosis must be constant since skewness and kurtosis depend only on shape parameters.

² The exponential and double-exponential distributions are defined in Appendix A.9.5.

The Lognormal(μ, σ^2) distribution, which is discussed in Appendix A.9.4, has the log-mean μ as a scale parameter and the log-standard deviation σ as a shape parameter—even though μ and σ are location and scale parameters for the normal distribution itself, they are scale and shape parameters for the lognormal. The effects of σ on lognormal shapes can be seen in Figs. 4.11 and A.1. The skewness coefficient of the lognormal(μ, σ^2) distribution is

$$\{\exp(\sigma^2) + 2\} \sqrt{\exp(\sigma^2) - 1}. \quad (5.2)$$

Since μ is a scale parameter, it has no effect on the skewness. The skewness is always positive and increases from 0 to ∞ as σ increases from 0 to ∞ .

Estimation of the skewness and kurtosis of a distribution is relatively straightforward if we have a sample, Y_1, \dots, Y_n , from that distribution. Let the sample mean and standard deviation be \bar{Y} and s . Then the sample skewness, denoted by $\widehat{\text{Sk}}$, is

$$\widehat{\text{Sk}} = \frac{1}{n} \sum_{i=1}^n \left(\frac{Y_i - \bar{Y}}{s} \right)^3, \quad (5.3)$$

and the sample kurtosis, denoted by $\widehat{\text{Kur}}$, is

$$\widehat{\text{Kur}} = \frac{1}{n} \sum_{i=1}^n \left(\frac{Y_i - \bar{Y}}{s} \right)^4. \quad (5.4)$$

Often the factor $1/n$ in (5.3) and (5.4) is replaced by $1/(n-1)$, in analogy with the sample variance. Both the sample skewness and the excess kurtosis should be near 0 if a sample is from a normal distribution. Deviations of the sample skewness and kurtosis from these values are an indication of nonnormality.

A word of caution is in order. Skewness and kurtosis are highly sensitive to outliers. Sometimes outliers are due to *contaminants*, that is, bad data not from the population being sampled. An example would be a data recording error. A sample from a normal distribution with even a single contaminant that is sufficiently outlying will appear highly nonnormal according to the sample skewness and kurtosis. In such a case, a normal plot *will* look linear, except that the single contaminant will stick out. See Fig. 5.4, which is a normal plot of a sample of 999 $N(0, 1)$ data points plus a contaminant equal to 30. This figure shows clearly that the sample is nearly normal but with an outlier. The sample skewness and kurtosis, however, are 10.85 and 243.04, which might give the false impression that the sample is far from normal. Also, even if there were no contaminants, a distribution could be extremely close to a normal distribution except in the extreme tails and yet have a skewness or excess kurtosis that is very different from 0.

5.4.1 The Jarque–Bera Test

The Jarque–Bera test of normality compares the sample skewness and kurtosis to 0 and 3, their values under normality. The test statistic is

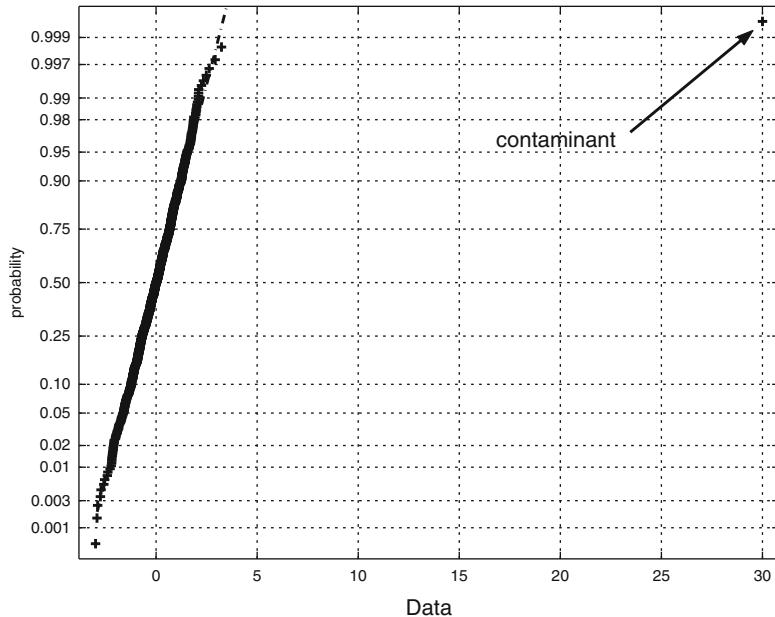


Fig. 5.4. Normal plot of a sample of 999 $N(0, 1)$ data plus a contaminant.

$$JB = n\{\widehat{Sk}^2/6 + (\widehat{Kur} - 3)^2/24\},$$

which, of course, is 0 when \widehat{Sk} and \widehat{Kur} , respectively, have the values 0 and 3, the values expected under normality, and increases as \widehat{Sk} and \widehat{Kur} deviate from these values. In R, the test statistic and its p -value can be computed with the `jarque.bera.test()` function.

A large-sample approximation is used to compute a p -value. Under the null hypothesis, JB converges to the chi-square distribution with 2 degrees of freedom (χ_2^2) as the sample size becomes infinite, so the approximate p -value is $1 - F_{\chi_2^2}(JB)$, where $F_{\chi_2^2}$ is the CDF of the χ_2^2 -distribution.

5.4.2 Moments

The expectation, variance, skewness coefficient, and kurtosis of a random variable are all special cases of moments, which will be defined in this section.

Let X be a random variable. The k th moment of X is $E(X^k)$, so in particular the first moment is the expectation of X . The k th absolute moment is $E|X|^k$.

The k th central moment is

$$\mu_k = E \left[\{X - E(X)\}^k \right], \quad (5.5)$$

so, for example, μ_2 is the variance of X . The skewness coefficient of X is

$$\text{Sk}(X) = \frac{\mu_3}{(\mu_2)^{3/2}}, \quad (5.6)$$

and the kurtosis of X is

$$\text{Kur}(X) = \frac{\mu_4}{(\mu_2)^2}. \quad (5.7)$$

5.5 Heavy-Tailed Distributions

As discussed in earlier chapters, distributions with higher tail probabilities compared to a normal distribution are called *heavy-tailed*. Because kurtosis is particularly sensitive to tail weight, high kurtosis is nearly synonymous with having a heavy tailed distribution. Heavy-tailed distributions are important models in finance, because equity returns and other changes in market prices usually have heavy tails. In finance applications, one is especially concerned when the return distribution has heavy tails because of the possibility of an extremely large negative return, which could, for example, entirely deplete the capital reserves of a firm. If one sells short,³ then large positive returns are also worrisome.

5.5.1 Exponential and Polynomial Tails

Double-exponential distributions have slightly heavier tails than normal distributions. This fact can be appreciated by comparing their densities. The density of the double-exponential with scale parameter θ is proportional to $\exp(-|y/\theta|)$ and the density of the $N(0, \sigma^2)$ distribution is proportional to $\exp\{-0.5(y/\sigma)^2\}$. The term $-y^2$ converges to $-\infty$ much faster than $-|y|$ as $|y| \rightarrow \infty$. Therefore, the normal density converges to 0 much faster than the double-exponential density as $|y| \rightarrow \infty$. The generalized error distributions discussed soon in Sect. 5.6 have densities proportional to

$$\exp(-|y/\theta|^\alpha), \quad (5.8)$$

where $\alpha > 0$ is a shape parameter and θ is a scale parameter. The special cases of $\alpha = 1$ and 2 are, of course, the double-exponential and normal densities. If $\alpha < 2$, then a generalized error distribution will have heavier tails than a normal distribution, with smaller values of α implying heavier tails. In particular, $\alpha < 1$ implies a tail heavier than that of a double-exponential distribution.

However, no density of the form (5.8) will have truly heavy tails, and, in particular, $E(|Y|^k) < \infty$ for all k , so all moments are finite. To achieve a very heavy right tail, the density must be such that

$$f(y) \sim Ay^{-(\alpha+1)} \text{ as } y \rightarrow \infty \quad (5.9)$$

³ See Sect. 16.5 for a discussion of short selling.

for some $A > 0$ and $a > 0$, which will be called a *right polynomial tail*, in contrast to

$$f(y) \sim A \exp(-y/\theta) \text{ as } y \rightarrow \infty \quad (5.10)$$

for some $A > 0$ and $\theta > 0$, which will be called an *exponential right tail*. Polynomial and exponential left tails are defined analogously.

A polynomial tail is also called a *Pareto tail* after the Pareto distribution defined in Appendix A.9.8. The parameter a of a polynomial tail is called the *tail index*. The smaller the value of a , the heavier the tail. The value of a must be greater than 0, because if $a \leq 0$, then the density integrates to ∞ , not 1. An exponential tail as in (5.8) is lighter than any polynomial tail, since

$$\frac{\exp(-|y/\theta|^\alpha)}{|y|^{-(a+1)}} \rightarrow 0 \text{ as } |y| \rightarrow \infty$$

for all $\theta > 0$, $\alpha > 0$, and $a > 0$.

It is, of course, possible to have left and right tails that behave quite differently from each other. For example, one could be polynomial and the other exponential, or they could both be polynomial but with different indices.

A density with both tails polynomial will have a finite k th absolute moment only if the smaller of the two tail indices is larger than k . If both tails are exponential, then all moments are finite.

5.5.2 *t*-Distributions

The *t*-distributions have played an extremely important role in classical statistics because of their use in testing and confidence intervals when the data are modeled as having normal distributions. More recently, *t*-distributions have gained added importance as models for the distribution of heavy-tailed phenomena such as financial markets data.

We will start with some definitions. If Z is $N(0, 1)$, W is chi-squared⁴ with ν degrees of freedom, and Z and W are independent, then the distribution of

$$Z/\sqrt{W/\nu} \quad (5.11)$$

is called the *t-distribution* with ν *degrees of freedom* and denoted t_ν . The α -upper quantile of the t_ν -distribution is denoted by $t_{\alpha,\nu}$ and is used in tests and confidence intervals about population means, regression coefficients, and parameters in time series models.⁵ In testing and interval estimation, the parameter ν generally assumes only positive integer values, but when the *t*-distribution is used as a model for data, ν is restricted only to be positive.

The density of the t_ν -distribution is

$$f_{t,\nu}(y) = \left[\frac{\Gamma\{(\nu+1)/2\}}{(\pi\nu)^{1/2}\Gamma(\nu/2)} \right] \frac{1}{\{1+(y^2/\nu)\}^{(\nu+1)/2}}. \quad (5.12)$$

⁴ Chi-squared distributions are discussed in Appendix A.10.1.

⁵ See Appendix A.17.1 for confidence intervals for the mean.

Here Γ is the *gamma function* defined by

$$\Gamma(t) = \int_0^\infty x^{t-1} \exp(-x) dx, \quad t > 0. \quad (5.13)$$

The quantity in large square brackets in (5.12) is just a constant, though a somewhat complicated one.

The variance of a t_ν is finite and equals $\nu/(\nu - 2)$ if $\nu > 2$. If $0 < \nu \leq 1$, then the expected value of the t_ν -distribution does not exist and the variance is not defined.⁶ If $1 < \nu \leq 2$, then the expected value is 0 and the variance is infinite. If Y has a t_ν -distribution, then

$$\mu + \lambda Y$$

is said to have a $t_\nu(\mu, \lambda^2)$ distribution, and λ will be called *the scale parameter*. With this notation, the t_ν and $t_\nu(0, 1)$ distributions are the same. If $\nu > 1$, then the $t_\nu(\mu, \lambda^2)$ distribution has a mean equal to μ , and if $\nu > 2$, then it has a variance equal to $\lambda^2 \nu / (\nu - 2)$.

The t -distribution will also be called the *classical t-distribution* to distinguish it from the standardized t -distribution defined in the next section.

Standardized t -Distributions

Instead of the classical t -distribution just discussed, some software uses a “standardized” version of the t -distribution. The difference between the two versions is merely notational, but it is important to be aware of this difference.

The $t_\nu\{0, (\nu - 2)/\nu\}$ distribution with $\nu > 2$ has a mean equal to 0 and variance equal to 1 and is called a *standardized t-distribution*, and will be denoted by $t_\nu^{\text{std}}(0, 1)$. More generally, for $\nu > 2$, define the $t_\nu^{\text{std}}(\mu, \sigma^2)$ distribution to be equal to the $t_\nu[\mu, \{(\nu - 2)/\nu\}\sigma^2]$ distribution, so that μ and σ^2 are the mean and variance of the $t_\nu^{\text{std}}(\mu, \sigma^2)$ distribution. For $\nu \leq 2$, $t_\nu^{\text{std}}(\mu, \sigma^2)$ cannot be defined since the t -distribution does not have a finite variance in this case. The advantage in using the $t_\nu^{\text{std}}(\mu, \sigma^2)$ distribution is that σ^2 is the variance, whereas for the $t_\nu(\mu, \lambda^2)$ distribution, λ^2 is not the variance but instead λ^2 is the variance times $(\nu - 2)/\nu$.

Some software uses the standardized t -distribution while other software uses the classical t -distribution. It is, of course, important to understand which t -distribution is being used in any specific application. However, estimates from one model can be translated easily into the estimates one would obtain from the other model; see Sect. 5.14 for an example.

⁶ See Appendix A.3 for discussion of when the mean and the variance exist and when they are finite.

t-Distributions Have Polynomial Tails

The t -distributions are a class of heavy-tailed distributions and can be used to model heavy-tail returns data. For t -distributions, both the kurtosis and the weight of the tails increase as ν gets smaller. When $\nu \leq 4$, the tail weight is so high that the kurtosis is infinite. For $\nu > 4$, the kurtosis is given by (5.1).

By (5.12), the t -distribution's density is proportional to

$$\frac{1}{\{1 + (y^2/\nu)\}^{(\nu+1)/2}}$$

which for large values of $|y|$ is approximately

$$\frac{1}{(y^2/\nu)^{(\nu+1)/2}} \propto |y|^{-(\nu+1)}.$$

Therefore, the t -distribution has polynomial tails with tail index $a = \nu$. The smaller the value of ν , the heavier the tails.

5.5.3 Mixture Models

Discrete Mixtures

Another class of models containing heavy-tailed distributions is the set of *mixture models*. Consider a distribution that is 90 % $N(0, 1)$ and 10 % $N(0, 25)$. A random variable Y with this distribution can be obtained by generating a normal random variable X with mean 0 and variance 1 and a uniform(0,1) random variable U that is independent of X . If $U < 0.9$, then $Y = X$. If $U \geq 0.9$, then $Y = 5X$. If an independent sample from this distribution is generated, then the expected percentage of observations from the $N(0, 1)$ component is 90 %. The actual percentage is random; in fact, it has a Binomial($n, 0.9$) distribution, where n is a sample size. By the law of large numbers, the actual percentage converges to 90 % as $n \rightarrow \infty$. This distribution could be used to model a market that has two *regimes*, the first being “normal volatility” and second “high volatility,” with the first regime occurring 90 % of the time.

This is an example of a *finite* or *discrete normal mixture distribution*, since it is a mixture of a finite number, here two, different normal distributions called the *components*. A random variable with this distribution has a variance equal to 1 with 90 % probability and equal to 25 with 10 % probability. Therefore, the variance of this distribution is $(0.9)(1) + (0.1)(25) = 3.4$, so its standard deviation is $\sqrt{3.4} = 1.84$. This distribution is much different from an $N(0, 3.4)$ distribution, even though the two distributions have the same mean and variance. To appreciate this, look at Fig. 5.5.

You can see in Fig. 5.5a that the two densities look quite different. The normal density looks much more dispersed than the normal mixture, but they actually have the same variances. What is happening? Look at the detail of

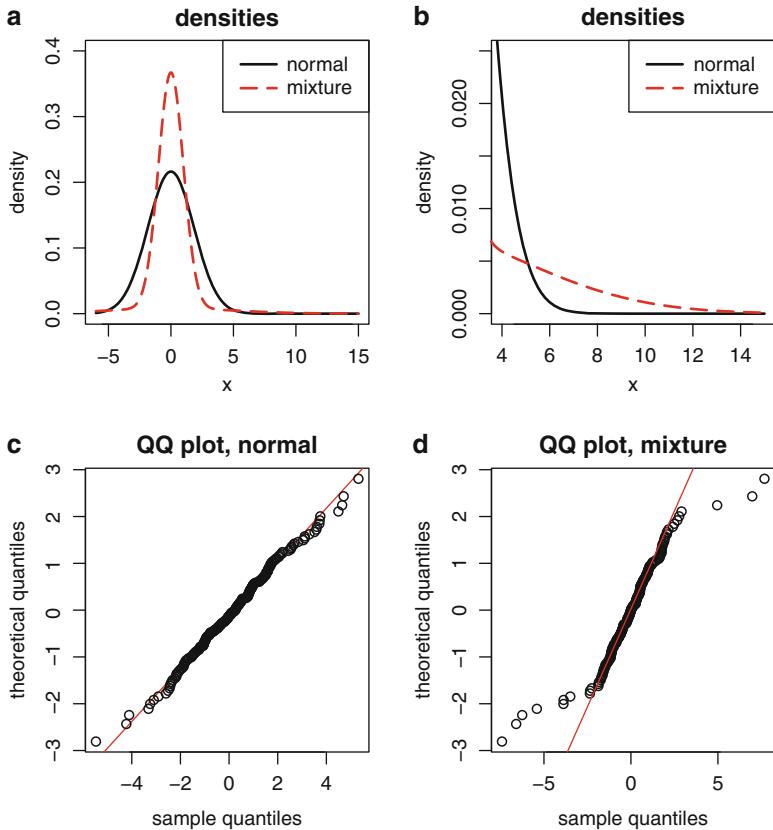


Fig. 5.5. Comparison of $N(0, 3.4)$ distribution and heavy-tailed normal mixture distributions. These distributions have the same mean and variance. The normal mixture distribution is 90 % $N(0, 1)$ and 10 % $N(0, 25)$. In (c) and (d) the sample size is 200. In panel (a), the left tail is not shown fully to provide detail at the center and because the left tail is the mirror image of the right tail. (b) Detail of right tail.

the right tails in panel (b). The normal mixture density is much higher than the normal density when x is greater than 6. This is the “outlier” region (along with $x < -6$).⁷ The normal mixture has far more outliers than the normal distribution and the outliers come from the 10 % of the population with a variance of 25. Remember that ± 6 is only 6/5 standard deviations from the mean, using the standard deviation 5 of the component from which they come. Thus, these observations are not outlying relative to their component’s standard deviation of 5, only relative to the population standard deviation of

⁷ There is nothing special about “6” to define the boundary of the outlier range, but a specific number was needed to make numerical comparisons. Clearly, $|x| > 7$ or $|x| > 8$, say, would have been just as appropriate as outlier ranges.

$\sqrt{3.4} = 1.84$ since $6/1.84 = 3.25$ and three or more standard deviations from the mean is generally considered rather outlying.

Outliers have a powerful effect on the variance and this small fraction of outliers inflates the variance from 1.0 (the variance of 90 % of the population) to 3.4.

Let's see how much more probability the normal mixture distribution has in the outlier range $|x| > 6$ compared to the normal distribution. For an $N(0, \sigma^2)$ random variable Y ,

$$P\{|Y| > y\} = 2\{1 - \Phi(y/\sigma)\}.$$

Therefore, for the normal distribution with variance 3.4,

$$P\{|Y| > 6\} = 2\{1 - \Phi(6/\sqrt{3.4})\} = 0.0011.$$

For the normal mixture population that has variance 1 with probability 0.9 and variance 25 with probability 0.1, we have that

$$\begin{aligned} P\{|Y| > 6\} &= 2\left[0.9\{1 - \Phi(6)\} + 0.1\{1 - \Phi(6/5)\}\right] \\ &= 2\{(0.9)(0) + (0.1)(0.115)\} = 0.023. \end{aligned}$$

Since $0.023/0.0011 \approx 21$, the normal mixture distribution is 21 times more likely to be in this outlier range than the $N(0, 3.4)$ population, even though both have a variance of 3.4. In summary, the normal mixture is much more prone to outliers than a normal distribution with the same mean and standard deviation. So, we should be much more concerned about very large negative returns if the return distribution is more like the normal mixture distribution than like a normal distribution. Large positive returns are also likely under a normal mixture distribution and would be of concern if an asset was sold short.

It is not difficult to compute the kurtosis of this normal mixture. Because a normal distribution has kurtosis equal to 3, if Z is $N(\mu, \sigma^2)$, then $E(Z - \mu)^4 = 3\sigma^4$. Therefore, if Y has this normal mixture distribution, then

$$E(Y^4) = 3\{0.9 + (0.1)25^2\} = 190.2$$

and the kurtosis of X is $190.2/3.4^2 = 16.45$.

Normal probability plots of samples of size 200 from the normal and normal mixture distributions are shown in panels (c) and (d) of Fig. 5.5. Notice how the outliers in the normal mixture sample give the probability plot a convex-concave pattern typical of heavy-tailed data. The deviation of the plot of the normal sample from linearity is small and is due entirely to randomness.

In this example, the conditional variance of any observation is 1 with probability 0.9 and 25 with probability 0.1. Because there are only two components, the conditional variance is discrete, in fact, with only two possible values, and the example was easy to analyze. This example is a normal *scale mixture* because only the scale parameter σ varies between components. It is also a *discrete mixture* because there are only a finite number of components.

Continuous Mixtures

The marginal distributions of the GARCH processes studied in Chap. 14 are also normal scale mixtures, but with infinitely many components and a continuous distribution of the conditional variance. Although GARCH processes are more complex than the simple mixture model in this section, the same theme applies—a nonconstant conditional variance of a mixture distribution induces heavy-tailed marginal distributions even though the conditional distributions are normal distributions and have relatively light tails.

The general definition of a normal scale mixture is that it is the distribution of the random variable

$$\mu + \sqrt{U}Z \quad (5.14)$$

where μ is a constant equal to the mean, Z is $N(0, 1)$, U is a positive random variable giving the variance of each component, and Z and U are independent. If U can assume only a finite number of values, then (5.14) is a *discrete* (or finite) scale mixture distribution. If U is continuously distributed, then we have a *continuous scale mixture distribution*. The distribution of U is called the *mixing distribution*. By (5.11), a t_ν -distribution is a continuous normal scale mixture with $\mu = 0$ and $U = \nu/W$, where ν and W are as defined above Eq. (5.11).

Despite the apparent heavy tails of a *finite* normal mixture, the tails are exponential, not polynomial. A continuous normal mixture can have a polynomial tail if the mixture distribution's tail is heavy enough, e.g., as in t -distributions.

5.6 Generalized Error Distributions

Generalized error distributions mentioned briefly in Sect. 5.5.1 have exponential tails. This section provides more detailed information about them. The standardized generalized error distribution, or GED, with shape parameter ν has density

$$f_{\text{ged}}^{\text{std}}(y|\nu) = \kappa(\nu) \exp \left\{ -\frac{1}{2} \left| \frac{y}{\lambda_\nu} \right|^\nu \right\}, \quad -\infty < y < \infty,$$

where $\kappa(\nu)$ and λ_ν are constants given by

$$\lambda_\nu = \left\{ \frac{2^{-2/\nu} \Gamma(\nu^{-1})}{\Gamma(3/\nu)} \right\}^{1/2} \quad \text{and} \quad \kappa(\nu) = \frac{\nu}{\lambda_\nu 2^{1+1/\nu} \Gamma(\nu^{-1})}$$

and chosen so that the function integrates to 1, as it must to be a density, and the variance is 1. The latter property is not necessary but is often convenient.

The shape parameter $\nu > 0$ determines the tail weight, with smaller values of ν giving greater tail weight. When $\nu = 2$, a GED is a normal distribution,

and when $\nu = 1$, it is a double-exponential distribution. The generalized error distributions can give tail weights intermediate between the normal and double-exponential distributions by having $1 < \nu < 2$. They can also give tail weights more extreme than the double-exponential distribution by having $\nu < 1$.

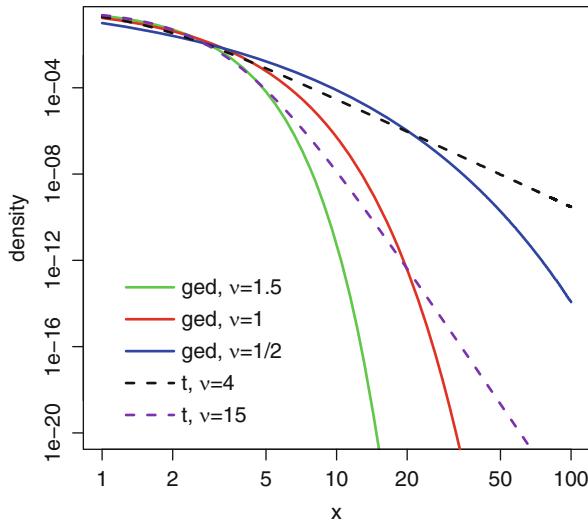


Fig. 5.6. A comparison of the tails of several generalized error (solid curves) and t -distributions (dashed curves).

Figure 5.6 shows the right tails of several t - and generalized error densities with mean 0 and variance 1.⁸ Since they are standardized, the argument y is the number of standard deviations from the median of 0. Because t -distributions have polynomial tails, any t -distribution is heavier-tailed than any generalized error distribution. However, this is only an asymptotic result as $y \rightarrow \infty$. In the more practical range of y , tail weight depends as much on the tail weight parameter as it does on the choice between a t -distribution or a generalized error distribution.

The t -distributions and generalized error densities also differ in their shapes at the median. This can be seen in Fig. 5.7, where the generalized error densities have sharp peaks at the median with the sharpness increasing as ν decreases. In comparison, a t -density is smooth and rounded near the median, even with ν small. If a sample is better fit by a t -distribution than by a generalized error distribution, this may be due more to the sharp central peaks of generalized error densities than to differences between the tails of the two types of distributions.

⁸ This plot and Fig. 5.7 used the R functions `dged()` and `dstd()` in the `fGarch` package.

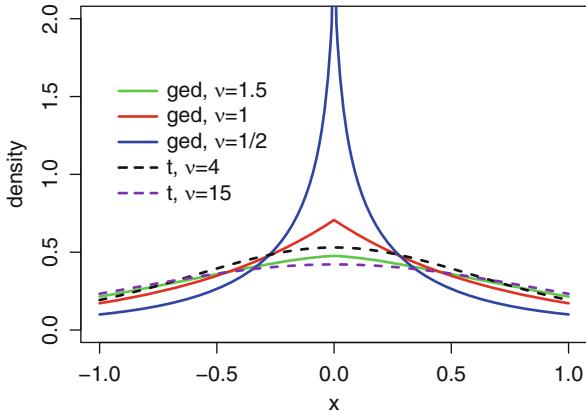


Fig. 5.7. A comparison of the centers of several generalized error (solid) and t -densities (dashed) with mean 0 and variance 1.

The $f_{\text{ged}}^{\text{std}}(y|\nu)$ density is symmetric about 0, which is its mean, median, and mode, and has a variance equal to 1. However, it can be shifted and rescaled to create a location-scale family. The GED distribution with mean μ , variance σ^2 , and shape parameter ν has density

$$f_{\text{ged}}^{\text{std}}(y|\mu, \sigma^2, \nu) := f_{\text{ged}}^{\text{std}}\{(y - \mu)/\sigma|\nu\}/\sigma.$$

5.7 Creating Skewed from Symmetric Distributions

Returns and other financial markets data typically have no natural lower or upper bounds, so one would like to use models with support equal to $(-\infty, \infty)$. This is fine if the data are symmetric since then one can use, for example, normal, t , or generalized error distributions as models. What if the data are skewed? Unfortunately, many of the well-known skewed distributions, such as gamma and log-normal distributions, have support $[0, \infty)$ and so are not suitable for modeling the changes in many types of financial markets data. This section describes a remedy to this problem.

Fernandez and Steel (1998) have devised a clever way for inducing skewness in symmetric distributions such as normal and t -distributions. The `fGarch` package in R implements their idea. Let ξ be a positive constant and f a density that is symmetric about 0. Define

$$f^*(y|\xi) = \begin{cases} f(y\xi) & \text{if } y < 0, \\ f(y/\xi) & \text{if } y \geq 0. \end{cases} \quad (5.15)$$

Since $f^*(y|\xi)$ integrates to $(\xi^{-1} + \xi)/2$, $f^*(y|\xi)$ is divided by this constant to create a probability density. After this normalization, the density is given a

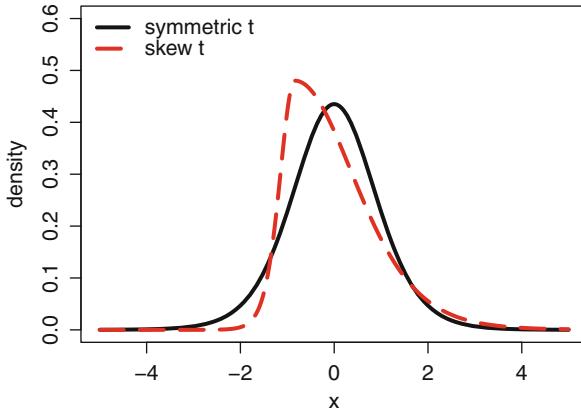


Fig. 5.8. Symmetric (solid) and skewed (dashed) t -densities, both with mean 0, standard deviation 1, and $\nu = 10$. $\xi = 2$ in the skewed density. Notice that the mode of the skewed density lies to the left of its mean, a typical behavior of right-skewed densities.

location shift and scale change to induce a mean equal to 0 and variance of 1. The final result is denoted by $f(y|\xi)$.

If $\xi > 1$, then the right half of $f(y|\xi)$ is elongated relative to the left half, which induces right skewness. Similarly, $\xi < 1$ induces left skewness. Figure 5.8 shows standardized symmetric and skewed t -distributions⁹ with $\nu = 10$ in both cases and $\xi = 2$ for the skewed distribution. Similarly, if $\xi < 1$, then $f(y|\xi)$ is left skewed.

If f is a t -distribution, then $f(y|\xi)$ is called a skewed t -distribution. Skewed t -distributions include symmetric t -distributions as special cases where $\xi = 1$. In the same way, skewed generalized error distributions are created when f is a generalized error distribution. The skewed distributions just described will be called Fernandez–Steel or F-S skewed distributions.

Fernandez and Steel's technique is not the only method for creating skewed versions of the normal and t -distributions. Azzalini and Capitanio (2003) have created somewhat different skewed normal and t -distributions.¹⁰ These distributions have a shape parameter α that determines the skewness; the distribution is left-skewed, symmetric, or right-skewed according to whether α is negative, zero, or positive. An example is given in Sect. 5.14 and multivariate versions are discussed in Sect. 7.9. We will refer to these as Azzalini–Capitanio or A-C skewed distributions.

⁹ R's `dstd()` (for symmetric t) and `dsstd()` (for skewed t) functions in the `fGarch` package were used for to create this plot.

¹⁰ Programs for fitting these distributions, computing their densities, quantile, and distribution functions, and generating random samples are available in R's `sn` package.

The A-C skewed normal density is $g(y|\xi, \omega, \alpha) = (2/\omega)\phi(z)\Phi(\alpha z)$ where $z = (y - \xi)/\omega$ and $\phi()$ and $\Phi()$ are the $N(0, 1)$ density and CDF, respectively. The parameters ξ , ω , and α determine location, scale, and skewness and are called the direct parameters or DP. The parameters ξ and ω are the mean and standard deviation of $\phi(z)$ and α determines the amount of skewness induced by $\Phi(\alpha z)$. The skewness of $g(y|\xi, \omega, \alpha)$ is positive if $\alpha > 0$ and negative if $\alpha < 0$.

The direct parameters do not have simple interpretations for the skew normal density $g(y|\xi, \omega, \alpha)$. Therefore, the so-called centered parameters (CP) are defined to be the mean, standard deviation, and skewness of $g(y|\xi, \omega, \alpha)$.

The A-C skew- t distribution has four parameters. The four DP are the mean, scale, and degrees of freedom of a t -density and α which measures the amount of skewness induced into that density. The CP are the mean, standard deviation, skewness, and kurtosis of the skew t .

5.8 Quantile-Based Location, Scale, and Shape Parameters

As has been seen, the mean, standard deviation, skewness coefficient, and kurtosis are moments-based location, scale, and shape parameters. Although they are widely used, they have the drawbacks that they are sensitive to outliers and may be undefined or infinite for distributions with heavy tails. An alternative is to use parameters based on quantiles.

Any quantile $F^{-1}(p)$, $0 < p < 1$, is a location parameter. A positive weighted average of quantiles, that is, $\sum_{\ell=1}^L w_\ell F^{-1}(p_\ell)$, where $w_\ell > 0$ for all ℓ and $\sum_{\ell=1}^L w_\ell = 1$, is also a location parameter. A simple example is $\{F^{-1}(1-p) + F^{-1}(p)\}/2$ where $0 < p < 1/2$, which equals the mean and median if F is symmetric.

A scale parameter can be obtained from the difference between two quantiles:

$$s(p_1, p_2) = \frac{F^{-1}(p_2) - F^{-1}(p_1)}{a}$$

where $0 < p_1 < p_2 < 1$ and a is a positive constant. An obvious choice is $p_1 < 1/2$ and $p_2 = 1 - p_1$. If $a = \Phi^{-1}(p_2) - \Phi^{-1}(p_1)$, then $s(p_1, p_2)$ is equal to the standard deviation when F is a normal distribution. If $a = 1$, then $s(1/4, 3/4)$ is called the *interquartile range* or IQR.

A quantile-based shape parameter that quantifies skewness is a ratio with the numerator the difference between two scale parameters and the denominator a scale parameter:

$$\frac{s(1/2, p_2) - s(1/2, p_1)}{s(p_3, p_4)}. \quad (5.16)$$

where $p_1 < 1/2$, $p_2 > 1/2$, and $0 < p_3 < p_4 < 1$. For example, one could use $p_2 = 1 - p_1$, $p_4 = p_2$, and $p_3 = p_1$.

A quantile-based shape parameter that quantifies tail weight is the ratio of two scale parameters:

$$\frac{s(p_1, 1 - p_1)}{s(p_2, 1 - p_2)}, \quad (5.17)$$

where $0 < p_1 < p_2 < 1/2$. For example, one might have $p_1 = 0.01$ or 0.05 and $p_2 = 0.25$.

5.9 Maximum Likelihood Estimation

Maximum likelihood is the most important and widespread method of estimation. Many well-known estimators such as the sample mean, and the least-squares estimator in regression are maximum likelihood estimators if the data have a normal distribution. Maximum likelihood estimation generally provides more efficient (less variable) estimators than other techniques of estimation. As an example, for a t -distribution, the maximum likelihood estimator of the mean is more efficient than the sample mean.

Let $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$ be a vector of data and let $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^\top$ be a vector of parameters. Let $f(\mathbf{Y}|\boldsymbol{\theta})$ be the density of \mathbf{Y} , which depends on the parameters.

The function $L(\boldsymbol{\theta}) = f(\mathbf{Y}|\boldsymbol{\theta})$ viewed as a function of $\boldsymbol{\theta}$ with \mathbf{Y} fixed at the observed data is called the *likelihood function*. It tells us the likelihood of the sample that was actually observed. The *maximum likelihood estimator* (MLE) is the value of $\boldsymbol{\theta}$ that maximizes the likelihood function. In other words, the MLE is the value of $\boldsymbol{\theta}$ at which the likelihood of the observed data is largest. We denote the MLE by $\hat{\boldsymbol{\theta}}_{\text{ML}}$. Often it is mathematically easier to maximize $\log\{L(\boldsymbol{\theta})\}$, which is called the log-likelihood. If the data are independent, then the likelihood is the product of the marginal densities and products are cumbersome to differentiate. Taking the logarithm converts the product into an easily differentiated sum. Also, in numerical computations, using the log-likelihood reduces the possibility of underflow or overflow. Since the log function is increasing, maximizing $\log\{L(\boldsymbol{\theta})\}$ is equivalent to maximizing $L(\boldsymbol{\theta})$.

In examples found in introductory statistics textbooks, it is possible to find an explicit formula for the MLE. With more complex models such as the ones we will mostly be using, there is no explicit formula for the MLE. Instead, one must write a program that computes $\log\{L(\boldsymbol{\theta})\}$ for any $\boldsymbol{\theta}$ and then use optimization software to maximize this function numerically; see Example 5.3. The R functions `optim()` and `nlsminb()` minimize functions and can be applied to $-L(\boldsymbol{\theta})$.

For many important models, such as the examples in the Sect. 5.14 and the ARIMA and GARCH time series models discussed in Chap. 12, R and other software packages contain functions to find the MLE for these models.

5.10 Fisher Information and the Central Limit Theorem for the MLE

Standard errors are essential for measuring the accuracy of estimators. We have formulas for the standard errors of simple estimators such as \bar{Y} , but what about standard errors for other estimators? Fortunately, there is a simple method for calculating the standard error of a maximum likelihood estimator.

We assume for now that θ is one-dimensional. The *Fisher information* is defined to be minus the expected second derivative of the log-likelihood, so if $\mathcal{I}(\theta)$ denotes the Fisher information, then

$$\mathcal{I}(\theta) = -E \left[\frac{d^2}{d\theta^2} \log\{L(\theta)\} \right]. \quad (5.18)$$

The standard error of $\hat{\theta}$ is simply the inverse square root of the Fisher information, with the unknown θ replaced by $\hat{\theta}$:

$$s_{\hat{\theta}} = \frac{1}{\sqrt{\mathcal{I}(\hat{\theta})}}. \quad (5.19)$$

Example 5.1. Fisher information for a normal model mean

Suppose that Y_1, \dots, Y_n are i.i.d. $N(\mu, \sigma^2)$ with σ^2 known. The log-likelihood for the unknown parameter μ is

$$\log\{L(\mu)\} = -\frac{n}{2}\{\log(\sigma^2) + \log(2\pi)\} - \frac{1}{2\sigma^2} \sum_{i=1}^n (Y_i - \mu)^2.$$

Therefore,

$$\frac{d}{d\mu} \log\{L(\mu)\} = \frac{1}{\sigma^2} \sum_{i=1}^n (Y_i - \mu),$$

so that \bar{Y} is the MLE of μ and

$$\frac{d^2}{d\mu^2} \log\{L(\mu)\} = -\frac{\sum_{i=1}^n 1}{\sigma^2} = -\frac{n}{\sigma^2}.$$

It follows that $\mathcal{I}(\hat{\mu}) = n/\sigma^2$ and $s_{\hat{\mu}} = \sigma/\sqrt{n}$. Since the MLE of μ is \bar{Y} , this result is the familiar fact that when σ is known, then $s_{\bar{Y}} = \sigma/\sqrt{n}$ and when σ is unknown, then $s_{\bar{Y}} = s/\sqrt{n}$. \square

The theory justifying using these standard errors is the central limit theorem for the maximum likelihood estimator. This theorem can be stated

in a mathematically precise manner that is difficult to understand without advanced probability theory. The following less precise statement is more easily understood:

Result 5.1 Under suitable assumptions, for large enough sample sizes, the maximum likelihood estimator is approximately normally distributed with mean equal to the true parameter and with variance equal to the inverse of the Fisher information.

The central limit theorem for the maximum likelihood estimator justifies the following large-sample confidence interval for the MLE of θ :

$$\hat{\theta} \pm s_{\hat{\theta}} z_{\alpha/2}, \quad (5.20)$$

where $z_{\alpha/2}$ is the $\alpha/2$ -upper quantile of the normal distribution and $s_{\hat{\theta}}$ is defined in (5.19).

The observed Fisher information is

$$\mathcal{I}^{\text{obs}}(\theta) = -\frac{d^2}{d\theta^2} \log\{L(\theta)\}, \quad (5.21)$$

which differs from (5.18) in that there is no expectation taken. In many examples, (5.21) is a sum of many independent terms and, by the law of large numbers, will be close to (5.18). The expectation in (5.18) may be difficult to compute and using (5.21) instead is a convenient alternative.

The standard error of $\hat{\theta}$ based on observed Fisher information is

$$s_{\hat{\theta}}^{\text{obs}} = \frac{1}{\sqrt{\mathcal{I}^{\text{obs}}(\hat{\theta})}}. \quad (5.22)$$

Often $s_{\hat{\theta}}^{\text{obs}}$ is used in place of $s_{\hat{\theta}}$ in the confidence interval (5.20). There is theory suggesting that using the observed Fisher information will result in a more accurate confidence interval, that is, an interval with the true coverage probability closer to the nominal value of $1-\alpha$, so observed Fisher information can be justified by more than mere convenience; see Sect. 5.18.

So far, it has been assumed that θ is one-dimensional. In the multivariate case, the second derivative in (5.18) is replaced by the Hessian matrix of second derivatives,¹¹ and the result is called the *Fisher information matrix*. Analogously, the observed Fisher information matrix is the multivariate analog of (5.21). The covariance matrix of the MLE can be estimated by the inverse of the observed Fisher information matrix. If the negative of the log-likelihood is minimized by the R function `optim()`, then the observed Fisher information matrix is computed numerically and returned if `hessian = TRUE`

¹¹ The Hessian matrix of a function $f(x_1, \dots, x_m)$ of m variables is the $m \times m$ matrix whose i, j th entry is the second partial derivative of f with respect to x_i and x_j .

in the call to this function. See Example 5.3 for an example where standard errors of the MLEs are computed numerically. Fisher information matrices are discussed in more detail in Sect. 7.10.

Bias and Standard Deviation of the MLE

In many examples, the MLE has a small bias that decreases to 0 at rate n^{-1} as the sample size n increases to ∞ . More precisely,

$$\text{BIAS}(\hat{\theta}_{\text{ML}}) = E(\hat{\theta}_{\text{ML}}) - \theta \sim \frac{A}{n}, \text{ as } n \rightarrow \infty, \quad (5.23)$$

for some constant A . The bias of the MLE of a normal variance is an example and $A = -\sigma^2$ in this case.

Although this bias can be corrected in some special problems, such as estimation of a normal variance, usually the bias is ignored. There are two good reasons for this. First, the log-likelihood usually is the sum of n terms and so grows at rate n . The same is true of the Fisher information. Therefore, the variance of the MLE decreases at rate n^{-1} , that is,

$$\text{Var}(\hat{\theta}_{\text{ML}}) \sim \frac{B}{n}, \text{ as } n \rightarrow \infty, \quad (5.24)$$

for some $B > 0$. Variability should be measured by the standard deviation, not the variance, and by (5.24),

$$\text{SD}(\hat{\theta}_{\text{ML}}) \sim \frac{\sqrt{B}}{\sqrt{n}}, \text{ as } n \rightarrow \infty. \quad (5.25)$$

The convergence rate in (5.25) can also be obtained from the CLT for the MLE. Comparing (5.23) and (5.25), one sees that as n gets larger, the bias of the MLE becomes negligible compared to the standard deviation. This is especially important with financial markets data, where sample sizes tend to be large.

Second, even if the MLE of a parameter θ is unbiased, the same is not true for a nonlinear function of θ . For example, even if $\hat{\sigma}^2$ is unbiased for σ^2 , $\hat{\sigma}$ is biased for σ . The reason for this is that for a nonlinear function g , in general,

$$E\{g(\hat{\theta})\} \neq g\{E(\hat{\theta})\}.$$

Therefore, it is impossible to correct for all biases.

5.11 Likelihood Ratio Tests

Some readers may wish to review hypothesis testing by reading Appendix A.18 before starting this section.

Likelihood ratio tests, like maximum likelihood estimation, are based upon the likelihood function. Both are convenient, all-purpose tools that are widely used in practice.

Suppose that $\boldsymbol{\theta}$ is a parameter vector and that the null hypothesis puts m equality constraints on $\boldsymbol{\theta}$. More precisely, there are m functions g_1, \dots, g_m and the null hypothesis is that $g_i(\boldsymbol{\theta}) = 0$ for $i = 1, \dots, m$. The models without and with the constraints are called the full and reduced models, respectively.

It is also assumed that none of these constraints is redundant, that is, implied by the others. To illustrate redundancy, suppose that $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ and the constraints are $\theta_1 = \theta_2$, $\theta_2 = \theta_3$, and $\theta_1 = \theta_3$. Then the constraints have a redundancy since any two of them imply the third. Thus, $m = 2$, not 3.

Of course, redundancies need not be so easy to detect. One way to check is that the $m \times \dim(\boldsymbol{\theta})$ matrix

$$\begin{pmatrix} \nabla g_1(\boldsymbol{\theta}) \\ \vdots \\ \nabla g_m(\boldsymbol{\theta}) \end{pmatrix} \quad (5.26)$$

must have rank m . Here $\nabla g_i(\boldsymbol{\theta})$ is the gradient of g_i .

As an example, one might want to test that a population mean is zero; then $\boldsymbol{\theta} = (\mu, \sigma)^\top$ and $m = 1$ since the null hypothesis puts one constraint on $\boldsymbol{\theta}$, specifically that $\mu = 0$.

Let $\hat{\boldsymbol{\theta}}_{\text{ML}}$ be the maximum likelihood estimator without restrictions and let $\hat{\boldsymbol{\theta}}_{0,\text{ML}}$ be the value of $\boldsymbol{\theta}$ that maximizes $L(\boldsymbol{\theta})$ subject to the restrictions of the null hypothesis. If H_0 is true, then $\hat{\boldsymbol{\theta}}_{0,\text{ML}}$ and $\hat{\boldsymbol{\theta}}_{\text{ML}}$ should both be close to $\boldsymbol{\theta}$ and therefore $L(\hat{\boldsymbol{\theta}}_{0,\text{ML}})$ should be similar to $L(\hat{\boldsymbol{\theta}}_{\text{ML}})$. If H_0 is false, then the constraints will keep $\hat{\boldsymbol{\theta}}_{0,\text{ML}}$ far from $\hat{\boldsymbol{\theta}}_{\text{ML}}$ and so $L(\hat{\boldsymbol{\theta}}_{0,\text{ML}})$ should be noticeably smaller than $L(\hat{\boldsymbol{\theta}}_{\text{ML}})$.

The likelihood ratio test rejects H_0 if

$$2 \left[\log\{L(\hat{\boldsymbol{\theta}}_{\text{ML}})\} - \log\{L(\hat{\boldsymbol{\theta}}_{0,\text{ML}})\} \right] \geq c, \quad (5.27)$$

where c is a critical value. The left-hand side of (5.27) is twice the log of the likelihood ratio $L(\hat{\boldsymbol{\theta}}_{\text{ML}})/L(\hat{\boldsymbol{\theta}}_{0,\text{ML}})$, hence the name likelihood ratio test. Often, an *exact critical value* can be found. A critical value is exact if it gives a level that is exactly equal to α . When an exact critical value is unknown, then the usual choice of the critical value is

$$c = \chi^2_{\alpha,m}, \quad (5.28)$$

where, as defined in Appendix A.10.1, $\chi^2_{\alpha,m}$ is the α -upper quantile value of the chi-squared distribution with m degrees of freedom.¹² The critical value (5.28)

¹² The reader should now appreciate why it is essential to calculate m correctly by eliminating redundant constraints. The wrong value of m will cause an incorrect critical value to be used.

is only approximate and uses the fact that under the null hypothesis, as the sample size increases the distribution of twice the log-likelihood ratio converges to the chi-squared distribution with m degrees of freedom if certain assumptions hold. One of these assumptions is that the null hypothesis is *not* on the boundary of the parameter space. For example, if the null hypothesis is that a variance parameter is zero, then the null hypothesis is on the boundary of the parameter space since a variance must be zero or greater. In this case (5.27) should not be used; see Self and Liang (1987). Also, if the sample size is small, then the large-sample approximation (5.27) is suspect and should be used with caution. An alternative is to use the bootstrap to determine the rejection region. The bootstrap is discussed in Chap. 6.

Computation of likelihood ratio tests is often very simple. In some cases, the test is computed automatically by statistical software. In other cases, software will compute the log-likelihood for each model (full and reduced) and these can be plugged into the left-hand side of (5.27).

5.12 AIC and BIC

An important practical problem is choosing between two or more statistical models that might be appropriate for a data set. The maximized value of the log-likelihood, denoted here by $\log\{L(\hat{\boldsymbol{\theta}}_{ML})\}$, can be used to measure how well a model fits the data or to compare the fits of two or more models. However, $\log\{L(\hat{\boldsymbol{\theta}}_{ML})\}$ can be increased simply by adding parameters to the model. The additional parameters do not necessarily mean that the model is a better description of the data-generating mechanism, because the additional model complexity due to added parameters may simply be fitting random noise in the data, a problem that is called *overfitting*. Therefore, models should be compared both by fit to the data and by model complexity. To find a parsimonious model one needs a good tradeoff between maximizing fit and minimizing model complexity.

AIC (Akaike's information criterion) and BIC (Bayesian information criterion) are two means for achieving a good tradeoff between fit and complexity. They differ slightly and BIC seeks a somewhat simpler model than AIC. They are defined by

$$\text{AIC} = -2 \log\{L(\hat{\boldsymbol{\theta}}_{ML})\} + 2p \quad (5.29)$$

$$\text{BIC} = -2 \log\{L(\hat{\boldsymbol{\theta}}_{ML})\} + \log(n)p, \quad (5.30)$$

where p equals the number of parameters in the model and n is the sample size. For both criteria, “smaller is better,” since small values tend to maximize $L(\hat{\boldsymbol{\theta}}_{ML})$ ($\text{minimize } -\log\{L(\hat{\boldsymbol{\theta}}_{ML})\}$) and minimize p , which measures model complexity. The terms $2p$ and $\log(n)p$ are called “complexity penalties” since they penalize larger models.

The term *deviance* is often used for minus twice the log-likelihood, so $AIC = \text{deviance} + 2p$ and $BIC = \text{deviance} + \log(n)p$. Deviance quantifies model fit, with smaller values implying better fit.

Generally, from a group of candidate models, one selects the model that minimizes whichever criterion, AIC or BIC, is being used. However, any model that is within 2 or 3 of the minimum value is a good candidate and might be selected instead, for example, because it is simpler or more convenient to use than the model achieving the absolute minimum. Since $\log(n) > 2$ provided, as is typical, that $n > 8$, BIC penalizes model complexity more than AIC does, and for this reason BIC tends to select simpler models than AIC. However, it is common for both criteria to select the same, or nearly the same, model. Of course, if several candidate models all have the same value of p , then AIC, BIC, and $-2 \log\{L(\hat{\theta}_{ML})\}$ are minimized by the same model.

5.13 Validation Data and Cross-Validation

When the same data are used both to estimate parameters and to assess fit, there is a strong tendency towards overfitting. Data contain both a *signal* and *noise*. The signal contains characteristics that are present in the population and therefore in each sample from the population, but the noise is random and varies from sample to sample. *Overfitting* means selecting an unnecessarily complex model to fit the noise. The obvious remedy to overfitting is to diagnose model fit using data that are independent of the data used for parameter estimation. We will call the data used for estimation the *training data* and the data used to assess fit the *validation data* or *test data*.

Example 5.2. Estimating the expected returns of midcap stocks

This example uses 500 daily returns on 20 midcap stocks in the file `midcapD.ts.csv` on the book's web site. The data were originally in the `midcapD.ts` data set in R's `fEcofin` package. The data are from 28-Feb-91 to 29-Dec-95. Suppose we need to estimate the 20 expected returns. Consider two estimators. The first, called "separate-means," is simply the 20 sample means. The second, "common-mean," uses the average of the 20 sample means as the common estimator of all 20 expected returns.

The rationale behind the common-mean estimator is that midcap stocks should have similar expected returns. The common-mean estimator pools data and greatly reduces the variance of the estimator. The common-mean estimator has some bias because the true expected returns will not be identical, which is the requirement for unbiasedness of the common-mean estimator. The separate-means estimator is unbiased but at the expense of a higher variance. This is a classic example of a bias-variance tradeoff.

Which estimator achieves the best tradeoff? To address this question, the data were divided into the returns for the first 250 days (training data) and for the last 250 days (validation data). The criterion for assessing goodness-of-fit was the sum of squared errors, which is

$$\sum_{k=1}^{20} \left(\hat{\mu}_k^{\text{train}} - \bar{Y}_k^{\text{val}} \right)^2,$$

where $\hat{\mu}_k^{\text{train}}$ is the estimator (using the training data) of the k th expected return and \bar{Y}_k^{val} is the validation data sample mean of the returns on the k th stock. The sum of squared errors are 3.262 and 0.898, respectively, for the separate-means and common-mean estimators. The conclusion, of course, is that in this example the common-mean estimator is much more accurate than using separate means.

Suppose we had used the training data also for validation? The goodness-of-fit criterion would have been

$$\sum_{k=1}^{20} \left(\hat{\mu}_k^{\text{train}} - \bar{Y}_k^{\text{train}} \right)^2,$$

where \bar{Y}_k^{train} is the training data sample mean for the k th stock and is also the separate-means estimator for that stock. What would the results have been? Trivially, the sum of squared errors for the separate-means estimator would have been 0—each mean is estimated by itself with perfect accuracy! The common-mean estimator has a sum of squared errors equal to 0.920. The inappropriate use of the training data for validation would have led to the erroneous conclusion that the separate-means estimator is more accurate.

There are compromises between the two extremes of a common mean and separate means. These compromise estimators shrink the separate means toward the common mean. Bayesian estimation, discussed in Chap. 20, is an effective method for selecting the amount of shrinkage; see Example 20.12, where this set of returns is analyzed further. \square

A common criterion for judging fit is the deviance, which is -2 times the log-likelihood. The deviance of the validation data is

$$-2 \log f(\mathbf{Y}^{\text{val}} | \hat{\boldsymbol{\theta}}^{\text{train}}), \quad (5.31)$$

where $\hat{\boldsymbol{\theta}}^{\text{train}}$ is the MLE of the training data, \mathbf{Y}^{val} is the validation data, and $f(\mathbf{y}^{\text{val}} | \boldsymbol{\theta})$ is the density of the validation data.

When the sample size is small, splitting the data once into training and validation data is wasteful. A better technique is *cross-validation*, often called simply CV, where each observation gets to play both roles, training and validation. K -fold cross-validation divides the data set into K subsets of roughly

equal size. Validation is done K times. In the k th validation, $k = 1, \dots, K$, the k th subset is the validation data and the other $K - 1$ subsets are combined to form the training data. The K estimates of goodness-of-fit are combined, for example, by averaging them. A common choice is n -fold cross-validation, also called *leave-one-out* cross-validation. With leave-one-out cross-validation, each observation takes a turn at being the validation data set, with the other $n - 1$ observations as the training data.

An alternative to actually using validation data is to calculate what would happen if new data could be obtained and used for validation. This is how AIC was derived. AIC is an approximation to the expected deviance of a hypothetical new sample that is independent of the actual data. More precisely, AIC approximates

$$E \left[-2 \log f \left\{ \mathbf{Y}^{\text{new}} \mid \widehat{\boldsymbol{\theta}}(\mathbf{Y}^{\text{obs}}) \right\} \right], \quad (5.32)$$

where \mathbf{Y}^{obs} is the observed data, $\widehat{\boldsymbol{\theta}}(\mathbf{Y}^{\text{obs}})$ is the MLE computed from \mathbf{Y}^{obs} , and \mathbf{Y}^{new} is a hypothetical new data set such that \mathbf{Y}^{obs} and \mathbf{Y}^{new} are i.i.d. Stated differently, \mathbf{Y}^{new} is an unobserved independent replicate of \mathbf{Y}^{obs} . Since \mathbf{Y}^{new} is not observed but has the same distribution as \mathbf{Y}^{obs} , to obtain AIC one substitutes \mathbf{Y}^{obs} for \mathbf{Y}^{new} in (5.32) and omits the expectation in (5.32). Then one calculates the effect of this substitution. The approximate effect is to reduce (5.32) by twice the number of parameters. Therefore, AIC compensates by adding $2p$ to the deviance, so that

$$\text{AIC} = -2 \log f \left\{ \mathbf{Y}^{\text{obs}} \mid \widehat{\boldsymbol{\theta}}(\mathbf{Y}^{\text{obs}}) \right\} + 2p, \quad (5.33)$$

which is a reexpression of (5.29).

The approximation used in AIC becomes more accurate when the sample size increases. A small-sample correction to AIC is

$$\text{AIC}_c = \text{AIC} + \frac{2p(p+1)}{n-p-1}. \quad (5.34)$$

Financial markets data sets are often large enough that the correction term $2p(p+1)/(n-p-1)$ is small, so that AIC is adequate and AIC_c is not needed. For example, if $n = 200$, then $2p(p+1)/(n-p-1)$ is 0.12, 0.21, 0.31, and 0.44 and for $p = 3, 4, 5$, and 6, respectively. Since a difference less than 1 in AIC values is usually considered inconsequential, the correction would have little effect when comparing models with 3 to 6 parameters when n is at least 200. Even more dramatically, when n is 500, then the corrections for 3, 4, 5, and 6 parameters are only 0.05, 0.08, 0.12, and 0.17.

Traders often develop trading strategies using a set of historical data and then test the strategies on new data. This is called *back-testing* and is a form of validation.

5.14 Fitting Distributions by Maximum Likelihood

As mentioned previously, one can find a formula for the MLE only for a few “textbook” examples. In most cases, the MLE must be found numerically. As an example, suppose that Y_1, \dots, Y_n is an i.i.d. sample from a t -distribution. Let

$$f_{t,\nu}^{\text{std}}(y | \mu, \sigma) \quad (5.35)$$

be the density of the standardized t -distribution with ν degrees of freedom and with mean μ and standard deviation σ . Then the parameters ν , μ , and σ are estimated by maximizing

$$\sum_{i=1}^n \log \left\{ f_{t,\nu}^{\text{std}}(Y_i | \mu, \sigma) \right\} \quad (5.36)$$

using any convenient optimization software. Estimation of other models is similar.

In the following examples, t -distributions and generalized error distributions are fit.

Example 5.3. Fitting a t -distribution to changes in risk-free returns

This example uses one of the time series in Chap. 4, the changes in the risk-free returns that has been called `diffrf`. This time series will be used to illustrate several methods for fitting a t -distribution. The simplest method uses the R function `fitdistr()`.

```
data(Capm, package = "Ecdat")
x = diff(Capm$rf)
fitdistr(x, "t")
```

The output is:

```
> fitdistr(x, "t")
      m           s          df
  0.0012243   0.0458549   3.3367036
(0.0024539) (0.0024580) (0.5000096)
```

The parameters, in order, are the mean, the scale parameter, and the degrees of freedom. The numbers in parentheses are the standard errors.

Next, we fit the t -distribution by writing a function to return the negative log-likelihood and using R’s `optim()` function to minimize the log-likelihood. We compute standard errors by using `solve()` to invert the Hessian and then taking the square roots of the diagonal elements of the inverted Hessian. We also compute AIC and BIC.

```

library(fGarch)
n = length(x)
start = c(mean(x), sd(x), 5)
loglik_t = function(beta) sum( - dt((x - beta[1]) / beta[2],
beta[3], log = TRUE) + log(beta[2]) )
fit_t = optim(start, loglik_t, hessian = T,
method = "L-BFGS-B", lower = c(-1, 0.001, 1))
AIC_t = 2 * fit_t$value + 2 * 3
BIC_t = 2 * fit_t$value + log(n) * 3
sd_t = sqrt(diag(solve(fit_t$hessian)))
fit_t$par
sd_t
AIC_t
BIC_t

```

The results are below. The estimates and the standard errors agree with those produced by `fitdistr()`, except for small numerical errors.

```

> fit_t$par
[1] 0.00122 0.04586 3.33655
> sd_t
[1] 0.00245 0.00246 0.49982
> AIC_t
[1] -1380.4
> BIC_t
[1] -1367.6

```

The standardized t -distribution can be fit by changing `dt()` to `dstd()`. Then the parameters are the mean, standard deviation, and degrees of freedom.

```

loglik_std = function(beta) sum(- dstd(x, mean = beta[1],
sd = beta[2], nu = beta[3], log = TRUE))
fit_std = optim(start, loglik_std, hessian = T,
method = "L-BFGS-B", lower = c(-0.1, 0.01, 2.1))
AIC_std = 2*fit_std$value + 2 * 3
BIC_std = 2*fit_std$value + log(n) * 3
sd_std = sqrt(diag(solve(fit_std$hessian)))
fit_std$par
sd_std
AIC_std
BIC_std

```

The results are below. The estimates agree with those when using `dt()` since $0.0725 = 0.0459\sqrt{3.33/(3.33 - 2)}$, aside from numerical error. Notice that AIC and BIC are unchanged, as expected since we are fitting the same model as before and only changing the parameterization.

```
> fit_std$par
[1] 0.0012144 0.0725088 3.3316132
> sd_std
[1] 0.0024538 0.0065504 0.4986456
> AIC_std
[1] -1380.4
> BIC_std
[1] -1367.6
```

□

Example 5.4. Fitting an F-S skewed t -distribution to changes in risk-free returns

Next, we fit the F-S skewed t -distribution.

```
loglik_sstd = function(beta) sum(- dsstd(x, mean = beta[1],
  sd = beta[2], nu = beta[3], xi = beta[4], log = TRUE))
start = c(mean(x), sd(x), 5, 1)
fit_sstd = optim(start, loglik_sstd, hessian = T,
  method = "L-BFGS-B", lower = c(-0.1, 0.01, 2.1, -2))
AIC_sstd = 2*fit_sstd$value + 2 * 4
BIC_sstd = 2*fit_sstd$value + log(n) * 4
sd_sstd = sqrt(diag(solve(fit_sstd$hessian)))
fit_sstd$par
sd_sstd
AIC_sstd
BIC_sstd
```

The results are below. The estimate of ξ (the fourth parameter) is very close to 1, which corresponds to the usual t -distribution. Both AIC and BIC increase since the extra skewness parameter does not improve the fit but adds 1 to the number of parameters.

```
> fit_sstd$par
[1] 0.0011811 0.0724833 3.3342759 0.9988491
> sd_sstd
[1] 0.0029956 0.0065790 0.5057846 0.0643003
> AIC_sstd
[1] -1378.4
> BIC_sstd
[1] -1361.4
```

□

Example 5.5. Fitting a generalized error distribution to changes in risk-free returns

The fit of the generalized error distribution to `diffrrf` was obtained using `optim()` similarly to the previous example.

```
> fit_ged$par
[1] -0.00019493  0.06883004  1.00006805
> sd_ged
[1] 0.0011470  0.0033032  0.0761374
> AIC_ged
[1] -1361.4
> BIC_ged
[1] -1344.4
```

The three parameters are the estimates of the mean, standard deviation, and the shape parameter ν , respectively. The estimated shape parameter is extremely close to 1, implying a double-exponential distribution. Note that AIC and BIC are considerably larger than for the t -distribution. Therefore, t -distributions appear to be better models for these data compared to generalized error distributions. A possible reason for this is that, like the t -distributions, the density of the data seems to be rounded near the median; see the kernel density estimate in Fig. 5.9. QQ plots in Fig. 5.10 of `diffrrf` versus the quantiles of the fitted t - and generalized error distributions are similar, indicating that neither model has a decidedly better fit than the other. However, the QQ plot of the t -distribution is slightly more linear. \square

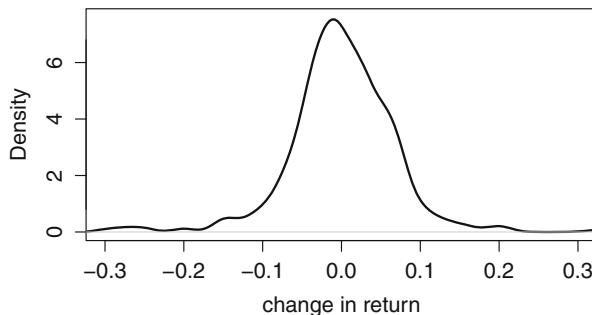


Fig. 5.9. Kernel estimate of the probability density of `diffrrf`, the changes in the risk-free returns.

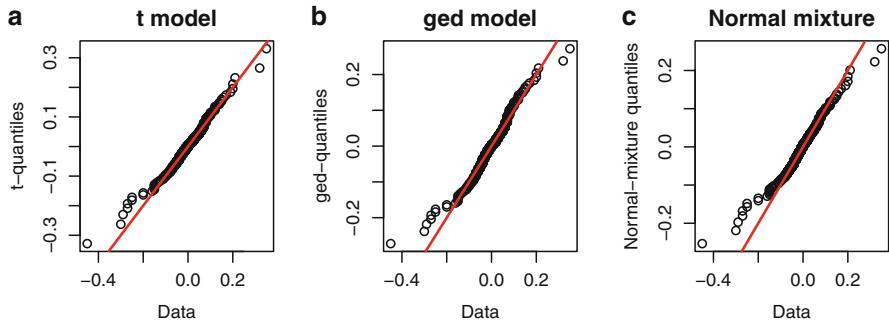


Fig. 5.10. (a) QQ plot of `diffrrf` versus the quantiles of a $t_{\nu}^{\text{std}}(\mu, s^2)$ distribution with μ , s^2 , and ν estimated by maximum likelihood. A 45° line through the origin has been added for reference. (b) A similar plot for the generalized error distribution. (c) A similar plot for a normal mixture model.

Example 5.6. A-C skewed t -distribution fit to pipeline flows

This example uses the daily flows in natural gas pipelines introduced in Example 4.2. Recall that all three distributions are left-skewed. There are many well-known parametric families of right-skewed distributions, such as, the gamma and log-normal distributions, but there are not as many families of left-skewed distributions. The F-S skewed t - and A-C skewed t -distributions, which contain both right- and left-skewed distributions, are important exceptions. In this example, the A-C skewed normal distributions will be used.

Figure 5.11 has one row of plots for each variable. The left plots have two density estimates, an estimate using the Azzalini–Capitanio skewed normal distribution (solid) and a KDE (dashed). The right plots are QQ plots using the fitted skewed normal distributions.

The flows in pipelines 1 and, to a lesser extent, 2 are fit reasonably well by the A-C skewed normal distribution. This can be seen in the agreement between the parametric density estimates and the KDEs and in the nearly straight patterns in the QQ plots. The flows in pipeline 3 have a KDE with either a wide, flat mode or, perhaps, two modes. This pattern cannot be accommodated very well by the A-C skewed normal distributions. The result is less agreement between the parametric and KDE fits and a curved QQ plot. Nonetheless, a skewed normal distribution might be an adequate approximation for some purposes.

The following code produced the top row of Fig. 5.11. The code for the remaining rows is similar. The function `sn.mple()` at line 7 computed the MLEs using the CD parametrization and the function `cp2dp()` at line 8 converted the MLEs to the DP parametrization, which is used by the functions `dsn()` and `qsn()` at lines 9 and 18 that were needed in the plots. The red reference line through the quartiles in the QQ plot is created at lines 20–22.

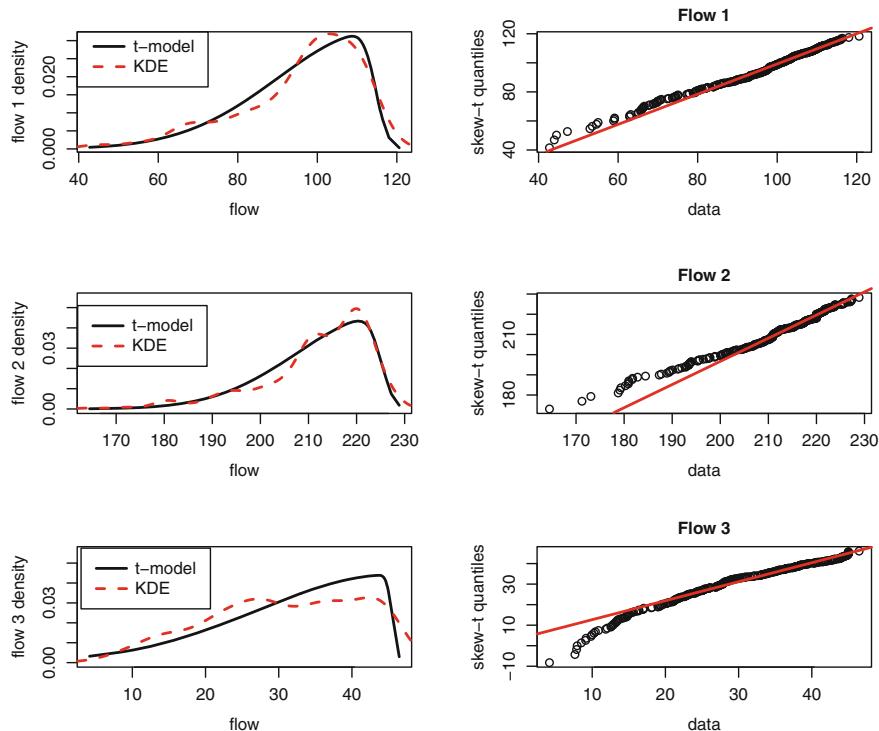


Fig. 5.11. Parametric (solid) and nonparametric (dashed) density estimates for daily flows in three pipelines (left) and QQ plots for the parametric fits (right). The reference lines go through the first and third quartiles.

```

1 library(sn)
2 dat = read.csv("FlowData.csv")
3 dat = dat/10000
4 par(mfrow = c(3, 2))
5 x = dat$Flow1
6 x1 = sort(x)
7 fit1 = sn.mple(y = x1, x = as.matrix(rep(1, length(x1))))
8 est1 = cp2dp(fit1$cp, family = "SN")
9 plot(x1, dsn(x1, dp = est1),
10       type = "l", lwd = 2, xlab = "flow",
11             ylab = "flow 1 density")
12 d = density(x1)
13 lines(d$x, d$y, lty = 2, lwd = 2)
14 legend(40, 0.034, c("t-model", "KDE"), lty = c(1, 2),
15         lwd = c(2, 2))
16 n = length(x1)
17 u=(1:n) / (n + 1)
18 plot(x1, qsn(u, dp = est1), xlab = "data",

```

```

19   ylab = "skew-t quantiles", main = "Flow 1")
20 lmfit = lm(qsn(c(0.25, 0.75), dp = est1) ~ quantile(x1,
21   c(0.25, 0.75)) )
22 abline(lmfit)

```

□

5.15 Profile Likelihood

Profile likelihood is a technique based on the likelihood ratio test introduced in Sect. 5.11. Profile likelihood is used to create confidence intervals and is often a convenient way to find a maximum likelihood estimator. Suppose the parameter vector is $\boldsymbol{\theta} = (\theta_1, \boldsymbol{\theta}_2)$, where θ_1 is a scalar parameter and the vector $\boldsymbol{\theta}_2$ contains the other parameters in the model. The profile log-likelihood for θ_1 is

$$L_{\max}(\theta_1) = \max_{\boldsymbol{\theta}_2} L(\theta_1, \boldsymbol{\theta}_2). \quad (5.37)$$

The right-hand side of (5.37) means the $L(\theta_1, \boldsymbol{\theta}_2)$ is maximized over $\boldsymbol{\theta}_2$ with θ_1 fixed to create a function of θ_1 only. Define $\widehat{\boldsymbol{\theta}}_2(\theta_1)$ as the value of $\boldsymbol{\theta}_2$ that maximizes the right-hand side of (5.37).

The MLE of θ_1 is the value, $\widehat{\theta}_1$, that maximizes $L_{\max}(\theta_1)$ and the MLE of $\boldsymbol{\theta}_2$ is $\widehat{\boldsymbol{\theta}}_2(\widehat{\theta}_1)$. Let $\theta_{0,1}$ be a hypothesized value of θ_1 . By the theory of likelihood ratio tests in Sect. 5.11, one accepts the null hypothesis $H_0 : \theta_1 = \theta_{0,1}$ if

$$L_{\max}(\theta_{0,1}) > L_{\max}(\widehat{\theta}_1) - \frac{1}{2}\chi^2_{\alpha,1}. \quad (5.38)$$

Here $\chi^2_{\alpha,1}$ is the α -upper quantile of the chi-squared distribution with one degree of freedom. The profile likelihood confidence interval (or, more properly, confidence region since it need not be an interval) for θ_1 is the set of all null values that would be accepted, that is,

$$\left\{ \theta_1 : L_{\max}(\theta_1) > L_{\max}(\widehat{\theta}_1) - \frac{1}{2}\chi^2_{\alpha,1} \right\}. \quad (5.39)$$

The profile likelihood can be defined for a subset of the parameters, rather than for just a single parameter, but this topic will not be pursued here.

Example 5.7. Estimating a Box–Cox transformation

An automatic method for estimating the transformation parameter for a Box–Cox transformation¹³ assumes that for some values of α , μ , and σ , the transformed data $Y_1^{(\alpha)}, \dots, Y_n^{(\alpha)}$ are i.i.d. $N(\mu, \sigma^2)$ -distributed. All three

¹³ See Eq. (4.5).

parameters can be estimated by maximum likelihood. For a fixed value of α , $\hat{\mu}$ and $\hat{\sigma}$ are the sample mean and variance of $Y_1^{(\alpha)}, \dots, Y_n^{(\alpha)}$ and these values can be plugged into the log-likelihood to obtain the profile log-likelihood for α . This can be done with the function `boxcox()` in R's MASS package, which plots the profile log-likelihood with confidence intervals.

Estimating α by the use of profile likelihood will be illustrated using the data on gas pipeline flows. Figure 5.12 shows the profile log-likelihoods and the KDEs and normal QQ plots of the flows transformed using the MLE of α .

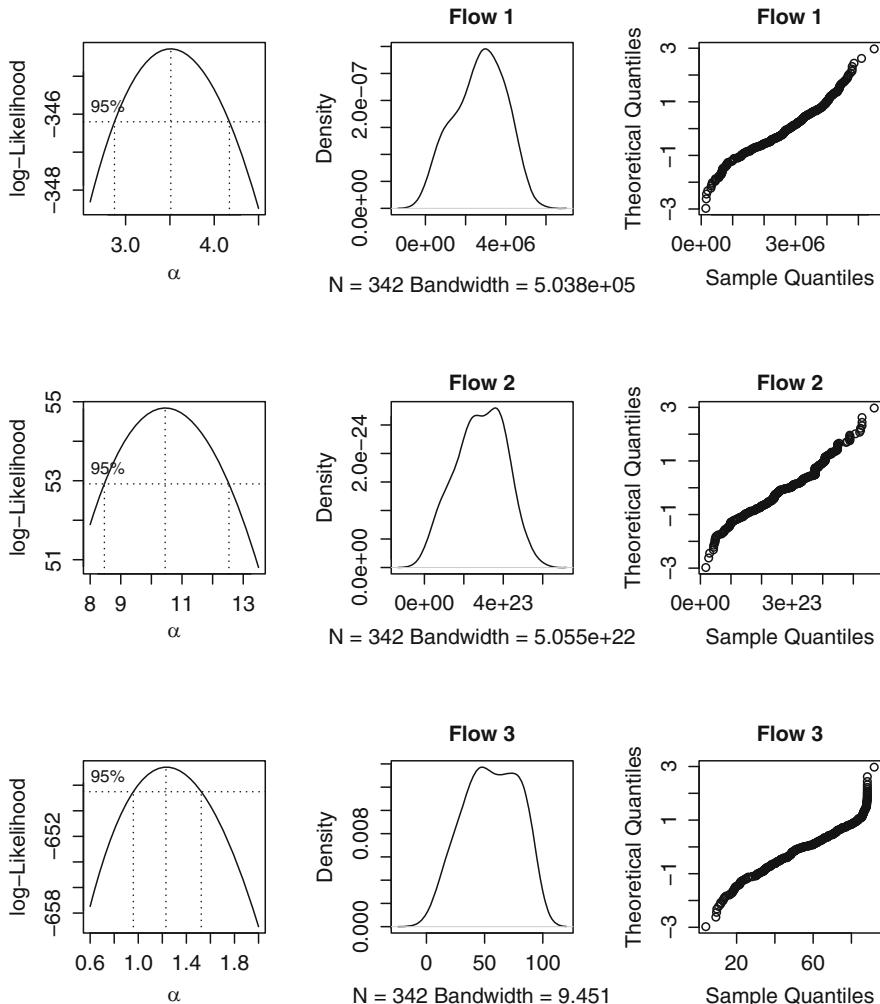


Fig. 5.12. Profile log-likelihoods and 95 % confidence intervals for the parameter α of the Box–Cox transformation (left), KDEs of the transformed data (middle column), and normal plots of the transformed data (right).

The KDE used `adjust = 1.5` to smooth out local bumpiness seen with the default bandwidth. For the flows in pipeline 1, the MLE is $\hat{\alpha} = 3.5$. Recall that in Example 4.2, we saw by trial-and-error that α between 3 and 4 was best for symmetrizing the data. It is gratifying to see that maximum likelihood corroborates this choice. The QQ plots show that the Box–Cox transformed flows have light tails. Light tails are not usually considered to be a problem and are to be expected here since the pipeline flows are bounded, below by 0 and above by the capacity of the pipeline.

The top row of Fig. 5.12 was produced by the following code. The function `boxcox()` at line 8 created the top-left plot containing the profile likelihood of the transformation parameter.

```

1 dat = read.csv("FlowData.csv")
2 dat = dat / 10000
3 library("MASS") #####  for boxcox()
4 adj = 1.5
5 par(mfrow = c(3, 3))
6 x = dat$Flow1
7 x1 = sort(x)
8 bcfit1 = boxcox(x1 ~ 1, lambda = seq(2.6, 4.5, 1 / 100),
9      xlab = expression(alpha))
10 text(3, -1898.75, "Flow 1")
11 plot(density((x1^3.5 - 1) / 3.5, adjust = adj), main = "Flow 1")
12 qqnorm((x1^3.5 - 1) / 3.5, datax = TRUE, main = "Flow 1")

```

□

It is worth pointing out that we have now seen two distinct methods for accommodating the left skewness in the pipeline flows, modeling the untransformed data by a skewed t -distribution (Example 5.6) and Box–Cox transformation to a normal distribution (Example 5.7). A third method would be to forego parametric modeling and use the kernel density estimation. This is not an atypical situation; often data can be analyzed in several different, but equally appropriate, ways.

5.16 Robust Estimation

Although maximum likelihood estimators have many attractive properties, they have one serious drawback of which anyone using them should be aware. Maximum likelihood estimators can be very sensitive to the assumptions of the statistical model. For example, the MLE of the mean of a normal population is the sample mean and the MLE of σ^2 is the sample variance, except with the minor change of a divisor of n rather than $n - 1$. The sample mean and variance are efficient estimators when the population is truly normally distributed, but these estimators are very sensitive to outliers, especially the sample standard deviation. Because these estimators are averages of the data

and the squared deviations from the mean, respectively, a single outlier in the sample can drive the sample mean and variance to wildly absurd values if the outlier is far enough removed from the other data. Extreme outliers are nearly impossible with exactly normally distributed data, but if the data are only approximately normal with heavier tails than the normal distribution, then outliers are more probable and, when they do occur, more likely to be extreme. Therefore, the sample mean and variance can be very inefficient estimators. Statisticians say that the MLE is not *robust* to mild deviations from the assumed model. This is bad news and has led researchers to find estimators that are robust.

A robust alternative to the sample mean is the *trimmed mean*. An α -trimmed mean is computed by ordering the sample from smallest to largest, removing the fraction α of the smallest and the same fraction of the largest observations, and then taking the mean of the remaining observations. The idea behind trimming is simple and should be obvious: The sample is trimmed of extreme values before the mean is calculated. There is a mathematical formulation of the α -trimmed mean. Let $k = n\alpha$ rounded¹⁴ to an integer; k is the number of observations removed from both ends of the sample. Then the α -trimmed mean is

$$\bar{X}_\alpha = \frac{\sum_{i=k+1}^{n-k} Y_{(i)}}{n - 2k},$$

where $Y_{(i)}$ is the i th order statistic. Typical values of α are 0.1, 0.15, 0.2, and 0.25. As α approaches 0.5, the α -trimmed mean approaches the sample median, which is the 0.5-sample quantile.

Dispersion refers to the variation in a distribution or sample. The sample standard deviation is the most common estimate of dispersion, but as stated it is nonrobust. In fact, the sample standard deviation is even more nonrobust than the sample mean, because squaring makes outliers more extreme. A robust estimator of dispersion is the *MAD* (*median absolute deviation*) estimator, defined as

$$\hat{\sigma}^{\text{MAD}} = 1.4826 \times \text{median}\{|Y_i - \text{median}(Y_i)|\}. \quad (5.40)$$

This formula should be interpreted as follows. The expression “ $\text{median}(Y_i)$ ” is the sample median, $|Y_i - \text{median}(Y_i)|$ is the absolute deviation of the observations from their median, and $\text{median}\{|Y_i - \text{median}(Y_i)|\}$ is the median of these absolute deviations. For normally distributed data, the $\text{median}\{|Y_i - \text{median}(Y_i)|\}$ estimates not σ but rather $\Phi^{-1}(0.75)\sigma = \sigma/1.4826$, because for normally distributed data the $\text{median}\{|Y_i - \text{median}(Y_i)|\}$ will converge to $\sigma/1.4826$ as the sample size increases. Thus, the factor 1.4826 in Eq. (5.40) calibrates $\hat{\sigma}^{\text{MAD}}$ so that it estimates σ when applied to normally distributed data.

¹⁴ Definitions vary and the rounding could be either upward or to the nearest integer.

$\hat{\sigma}^{\text{MAD}}$ does not estimate σ for a nonnormal population. It does measure dispersion, but not dispersion as measured by the standard deviation. But this is just the point. For nonnormal populations the standard deviation can be very sensitive to the tails of the distribution and does not tell us much about the dispersion in the central range of the distribution, just in the tails.

In R, `mad()` computes (5.40). Some authors define MAD to be $\text{median}\{|Y_i - \text{median}(Y_i)|\}$, that is, without 1.4826. Here the notation $\hat{\sigma}^{\text{MAD}}$ is used to emphasize the standardization by 1.4826 in order to estimate a normal standard deviation.

An alternative to using robust estimators is to assume a model where outliers are more probable. Then the MLE will automatically downweight outliers. For example, the MLE of the parameters of a t -distribution is much more robust to outliers than the MLE of the parameters of a normal distribution.

5.17 Transformation Kernel Density Estimation with a Parametric Transformation

We saw in Sect. 4.8 that the transformation kernel density estimator (TKDE) can avoid the bumps seen when the ordinary KDE is applied to skewed data. The KDE also can exhibit bumps in the tails when both tails are long, as is common with financial markets data. An example is the variable `diffrf` whose KDE is in Fig. 5.9. For such data, the TKDE needs a transformation that is convex to the right of the mode and concave to the left of the mode. There are many such transformations, and in this section we will use some facts from probability theory, as well as maximum likelihood estimation, to select a suitable one.

The key ideas used here are that (1) normally distributed data have light tails and are suitable for estimation with the KDE, (2) it is easy to transform data to normality if one knows the CDF, and (3) the CDF can be estimated by assuming a parametric model and using maximum likelihood. If a random variable has a continuous distribution F , then $F(X)$ has a uniform distribution and $\Phi^{-1}\{F(X)\}$ has an $N(0, 1)$ distribution; here Φ is the standard normal CDF. Of course, in practice F is unknown, but one can estimate F parametrically, assuming, for example, that F is some t -distribution. It is not necessary that F actually be a t -distribution, only that a t -distribution can provide a reasonable enough fit to F in the tails so that an appropriate transformation is selected. If it was known that F was a t -distribution, then, of course, there would be no need to use a KDE or TKDE to estimate its density. The transformation to use in the TKDE is $g(y) = \Phi^{-1}\{F(y)\}$, which has inverse $g^{-1}(x) = F^{-1}\{\Phi(x)\}$. The derivative of g is needed to compute the TKDE and is

$$g'(y) = \frac{f(y)}{\phi[\Phi^{-1}\{F(y)\}]}. \quad (5.41)$$

Example 5.8. TKDE for risk-free returns

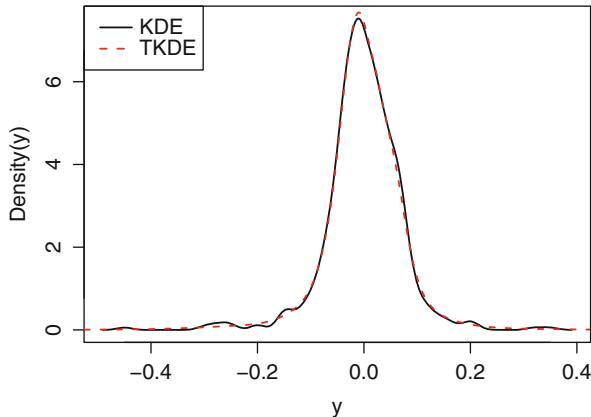


Fig. 5.13. Kernel density and transformation kernel density estimates of monthly changes in the risk-free returns, January 1960 to December 2002. The data are in the `Capm` series in the `Ecdat` package in R.

This example uses the changes in the risk-free returns in Fig. 4.3. We saw in Sect. 5.14 that these data are reasonably well fit by a t -distribution with mean, standard deviation, and ν equal to 0.00121, 0.0724, and 3.33, respectively. This distribution will be used as F . Figure 5.13 compares the ordinary KDE to the TKDE for this example. Notice that the TKDE is much smoother in the tails; this can be seen better in Fig. 5.14, which gives detail on the left tail.

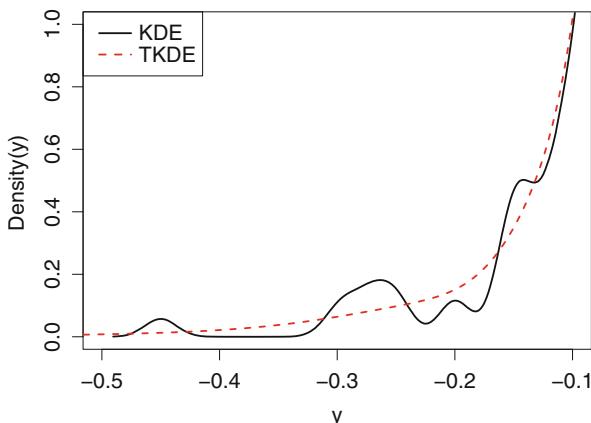


Fig. 5.14. Kernel density and transformation kernel density estimates of monthly changes in the risk-free returns, January 1960 to December 2002, zooming in on left tail.

The transformation used in this example is shown in Fig. 5.15. Notice the concave-convex shape that brings the left and right tails closer to the center and results in transformed data without the heavy tails seen in the original data. The removal of the heavy tails can be seen in Fig. 5.16, which is a normal plot of the transformed data.

The code to create Fig. 5.13 is below:

```

1 data(Capm, package = "Ecdat")
2 y = diff(Capm$rf)
3 diffrrf = y
4 library(fGarch)
5 x1 = pstd(y, mean = 0.001, sd = .0725, nu = 3.34)
6 x = qnorm(x1)

```

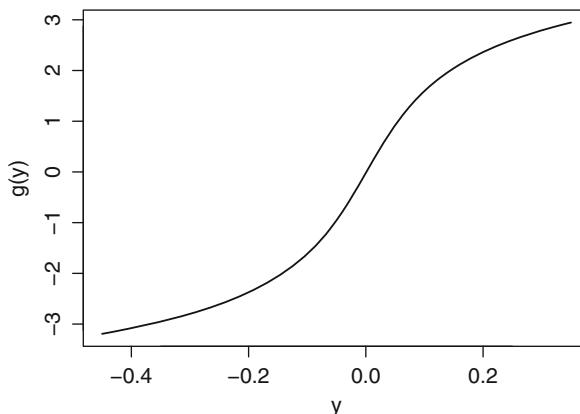


Fig. 5.15. Plot of the transformation used in Example 5.8.

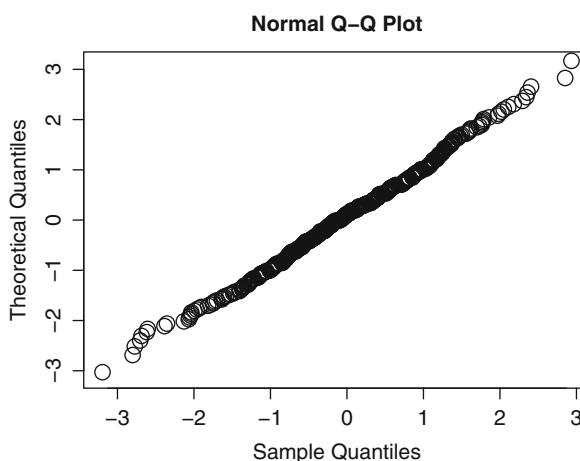


Fig. 5.16. Normal plot of the transformed data used in Example 5.8.

```

7 par(mfrow = c(1, 1))
8 d1 = density(diffrrf)
9 plot(d1$x, d1$y, type = "l", xlab = "y", ylab = "Density(y)",
10      lwd = 2)
11 d2 = density(x)
12 ginvx = qstd(pnorm(d2$x), mean = 0.001, sd = .0725, nu = 3.34)
13 gprime_num = dstd(ginvx, mean = 0.001, sd = .0725, nu = 3.34)
14 gprime_den = dnorm(qnorm(pstd(ginvx, mean = 0.001,
15      sd = .0725, nu = 3.34)))
16 gprime = gprime_num / gprime_den
17 lines(ginvx,d2$y * gprime, type = "l", lty = 2, col = "red", lwd = 2)
18 legend("topleft", c("KDE", "TKDE"), lty = c(1,2), lwd = 2,
19        col = c("black", "red"))

```

Lines 5–6 compute the transformation. Line 8 computes the KDE of the untransformed data and line 11 computes the KDE of the transformed data. Lines 12–16 compute g' in (5.41). At line 17 the KDE of the transformed data is multiplied by g' as in Eq. (4.6) to compute the TKDE. \square

5.18 Bibliographic Notes

Maximum likelihood estimation and likelihood ratio tests are discussed in all textbooks on mathematical statistics, including Boos and Stefanski (2013); Casella and Berger (2002), and Wasserman (2004).

Burnham and Anderson (2002) is a comprehensive introduction to model selection and is highly recommended for further reading. They also cover multimodel inference, a more advanced topic that includes *model averaging* where estimators or predictions are averaged across several models. Chapter 7 of Burnham and Anderson provides the statistical theory behind AIC as an approximate deviance of hypothetical validation data. The small-sample corrected AIC is due to Hurvich and Tsai (1989).

Buch-Larsen et al. (2005) and Ruppert and Wand (1992) discuss other methods for choosing the transformation when the TKDE is applied to heavy-tailed data.

The central limit theorem for the MLE is stated precisely and proved in textbooks on asymptotic theory such as Serfling (1980), van der Vaart (1998), and Lehmann (1999).

Observed and expected Fisher information are compared by Efron and Hinkley (1978), who argue that the observed Fisher information gives superior standard errors.

Box–Cox transformations were introduced by Box and Dox (1964). See Azzalini (2014); Azzalini and Capitanio (2003), and Arellano-Valle and Azzalini (2013) for discussion of the A-C skewed distributions.

5.19 R Lab

5.19.1 Earnings Data

Run the following R code to find a symmetrizing transformation for 1998 earnings data from the Current Population Survey. The code looks at the untransformed data and the square-root and log-transformed data. The transformed data are compared by normal plots, boxplots, and kernel density estimates.

```
library("Ecdat")
?CPSch3
data(CPSch3)
dimnames(CPSch3)[[2]]
```

```
male.earnings = CPSch3[CPSch3[,3] == "male", 2]
sqrt.male.earnings = sqrt(male.earnings)
log.male.earnings = log(male.earnings)

par(mfrow = c(2, 2))
qqnorm(male.earnings, datax = TRUE, main = "untransformed")
qqnorm(sqrt.male.earnings, datax = TRUE,
      main = "square-root transformed")
qqnorm(log.male.earnings, datax = TRUE, main = "log-transformed")

par(mfrow = c(2, 2))
boxplot(male.earnings, main = "untransformed")
boxplot(sqrt.male.earnings, main = "square-root transformed")
boxplot(log.male.earnings, main = "log-transformed")

par(mfrow = c(2,2))
plot(density(male.earnings), main = "untransformed")
plot(density(sqrt.male.earnings), main = "square-root transformed")
plot(density(log.male.earnings), main = "log-transformed")
```

Problem 1 Which of the three transformation provides the most symmetric distribution? Try other powers beside the square root. Which power do you think is best for symmetrization? You may include plots with your work if you find it helpful to do that.

Next, you will estimate the Box–Cox transformation parameter by maximum likelihood. The model is that the data are $N(\mu, \sigma^2)$ -distributed after being transformed by some λ . The unknown parameters are λ , μ , and σ .

Run the following R code to plot the profile likelihood for λ on the grid `seq(-2, 2, 1/10)` (this is the default and can be changed). The command `boxcox` takes an R formula as input. The left-hand side of the formula is the variable to be transformed. The right-hand side is a linear model (see Chap. 9). In this application, the model has only an intercept, which is indicated by

“1.” “MASS” is an acronym for “Modern Applied Statistics with S-PLUS,” a highly-regarded textbook whose fourth edition also covers R. The MASS library accompanies this book.

```
library("MASS")
par(mfrow = c(1, 1))
boxcox(male.earnings ~ 1)
```

The default grid of λ values is large, but you can zoom in on the high-likelihood region with the following:

```
boxcox(male.earnings ~ 1, lambda = seq(0.3, 0.45, 1 / 100))
```

To find the MLE, run this R code:

```
bc = boxcox(male.earnings ~ 1, lambda = seq(0.3, 0.45, by = 1 / 100),
            interp = FALSE)
ind = (bc$y == max(bc$y))
ind2 = (bc$y > max(bc$y) - qchisq(0.95, df = 1) / 2)
bc$x[ind]
bc$x[ind2]
```

- Problem 2** (a) What are `ind` and `ind2` and what purposes do they serve?
 (b) What is the effect of `interp` on the output from `boxcox`?
 (c) What is the MLE of λ ?
 (d) What is a 95 % confidence interval for λ ?
 (e) Modify the code to find a 99 % confidence interval for λ .

Rather than trying to transform the variable `male.earnings` to a Gaussian distribution, we could fit a skewed Gaussian or skewed t -distribution. R code that fits a skewed t is listed below:

```
library("fGarch")
fit = sstdFit(male.earnings, hessian = TRUE)
```

- Problem 3** What are the estimates of the degrees-of-freedom parameter and of ξ ?

- Problem 4** Produce a plot of a kernel density estimate of the pdf of `male.earnings`. Overlay a plot of the skewed t -density with MLEs of the parameters. Make sure that the two curves are clearly labeled, say with a legend, so that it is obvious which curve is which. Include your plot with your work. Compare the parametric and nonparametric estimates of the pdf. Do they seem similar? Based on the plots, do you believe that the skewed t -model provides an adequate fit to `male.earnings`?

Problem 5 Fit a skewed GED model to `male.earnings` and repeat Problem 4 using the skewed GED model in place of the skewed t. Which parametric model fits the variable `male.earnings` best, skewed t or skewed GED?

5.19.2 DAX Returns

This section uses log returns on the DAX index in the data set `EuStockMarkets`. Your first task is to fit the standardized t-distribution (std) to the log returns. This is accomplished with the following R code.

Here `loglik_std` is an R function that is defined in the code. This function returns minus the log-likelihood for the std model. The std density function is computed with the function `dstd` in the `fGarch` package. Minus the log-likelihood, which is called the objective function, is minimized by the function `optim`. The L-BFGS-B method is used because it allows us to place lower and upper bounds on the parameters. Doing this avoids the errors that would be produced if, for example, a variance parameter were negative. When `optim` is called, `start` is a vector of starting values. Use R's help to learn more about `optim`. In this example, `optim` returns an object `fit_std`. The component `fit_std$par` contains the MLEs and the component `fig_std$value` contains the minimum value of the objective function.

```
data(Garch, package = "Ecdat")
library("fGarch")
data(EuStockMarkets)
Y = diff(log(EuStockMarkets[,1])) # DAX

##### std #####
loglik_std = function(x) {
  f = -sum(dstd(Y, x[1], x[2], x[3], log = TRUE))
  f}
start = c(mean(Y), sd(Y), 4)
fit_std = optim(start, loglik_std, method = "L-BFGS-B",
  lower = c(-0.1, 0.001, 2.1),
  upper = c(0.1, 1, 20), hessian = TRUE)
cat("MLE =", round(fit_std$par, digits = 5))
minus_logL_std = fit_std$value # minus the log-likelihood
AIC_std = 2 * minus_logL_std + 2 * length(fit_std$par)
```

Problem 6 What are the MLEs of the mean, standard deviation, and the degrees-of-freedom parameter? What is the value of AIC?

Problem 7 Modify the code so that the MLEs for the skewed t-distribution are found. Include your modified code with your work. What are the MLEs? Which distribution is selected by AIC, the t or the skewed t-distribution?

Problem 8 Compute and plot the TKDE of the density of the log returns using the methodology in Sects. 4.8 and 5.17. The transformation that you use should be $g(y) = \Phi^{-1}\{F(y)\}$, where F is the t-distribution with parameters estimated in Problem 6. Include your code and the plot with your work.

Problem 9 Plot the KDE, TKDE, and parametric estimator of the log-return density, all on the same graph. Zoom in on the right tail, specifically the region $0.035 < y < 0.06$. Compare the three densities for smoothness. Are the TKDE and parametric estimates similar? Include the plot with your work.

Problem 10 Fit the F-S skewed t-distribution to the returns on the FTSE index in EuStockMarkets. Find the MLE, the standard errors of the MLE, and AIC.

5.19.3 McDonald's Returns

This section continues the analysis of McDonald's stock returns begun in Sect. 2.4.4 and continued in Sect. 4.10.2. Run the code below.

```

1 data = read.csv('MCD_PriceDaily.csv')
2 adjPrice = data[,7]
3 LogRet = diff(log(adjPrice))
4 library(MASS)
5 library(fGarch)
6 fit.T = fitdistr(LogRet, "t")
7 params.T = fit.T$estimate
8 mean.T = params.T[1]
9 sd.T = params.T[2] * sqrt(params.T[3] / (params.T[3] - 2))
10 nu.T = params.T[3]
11 x = seq(-0.04, 0.04, by = 0.0001)
12 hist(LogRet, 80, freq = FALSE)
13 lines(x, dstd(x, mean = mean.T, sd = sd.T, nu = nu.T),
14       lwd = 2, lty = 2, col = 'red')
```

Problem 11 Referring to lines by number, describe in detail what the code does. Examine the plot and comment on the goodness of fit.

Problem 12 Is the mean significantly different than 0?

Problem 13 Discuss differences between the histogram and the parametric fit. Do you think that the parametric fit is adequate or should a nonparametric estimate be used instead?

Problem 14 How heavy is the tail of the parametric fit? Does it appear that the fitted t-distribution has a finite kurtosis? How confident are you that the kurtosis is finite?

5.20 Exercises

1. Load the CRSPday data set in the Ecdat package and get the variable names with the commands

```
library(Ecdat)
data(CRSPday)
dimnames(CRSPday)[[2]]
```

Plot the IBM returns with the commands

```
r = CRSPday[, 5]
plot(r)
```

Learn the mode and class of the IBM returns with

```
mode(r)
class(r)
```

You will see that the class of the variable `r` is “`ts`,” which means “time series.” Data of class `ts` are plotted differently than data not of this class. To appreciate this fact, use the following commands to convert the IBM returns to class `numeric` before plotting them:

```
r2 = as.numeric(r)
class(r2)
plot(r2)
```

The variable `r2` contains the same data as the variable `r`, but `r2` has class `numeric`.

Find the covariance matrix, correlation matrix, and means of GE, IBM, and Mobil with the commands

```
cov(CRSPday[, 4:6])
cor(CRSPday[, 4:6])
apply(CRSPday[, 4:6], 2, mean)
```

Use your R output to answer the following questions:

- (a) What is the mean of the Mobil returns?
- (b) What is the variance of the GE returns?
- (c) What is the covariance between the GE and Mobil returns?
- (d) What is the correlation between the GE and Mobil returns?

2. Suppose that Y_1, \dots, Y_n are i.i.d. $N(\mu, \sigma^2)$, where μ is *known*. Show that the MLE of σ^2 is

$$n^{-1} \sum_{i=1}^n (Y_i - \mu)^2.$$

3. Show that $f^*(y|\xi)$ given by Eq. (5.15) integrates to $(\xi + \xi^{-1})/2$.
4. Let X be a random variable with mean μ and standard deviation σ .
- (a) Show that the kurtosis of X is equal to 1 plus the variance of $\{(X - \mu)/\sigma\}^2$.
 - (b) Show that the kurtosis of any random variable is at least 1.
 - (c) Show that a random variable X has a kurtosis equal to 1 if and only if $P(X = a) = P(X = b) = 1/2$ for some $a \neq b$.
5. (a) What is the kurtosis of a normal mixture distribution that is 95% $N(0, 1)$ and 5% $N(0, 10)$?
- (b) Find a formula for the kurtosis of a normal mixture distribution that is $100p\% N(0, 1)$ and $100(1 - p)\% N(0, \sigma^2)$, where p and σ are parameters. Your formula should give the kurtosis as a function of p and σ .
- (c) Show that the kurtosis of the normal mixtures in part (b) can be made arbitrarily large by choosing p and σ appropriately. Find values of p and σ so that the kurtosis is 10,000 or larger.
- (d) Let $M > 0$ be arbitrarily large. Show that for any $p_0 < 1$, no matter how close to 1, there is a $p > p_0$ and a σ , such that the normal mixture with these values of p and σ has a kurtosis at least M . This shows that there is a normal mixture arbitrarily close to a normal distribution but with a kurtosis above any arbitrarily large value of M .
6. Fit the F-S skewed t -distribution to the gas flow data. The data set is in the file `GasFlowData.csv`, which can be found on the book's website.
7. Suppose that X_1, \dots, X_n are i.i.d. $\text{exponential}(\theta)$. Show that the MLE of θ is \bar{X} .
8. For any univariate parameter θ and estimator $\hat{\theta}$, we define the bias to be $\text{Bias}(\hat{\theta}) = E(\hat{\theta}) - \theta$ and the MSE (mean square error) to be $\text{MSE}(\hat{\theta}) = E(\hat{\theta} - \theta)^2$. Show that
- $$\text{MSE}(\hat{\theta}) = \{\text{Bias}(\hat{\theta})\}^2 + \text{Var}(\hat{\theta}).$$
9. Suppose that $X_1, \dots, X_n \stackrel{iid}{\sim} \text{Normal}(\mu, \sigma^2)$, with $0 < \sigma^2 < \infty$, and define $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$. What is $\text{Bias}(\hat{\mu})$? What is $\text{MSE}(\hat{\mu})$? What if the distribution of the X_i is not Normal, but Student's t distribution with the same mean μ and variance σ^2 , and tail index (ν, df) of 5?
10. Assume that you have a sample from a t -distribution and the sample kurtosis is 9. Based on this information alone, what would you use as an estimate of ν , the tail-index parameter?
11. The number of small businesses in a certain region defaulting on loans was observed for each month over a 4-year period. In the R program below,

the variable y is the number of defaults in a month and x is the value for that month of an economic variable thought to affect the default rate. The function `dpois` computes the Poisson density.

```
start =c(1,1)
loglik = function(theta) {-sum(log(dpois(y,
    lambda = exp(theta[1] + theta[2] * x))))}
mle = optim(start, loglik, hessian = TRUE)
invFishInfo = solve(mle$hessian)
options(digits = 4)
mle$par
mle$value
mle$convergence
sqrt(diag(invFishInfo))
```

The output is

```
> mle$par
[1] 1.0773 0.4529
> mle$value
[1] 602.4
> mle$convergence
[1] 0
> sqrt(diag(invFishInfo))
[1] 0.08742 0.03912
```

- (a) Describe the statistical model being used here.
 - (b) What are the parameter estimates?
 - (c) Find 95 % confidence intervals for the parameters in the model. Use a normal approximation.
12. In this problem you will fit a t -distribution by maximum likelihood to the daily log returns for BMW. The data are in the data set `bmw` that is part of the `evir` package. Run the following code:

```
library(evir)
library(fGarch)
data(bmw)
start_bmw = c(mean(bmw), sd(bmw), 4)
loglik_bmw = function(theta)
{
  -sum(dstd(bmw, mean = theta[1], sd = theta[2],
    nu = theta[3], log = TRUE))
}
mle_bmw = optim(start_bmw, loglik_bmw, hessian = TRUE)
CovMLE_bmw = solve(mle_bmw$hessian)
```

Note: The R code defines a function `loglik_bmw` that is minus the log-likelihood. See Chap. 10 of *An Introduction to R* for more information about functions in R. Also, see page 59 of this manual for more about maximum likelihood estimation in R. `optim` minimizes this objective function

and returns the MLE (which is `mle_bmw$par`) and other information, including the Hessian of the objective function evaluated at the MLE (because `hessian=TRUE`—the default is not to return the Hessian).

- (a) What does the function `dstd` do, and what package is it in?
 - (b) What does the function `solve` do?
 - (c) What is the estimate of ν , the degrees-of-freedom parameter?
 - (d) What is the standard error of ν ?
13. In this problem, you will fit a t -distribution to daily log returns of Siemens. You will estimate the degrees-of-freedom parameter graphically and then by maximum likelihood. Run the following code, which produces a 3×2 matrix of probability plots. If you wish, add reference lines as done in Sect. 4.10.1.

```
library(evir)
data(siemens)
n = length(siemens)
par(mfrow = c(3, 2))
qqplot(siemens, qt(((1 : n) - 0.5) / n, 2),
       ylab = "t(2) quantiles",
       xlab = "data quantiles")
qqplot(siemens, qt(((1:n)-.5)/n,3),ylab="t(3) quantiles",
       xlab="data quantiles")
qqplot(siemens,qt(((1:n)-.5)/n,4),ylab="t(4) quantiles",
       xlab="data quantiles")
qqplot(siemens,qt(((1:n)-.5)/n,5),ylab="t(5) quantiles",
       xlab="data quantiles")
qqplot(siemens,qt(((1:n)-.5)/n,8),ylab="t(8) quantiles",
       xlab="data quantiles")
qqplot(siemens,qt(((1:n)-.5)/n,12),ylab="t(12) quantiles",
       xlab="data quantiles")
```

R has excellent graphics capabilities—see Chap. 12 of *An Introduction to R* for more about R graphics and, in particular, pages 67 and 72 for more information about `par` and `mfrow`, respectively.

- (a) Do the returns have lighter or heavier tails than a t -distribution with 2 degrees of freedom?
- (b) Based on the QQ plots, what seems like a reasonable estimate of ν ?
- (c) What is the MLE of ν for the Siemens log returns?

References

- Arellano-Valle, R. B., and Azzalini, A. (2013) The centred parameterization and related quantities of the skew- t distribution. *Journal of Multivariate Analysis*, 113, 73–90.
- Azzalini, A. (2014) *The Skew-Normal and Related Families (Institute of Mathematical Statistics Monographs, Book 3)*, Cambridge University Press.

- Azzalini, A., and Capitanio, A. (2003) Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t distribution. *Journal of the Royal Statistics Society, Series B*, **65**, 367–389.
- Boos, D. D., and Stefanski, L. A. (2013) *Essential Statistical Inference*, Springer.
- Box, G. E. P., and Dox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, **26** 211–246.
- Buch-Larsen, T., Nielsen, J. P., Guillén, M., and Bolance, C. (2005), Kernel density estimation for heavy-tailed distributions using the champernowne transformation. *Statistics*, **39**, 503–518.
- Burnham, K. P. and Anderson, D. R. (2002) *Model Selection and Multimodel Inference*, Springer, New York.
- Casella, G. and Berger, R. L. (2002) *Statistical Inference*, 2nd ed., Duxbury/Thomson Learning, Pacific Grove, CA.
- Efron, B., and Hinkley, D. V. (1978) Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika*, **65**, 457–487.
- Fernandez, C., and Steel, M. F. J. (1998) On Bayesian Modelling of fat tails and skewness, *Journal of the American Statistical Association*, **93**, 359–371.
- Hurvich, C. M., and Tsai, C-L. (1989) Regression and time series model selection in small samples. *Biometrika*, **76**, 297–307.
- Lehmann, E. L. (1999) *Elements of Large-Sample Theory*, Springer-Verlag, New York.
- Ruppert, D., and Wand, M. P. (1992) Correction for kurtosis in density estimation. *Australian Journal of Statistics*, **34**, 19–29.
- Self, S. G., and Liang, K. Y. (1987) Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under non-standard conditions. *Journal of the American Statistical Association*, **82**, 605–610.
- Serfling, R. J. (1980) *Approximation Theorems of Mathematical Statistics*, Wiley, New York.
- van der Vaart, A. W. (1998) *Asymptotic Statistics*, Cambridge University Press, Cambridge.
- Wasserman, L. (2004) *All of Statistics*, Springer, New York.

Resampling

6.1 Introduction

Finding a single set of estimates for the parameters in a statistical model is not enough. An assessment of the uncertainty in these estimates is also needed. Standard errors and confidence intervals are common methods for expressing uncertainty.¹ In the past, it was sometimes difficult, if not impossible, to assess uncertainty, especially for complex models. Fortunately, the speed of modern computers, and the innovations in statistical methodology inspired by this speed, have largely overcome this problem. In this chapter we apply a computer simulation technique called the “bootstrap” or “resampling” to find standard errors and confidence intervals. The bootstrap method is very widely applicable and will be used extensively in the remainder of this book. The bootstrap is one way that modern computing has revolutionized statistics. Markov chain Monte Carlo (MCMC) is another; see Chap. 20.

The term “bootstrap” was coined by Bradley Efron (1979) and comes from the phrase “pulling oneself up by one’s bootstraps.”

When statistics are computed from a randomly chosen sample, then these statistics are random variables. Students often do not appreciate this fact. After all, what could be random about \bar{Y} ? We just averaged the data, so what is random? The point is that the sample is only one of many possible samples. Each possible sample gives a different value of \bar{Y} . Thus, although we only see one value of \bar{Y} , it was selected at random from the many possible values and therefore \bar{Y} is a random variable.

Methods of statistical inference such as confidence intervals and hypothesis tests are predicated on the randomness of statistics. For example, the

¹ See Appendices A.16.2 and A.17 for introductions to standard errors and confidence intervals.

confidence coefficient of a confidence interval tells us the probability, before a random sample is taken, that an interval constructed from the sample will contain the parameter. Therefore, by the law of large numbers, the confidence coefficient is also the long-run frequency of intervals that cover their parameter. Confidence intervals are usually derived using probability theory. Often, however, the necessary probability calculations are intractable, and in such cases we can replace theoretical calculations by Monte Carlo simulation.

But how do we simulate sampling from an *unknown* population? The answer, of course, is that we cannot do this exactly. However, a sample is a good representative of the population, and we can simulate sampling from the population by sampling from the sample, which is called *resampling*.

Each resample has the same sample size n as the original sample. The reason for this is that we are trying to simulate the original sampling, so we want the resampling to be as similar as possible to the original sampling. By *bootstrap approximation*, we mean the approximation of the sampling process by resampling.

There are two basic resampling methods, model-free and model-based, which are also known, respectively, as nonparametric and parametric. In this chapter, we assume that we have an i.i.d. sample from some population. For dependent data, resampling requires different techniques, which will be discussed in Sect. 13.6.

In *model-free resampling*, the resamples are drawn *with replacement* from the original sample. Why with replacement? The reason is that only sampling with replacement gives independent observations, and we want the resamples to be i.i.d. just as the original sample. In fact, if the resamples were drawn without replacement, then every resample would be exactly the same as the original sample, so the resamples would show no random variation. This would not be very satisfactory, of course.

Model-based resampling does not take a sample from the original sample. Instead, one assumes that the original sample was drawn i.i.d. from a density in the parametric family, $\{f(\mathbf{y}|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \Theta\}$, so, for an unknown value of $\boldsymbol{\theta}$, $f(\mathbf{y}|\boldsymbol{\theta})$ is the population density. The resamples are drawn i.i.d. from the density $f(\mathbf{y}|\hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}}$ is some estimate of the parameter vector $\boldsymbol{\theta}$.

The number of resamples taken should, in general, be large. Just how large depends on the context and is discussed more fully later. Sometimes thousands or even tens of thousands of resamples are used. We let B denote the number of resamples.

When reading the following section, keep in mind that with resampling, the original sample plays the role of the population, because the resamples are taken from the original sample. Therefore, estimates from the sample play the role of true population parameters.

6.2 Bootstrap Estimates of Bias, Standard Deviation, and MSE

Let θ be a one-dimensional parameter, let $\hat{\theta}$ be its estimate from the sample, and let $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$ be estimates from B resamples. Also, define $\bar{\hat{\theta}}^*$ to be the mean of $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$. An asterisk indicates a statistic calculated from a resample.

The bias of $\hat{\theta}$ is defined as $\text{BIAS}(\hat{\theta}) = E(\hat{\theta}) - \theta$. Since expectations, which are population averages, are estimated by averaging over resamples, the bootstrap estimate of bias is

$$\text{BIAS}_{\text{boot}}(\hat{\theta}) = \bar{\hat{\theta}}^* - \hat{\theta}. \quad (6.1)$$

Notice that, as discussed in the last paragraph of the previous section, in the bootstrap estimate of bias, the unknown population parameter θ is replaced by the estimate $\hat{\theta}$ from the sample. The bootstrap standard error for $\hat{\theta}$ is the sample standard deviation of $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$, that is,

$$s_{\text{boot}}(\hat{\theta}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^B (\hat{\theta}_b^* - \bar{\hat{\theta}}^*)^2}. \quad (6.2)$$

$s_{\text{boot}}(\hat{\theta})$ estimates the standard deviation of $\hat{\theta}$.

The mean-squared error (MSE) of $\hat{\theta}$ is $E(\hat{\theta} - \theta)^2$ and is estimated by

$$\text{MSE}_{\text{boot}}(\hat{\theta}) = \frac{1}{B} \sum_{b=1}^B (\hat{\theta}_b^* - \hat{\theta})^2.$$

As in the estimation of bias, when estimating MSE, the unknown θ is replaced by $\hat{\theta}$. The MSE reflects both bias and variability and, in fact,

$$\text{MSE}_{\text{boot}}(\hat{\theta}) \approx \text{BIAS}_{\text{boot}}^2(\hat{\theta}) + s_{\text{boot}}^2(\hat{\theta}). \quad (6.3)$$

We would have equality in (6.3), rather than an approximation, if in the denominator of (6.1) we used B rather than $B - 1$. Since B is usually large, the error of the approximation is typically very small.

6.2.1 Bootstrapping the MLE of the t -Distribution

Functions that compute the MLE, such as, `fitdistr()` in R, usually compute standard errors for the MLE along with the estimates themselves. The standard errors are justified theoretically by an “asymptotic” or “large-sample” approximation, called the CLT (central limit theorem) for the maximum likelihood estimator.² This approximation becomes exact only as the sample size

² See Sect. 5.10.

increases to ∞ . Since a sample size is always finite, one cannot be sure of the accuracy of the standard errors. Computing standard errors by the bootstrap can serve as a check on the accuracy of the large-sample approximation, as illustrated in the following example.

Example 6.1. Bootstrapping GE Daily Returns

This example uses the GE daily returns from January 3, 1969, to December 31, 1998, in the data set `CRSPday` in R's `Ecdat` package. The sample size is 2,528 and the number of resamples is $B = 1,000$. The t -distribution was fit using `fitdistr()` in R and the model-free bootstrap was used. The first and third lines in Table 6.1 are the estimates and standard errors returned by `fitdistr()`, which uses observed Fisher information to calculate standard errors. The second and fourth lines have the results from bootstrapping. The differences between "Estimate" and "Bootstrap mean" are the bootstrap estimates of bias. We can see that the biases are small relative to the standard errors in the row labeled "SE." Small, and even negligible, bias is common when the sample size is in the thousands, as in this example.

Table 6.1. Estimates from fitting a t -distribution to the 2,528 GE daily returns. "Estimate" = MLE. "SE" is standard error from observed Fisher information returned by the R function `fitdistr()`. "Bootstrap mean" and "Bootstrap SE" are the sample mean and standard deviation of the maximum likelihood estimates from 1,000 bootstrap samples. ν is the degrees-of-freedom parameter. The model-free bootstrap was used.

	μ	σ	ν
Estimate	0.000879	0.0113	6.34
Bootstrap mean	0.000874	0.0113	6.30
SE	0.000253	0.000264	0.73
Bootstrap SE	0.000252	0.000266	0.82

It is reassuring that "SE" and "Bootstrap SE" agree as closely as they do in Table 6.1. This is an indication that both are reliable estimates of the uncertainty in the parameter estimates. Such close agreement is more likely with samples as large as this one. \square

```

1 library(bootstrap)
2 library(MASS)
3 set.seed("3857")
4 data(CRSPday, package = "Ecdat")
5 ge = CRSPday[,4]
6 nboot = 1000
7 t_mle = function(x){as.vector(fitdistr(x, "t")$estimate)}
8 results = bootstrap(ge, nboot, t_mle)

```

```

9 rowMeans(results$thetastar[ , ])
10 apply(results$thetastar[,], 1, sd)
11 fitdistr(ge, "t")

```

The code above computes the results reported in Table 6.1. The bootstrap was performed at line 8 by the function `bootstrap()` in the `bootstrap` package which is loaded at line 1. The function `bootstrap()` has three arguments, the data, the value of B , and the function that computes the statistic to be bootstrapped; in this example that function is `t_mle()` which is defined at line 7. The function `fitdistr()` is in the package `MASS` that is loaded at line 2. Lines 9, 10, and 11 compute, respectively, the bootstrap mean, the bootstrap SEs, and the MLE and its standard errors.

Example 6.2. Bootstrapping GE daily returns, continued

To illustrate the bootstrap for a smaller sample size, we now use only the first 250 daily GE returns, approximately the first year of data. The number of bootstrap samples is 1,000. The results are in Table 6.2. For μ and s , the results in Tables 6.1 and 6.2 are comparable though the standard errors in Table 6.2 are, of course, larger because of the smaller sample size. For the parameter ν , the results in Table 6.2 are different in two respects from those in Table 6.1. First, the estimate and the bootstrap mean differ by more than 1, a sign that there is some bias. Second, the bootstrap standard deviation is 2.99, considerably larger than the SE, which is only 1.97. This suggests that the SE, which is based on large-sample theory, specifically the CLT for the MLE, is not an accurate measure of uncertainty in the parameter ν , at least not for the smaller sample.

Table 6.2. Estimates from fitting a t -distribution to the first 250 GE daily returns. Notation as in Table 6.1. The nonparametric bootstrap was used.

	μ	σ	ν
Estimate	0.00142	0.01055	5.52
Bootstrap mean	0.00145	0.01064	6.77
SE	0.000764	0.000817	1.98
Bootstrap SE	0.000777	0.000849	2.99

To gain some insight about why the results of ν in these two tables disagree, kernel density estimates of the two bootstrap samples were plotted in Fig. 6.1. We see that with the smaller sample size in panel (a), the density is bimodal and has noticeable right skewness. The density with the full sample is unimodal and has much less skewness.

Tail-weight parameters such as ν are difficult to estimate unless the sample size is in the thousands. With smaller sample sizes, such as 250, there will

not be enough extreme observations to obtain a precise estimate of the tail-weight parameters. This problem has been nicely illustrated by the bootstrap. The number of extreme observations will vary between bootstrap samples. The bootstrap samples with fewer extreme values will have larger estimates of ν , since larger values of ν correspond to thinner tails.

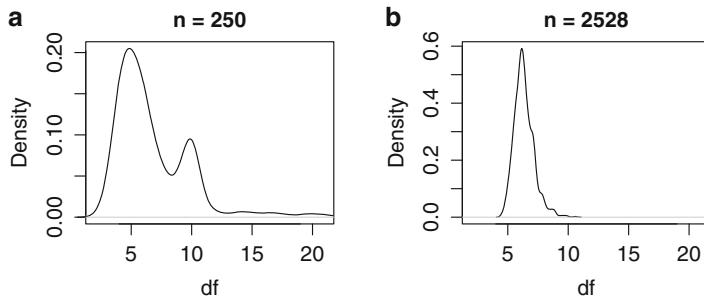


Fig. 6.1. Kernel density estimates of 1,000 bootstrap estimates of df using (a) the first 250 daily GE returns and (b) all 2,528 GE returns. The default bandwidth was used in R's `density` function to create the estimates.

However, even with only 250 observations, ν can be estimated accurately enough to show, for example, that for the GE daily returns ν is very likely less than 13, the 98th percentile of the bootstrap distribution of ν . Therefore, the bootstrap provides strong evidence that the normal model corresponding to $\nu = \infty$ is not as satisfactory as a t -model.

By the CLT for the MLE, we know that the MLE is nearly normally distributed for large enough values of n . But this theorem does not tell us how large is large enough. To answer that question, we can use the bootstrap. We have seen here that $n = 250$ is not large enough for near normality of $\hat{\nu}$, and, though $n = 2,528$ is sufficiently large so that the bootstrap distribution is unimodal, there is still some right skewness when $n = 2,528$. \square

6.3 Bootstrap Confidence Intervals

Besides its use in estimating bias and finding standard errors, the bootstrap is widely used to construct confidence intervals. There are many bootstrap confidence intervals and some are quite sophisticated. We can only describe a few and the reader is pointed to the references in Sect. 6.4 for additional information.

Except in certain simple cases, confidence intervals are based on approximations such as the CLT for the MLE. The bootstrap is based on the approximation of the population's probability distribution using the sample. When a confidence interval uses an approximation, there are two coverage probabilities, the nominal one that is stated and the actual one that is unknown. Only for exact confidence intervals making no use of approximations will the two

probabilities be equal. By the “accuracy” of a confidence interval, we mean the degree of agreement between the nominal and actual coverage probabilities. Even exact confidence intervals such as (A.44) for a normal mean and (A.45) for a normal variance are exact only when the data meet the assumptions exactly, e.g., are exactly normally distributed.

6.3.1 Normal Approximation Interval

Let $\hat{\theta}$ be an estimate of θ and let $s_{\text{boot}}(\hat{\theta})$ be the estimate of standard error given by (6.2). Then the normal theory confidence interval for θ is

$$\hat{\theta} \pm s_{\text{boot}}(\hat{\theta}) z_{\alpha/2}, \quad (6.4)$$

where $z_{\alpha/2}$ is the $\alpha/2$ -upper quantile of the normal distribution. When $\hat{\theta}$ is an MLE, this interval is essentially the same as (5.20) except that bootstrap, rather than the Fisher information, is used to find the standard error.

To avoid confusion, it should be emphasized that the normal approximation does not assume that the population is normally distributed but only that $\hat{\theta}$ is normally distributed by a CLT.

6.3.2 Bootstrap- t Intervals

Often one has available a standard error for $\hat{\theta}$, for example, from Fisher information. In this case, the bootstrap- t method can be used and, compared to normal approximation confidence intervals, offers the possibility of more accurate confidence intervals, that is, with nominal coverage probability closer to the actual coverage probability. We start by showing how the bootstrap- t method is related to the usual t -based confidence interval for a normal population mean, and then discuss the general theory.

Confidence Intervals for a Population Mean

Suppose we wish to construct a confidence interval for the population mean based on a random sample. One starts with the so-called “ t -statistic,”³ which is

$$t = \frac{\mu - \bar{Y}}{s/\sqrt{n}}. \quad (6.5)$$

The denominator of t , s/\sqrt{n} , is just the standard error of the mean, so that the denominator estimates the standard deviation of the numerator.

³ Actually, t is not quite a statistic since it depends on the unknown μ , whereas a statistic, by definition, is something that depends only on the sample, not on unknown parameters. However, the term “ t -statistic” is so widespread that we will use it here.

If we are sampling from a normally distributed population, then the probability distribution of t is known to be the t -distribution with $n - 1$ degrees of freedom. Using the notation of Sect. 5.5.2, we denote by $t_{\alpha/2,n-1}$ the $\alpha/2$ upper t -value, that is, the $\alpha/2$ -upper quantile of this distribution. Thus, t in (6.5) has probability $\alpha/2$ of exceeding $t_{\alpha/2,n-1}$. Because of the symmetry of the t -distribution, the probability is also $\alpha/2$ that t is less than $-t_{\alpha/2,n-1}$.

Therefore, for normally distributed data, the probability is $1 - \alpha$ that

$$-t_{\alpha/2,n-1} \leq t \leq t_{\alpha/2,n-1}. \quad (6.6)$$

Substituting (6.5) into (6.6), after a bit of algebra we find that

$$1 - \alpha = P \left\{ \bar{Y} - t_{\alpha/2,n-1} \frac{s}{\sqrt{n}} \leq \mu \leq \bar{Y} + t_{\alpha/2,n-1} \frac{s}{\sqrt{n}} \right\}, \quad (6.7)$$

which shows that

$$\bar{Y} \pm \frac{s}{\sqrt{n}} t_{\alpha/2,n-1}$$

is a $1 - \alpha$ confidence interval for μ , assuming normally distributed data. This is the confidence interval given by Eq. (A.44). Note that in (6.7) the random variables are \bar{Y} and s , and μ is fixed.

What if we are not sampling from a normal distribution? In that case, the distribution of t defined by (6.5) is *not* the t -distribution, but rather some other distribution that is not known to us. There are two problems. First, we do not know the distribution of the population. Second, even if the population distribution were known, it is a difficult, usually intractable, probability calculation to get the distribution of the t -statistic from the distribution of the population. This calculation has only been done for normal populations. Considering the difficulty of these two problems, can we still get a confidence interval? The answer is “yes, by resampling.”

We start with a large number, say B , of resamples from the original sample. Let $\bar{Y}_{\text{boot},b}$ and $s_{\text{boot},b}$ be the sample mean and standard deviation of the b th resample, $b = 1, \dots, B$, and let \bar{Y} be the mean of the original sample. Define

$$t_{\text{boot},b} = \frac{\bar{Y} - \bar{Y}_{\text{boot},b}}{s_{\text{boot},b}/\sqrt{n}}. \quad (6.8)$$

Notice that $t_{\text{boot},b}$ is defined in the same way as t except for two changes. First, \bar{Y} and s in t are replaced by $\bar{Y}_{\text{boot},b}$ and $s_{\text{boot},b}$ in $t_{\text{boot},b}$. Second, μ in t is replaced by \bar{Y} in $t_{\text{boot},b}$. The last point is a bit subtle, and uses the principle stated at the end of Sect. 6.1—a resample is taken using the original sample as the population. Thus, for the resample, the population mean is \bar{Y} !

Because the resamples are independent of each other, the collection $t_{\text{boot},1}, t_{\text{boot},2}, \dots$ can be treated as a random sample from the distribution of the t -statistic. After B values of $t_{\text{boot},b}$ have been calculated, one from each resample, we find the $\alpha/2$ -lower and -upper quantiles of these $t_{\text{boot},b}$ values. Call these percentiles t_L and t_U .

If the original population is skewed, then there is no reason to suspect that the $\alpha/2$ -lower quantile is minus the $\alpha/2$ -upper quantile as happens for symmetric populations such as the t -distribution. In other words, we do not necessarily expect that $t_L = -t_U$, but this causes us no problem since the bootstrap allows us to estimate t_L and t_U without assuming any relationship between them. Now we replace $-t_{\alpha/2,n-1}$ and $t_{\alpha/2,n-1}$ in the confidence interval (6.7) by t_L and t_U , respectively. Finally, the bootstrap confidence interval for μ is

$$\left(\bar{Y} + t_L \frac{s}{\sqrt{n}}, \bar{Y} + t_U \frac{s}{\sqrt{n}} \right). \quad (6.9)$$

In (6.9), \bar{Y} and s are the mean and standard deviation of the original sample, and only t_L and t_U are calculated from the B bootstrap resamples.

The bootstrap has solved both problems mentioned above. One does not need to know the population distribution since we can estimate it by the sample. A sample isn't a probability distribution. What is being done is creating a probability distribution, called the *empirical distribution*, from the sample by giving each observation in the sample probability $1/n$ where n is the sample size. Moreover, one doesn't need to calculate the distribution of the t -statistic using probability theory. Instead we can simulate from the empirical distribution.

Confidence Interval for a General Parameter

The method of constructing a t -confidence interval for μ can be generalized to other parameters. Let $\hat{\theta}$ and $s(\hat{\theta})$ be the estimate of θ and its standard error calculated from the sample. Let $\hat{\theta}_b^*$ and $s_b(\hat{\theta})$ be the same quantities from the b th bootstrap sample. Then the b th bootstrap t -statistic is

$$t_{\text{boot},b} = \frac{\hat{\theta} - \hat{\theta}_b^*}{s_b(\hat{\theta})}. \quad (6.10)$$

As when estimating a population mean, let t_L and t_U be the $\alpha/2$ -lower and $\alpha/2$ -upper sample quantiles of these t -statistics. Then the confidence interval for θ is

$$(\hat{\theta} + t_L s(\hat{\theta}), \hat{\theta} + t_U s(\hat{\theta}))$$

since

$$1 - \alpha \approx P \left\{ t_l \leq \frac{\hat{\theta} - \hat{\theta}_b^*}{s_b(\hat{\theta})} \leq t_U \right\} \quad (6.11)$$

$$\approx P \left\{ t_l \leq \frac{\theta - \hat{\theta}}{s(\hat{\theta})} \leq t_U \right\} \quad (6.12)$$

$$= P \left\{ \hat{\theta} + t_L s(\hat{\theta}) \leq \theta \leq \hat{\theta} + t_U s(\hat{\theta}) \right\}.$$

The approximation in (6.11) is due to Monte Carlo error and can be made small by choosing B large. The approximation in (6.12) is from the bootstrap approximation of the population's distribution by the empirical distribution. The error of the second approximation is independent of B and becomes small only as the sample size n becomes large. Though one generally has no control over the sample size, fortunately, sample sizes are often large in financial engineering.

6.3.3 Basic Bootstrap Interval

Let q_L and q_U be the $\alpha/2$ -lower and -upper sample quantiles of $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$. The fraction of bootstrap estimates that satisfy

$$q_L \leq \hat{\theta}_b^* \leq q_U \quad (6.13)$$

is $1 - \alpha$. But (6.13) is algebraically equivalent to

$$\hat{\theta} - q_U \leq \hat{\theta} - \hat{\theta}_b^* \leq \hat{\theta} - q_L, \quad (6.14)$$

so that $\hat{\theta} - q_U$ and $\hat{\theta} - q_L$ are lower and upper quantiles for the distribution of $\hat{\theta} - \hat{\theta}_b^*$. The basic bootstrap interval uses them as lower and upper quantiles for the distribution of $\theta - \hat{\theta}$. Using the bootstrap approximation, it is assumed that

$$\hat{\theta} - q_U \leq \theta - \hat{\theta} \leq \hat{\theta} - q_L \quad (6.15)$$

will occur in a fraction $1 - \alpha$ of samples. Adding $\hat{\theta}$ to each term in (6.15) gives $2\hat{\theta} - q_U \leq \theta \leq 2\hat{\theta} - q_L$, so that

$$(2\hat{\theta} - q_U, 2\hat{\theta} - q_L) \quad (6.16)$$

is a confidence interval for θ . Interval (6.16) is sometimes called the *basic bootstrap interval*.

6.3.4 Percentile Confidence Intervals

There are several bootstrap confidence intervals based on the so-called percentile method. Only one, the basic percentile interval, is discussed here in detail.

As in Sect. 6.3.3, let q_L and q_U be the $\alpha/2$ -lower and -upper sample quantiles of $\hat{\theta}_1^*, \dots, \hat{\theta}_B^*$. The basic percentile confidence interval is simply

$$(q_L, q_U). \quad (6.17)$$

By (6.13), the proportion of $\hat{\theta}_b^*$ -values in this interval is $1 - \alpha$. This interval can be justified by assuming that $\hat{\theta}^*$ is distributed symmetrically about $\hat{\theta}$. This assumption implies that for some $C > 0$, $q_L = \hat{\theta} - C$ and $q_U = \hat{\theta} + C$.

Then $2\hat{\theta} - q_U = q_L$ and $2\hat{\theta} - q_L = q_U$, so the basic bootstrap interval (6.16) coincides with the basic percentile interval (6.17).

What if $\hat{\theta}^*$ is not distributed symmetrically about $\hat{\theta}$? Fortunately, not all is lost. As discussed in Sect. 4.6, often random variables can be transformed to have a symmetric distribution. So, now assume only that for some monotonically increasing function g , $g(\hat{\theta}^*)$ is symmetrically distributed about $g(\hat{\theta})$. As we will now see, this weaker assumption is all that is needed to justify the basic percentile interval. Because g is monotonically strictly increasing and quantiles are transformation-respecting,⁴ $g(q_L)$ and $g(q_U)$ are lower- and upper- $\alpha/2$ quantiles of $g(\hat{\theta}_1^*), \dots, g(\hat{\theta}_B^*)$, and the basic percentile confidence interval for $g(\theta)$ is

$$\{g(q_L), g(q_U)\}. \quad (6.18)$$

Now, if (6.18) has coverage probability $(1 - \alpha)$ for $g(\theta)$, then, since g is monotonically increasing, (6.17) has coverage probability $(1 - \alpha)$ for θ . This justifies the percentile interval, at least if one is willing to assume the existence of a transformation to symmetry. Note that it is only assumed that such a g exists, not that it is known. No knowledge of g is necessary, since g is not used to construct the percentile interval.

The basic percentile method is simple, but it is not considered very accurate, except for large sample sizes. There are two problems with the percentile method. The first is an assumption of unbiasedness. The basic percentile interval assumes not only that $g(\hat{\theta}^*)$ is distributed symmetrically, but also that it is symmetric about $g(\hat{\theta})$ rather than $g(\hat{\theta})$ plus some bias. Most estimators satisfy a CLT, e.g., the CLTs for sample quantiles and for the MLE in Sects. 4.3.1 and 5.10, respectively. Therefore, bias becomes negligible in large enough samples, but in practice the sample size might not be sufficiently large and bias can cause the nominal and actual coverage probabilities to differ.

The second problem is that $\hat{\theta}$ may have a nonconstant variance, a problem called heteroskedasticity. If $\hat{\theta}$ is the MLE, then the variance of $\hat{\theta}$ is, at least approximately, the inverse of Fisher information and the Fisher information need not be constant—it often depends on θ .

More sophisticated percentile methods can correct for bias and heteroskedasticity. The BC_a and ABC (approximate bootstrap confidence) percentile intervals are improved percentile intervals in common use. In the name “BC_a,” “BC” means “bias-corrected” and “a” means “accelerated,” which refers to the rate at which the variance changes with θ . The BC_a method automatically estimates both the bias and the rate of change of the variance and then makes suitable adjustments. The theory behind the BC_a and ABC intervals is beyond the scope of this book, but is discussed in references found in Sect. 6.4. Both the BC_a and ABC methods have been implemented in statistical software such as R. In R’s `bootstrap` package, the functions `bcanon()`, `abcpars()`, and `abcnon()` implement the nonparametric BC_a, parametric ABC, and nonparametric ABC intervals, respectively.

⁴ See Appendix A.2.2.

Example 6.3. Confidence interval for a quantile-based tail-weight parameter

It was mentioned in Sect. 5.8 that a quantile-based parameter quantifying tail weight can be defined as the ratio of two scale parameters:

$$\frac{s(p_1, 1 - p_1)}{s(p_2, 1 - p_2)}, \quad (6.19)$$

where

$$s(p_1, p_2) = \frac{F^{-1}(p_2) - F^{-1}(p_1)}{a},$$

a is a positive constant that does not affect the ratio (6.19) and so can be ignored, and $0 < p_1 < p_2 < 1/2$. We will call (6.19) `quKurt`. Finding a confidence interval for `quKurt` can be a daunting task without the bootstrap, but with the bootstrap it is simple. In this example, BC_a confidence intervals will be found for `quKurt`. The parameter is computed from a sample `y` by this R function, which has default values $p_1 = 0.025$ and $p_2 = 0.25$:

```
quKurt = function(y, p1 = 0.025, p2 = 0.25)
{
  Q = quantile(y, c(p1, p2, 1 - p2, 1 - p1))
  (Q[4] - Q[1]) / (Q[3] - Q[2])
}
```

The BC_a intervals are found with the `bcanon()` function in the `bootstrap` package using $B = 5,000$. The seed of the random number generator was fixed so that these results can be reproduced.

```
bmw = read.csv("bmw.csv")
library("bootstrap")
set.seed("5640")
bca_kurt = bcanon(bmwRet[, 2], 5000, quKurt)
bca_kurt$confpoints
```

By default, the output gives four pairs of confidence limits.

```
> bca_kurt$confpoints
   alpha bca point
 [1,] 0.025    4.07
 [2,] 0.050    4.10
 [3,] 0.100    4.14
 [4,] 0.160    4.18
 [5,] 0.840    4.41
 [6,] 0.900    4.45
 [7,] 0.950    4.50
 [8,] 0.975    4.54
```

The results above show, for example, that the 90 % BC_a confidence interval is (4.10, 4.50). For reference, any normal distribution has `quKurt` equal 2.91, so these data have heavier than Gaussian tails, at least as measured by `quKurt`.

□

Example 6.4. Confidence interval for the ratio of two quantile-based tail-weight parameters

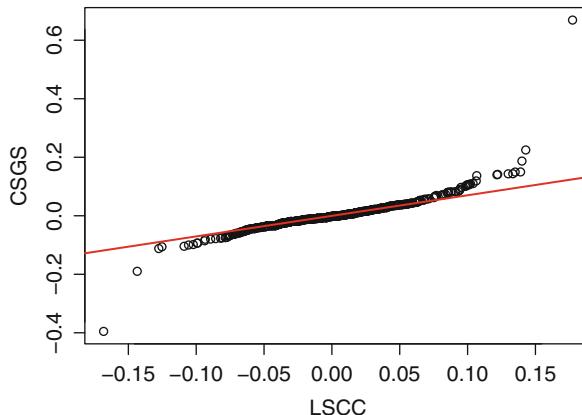


Fig. 6.2. QQ plot of returns on two stocks in the `midcapD.ts` data set. The reference line goes through the first and third quartiles.

This example uses the data set `midcapD.ts.csv` of returns on midcap stocks. Two of the stocks in this data set are LSCC and CSGS. From Fig. 6.2, which is a QQ plot comparing the returns from these two companies, it appears that LSCC returns have lighter tails than CSGS returns. The values of `quKurt` are 2.91 and 4.13 for LSCC and GSGS, respectively, and the ratio of the two values is 0.704. This is further evidence that LSCC returns have the lesser tail weight. A BC_a confidence interval for the ratio of `quKurt` for LSCC and CSGS is found with the following R program.

```

1 midcapD.ts = read.csv("midcapD.ts.csv")
2 attach(midcapD.ts)
3 quKurt = function(y, p1 = 0.025, p2 = 0.25)
4 {
5   Q = quantile(y, c(p1, p2, 1 - p2, 1 - p1))
6   as.numeric((Q[4] - Q[1]) / (Q[3] - Q[2]))
7 }
8 compareQuKurt = function(x, p1 = 0.025, p2 = 0.25, xdata)
9 {
10   quKurt(xdata[x,1], p1, p2) / quKurt(xdata[x,2], p1, p2)
11 }
12 quKurt(LSCC)
13 quKurt(CSGS)
14 xdata = cbind(LSCC, CSGS)
15 compareQuKurt(1:n, xdata = xdata)
16 library("bootstrap")

```

```

17 set.seed("5640")
18 bca_kurt = bcanon((1:n), 5000, compareQuKurt, xdata = xdata)
19 bca_kurt$confpoints

```

The function `compareQuKurt()` (lines 8–11) computes a `quKurt` ratio. The function `bcanon()` is designed to bootstrap a vector, but this example has bivariate data in a matrix with two columns. To bootstrap multivariate data, there is a trick given in R’s help for `bcanon()`—bootstrap the integers 1 to n where n is the sample size. This is done at line 18. The resamples of $1, \dots, n$ allow one to resample the rows of the data vector. Thus, in this example `bcanon()` draws a random sample with replacement from $1, \dots, n$ and selects the rows of `xdata` corresponding to these indices to create a resample.

The 95 % confidence interval for the `quKurt` ratio is 0.587 to 0.897, so with 95 % confidence it can be concluded that LSCC has a smaller value of `quKurt`.

```

> bca_kurt$confpoints
  alpha bca point
[1,] 0.025    0.587
[2,] 0.050    0.607
[3,] 0.100    0.634
[4,] 0.160    0.653
[5,] 0.840    0.811
[6,] 0.900    0.833
[7,] 0.950    0.864
[8,] 0.975    0.897

```

□

6.4 Bibliographic Notes

Efron (1979) introduced the name “bootstrap” and did much to popularize resampling methods. Efron and Tibshirani (1993), Davison and Hinkley (1997), Good (2005), and Chernick (2007) are introductions to the bootstrap that discuss many topics not treated here, including the theory behind the BC_a and ABC methods for confidence intervals. The R package `bootstrap` is described by its authors as “functions for Efron and Tibshirani (1993)” and the package contains the data sets used in that book. The R package `boot` is a more recent set of resampling functions and data sets to accompany Davison and Hinkley (1997).

6.5 R Lab

6.5.1 BMW Returns

This lab uses a data set containing 6146 daily returns on BMW stock from January 3, 1973 to July 23, 1996. Run the following code to fit a skewed t -distribution to the returns and check the fit with a QQ plot.

```

1 library("fGarch")
2 bmwRet = read.csv("bmwRet.csv")
3 n = dim(bmwRet)[1]
4
5 kurt = kurtosis(bmwRet[,2], method = "moment")
6 skew = skewness(bmwRet[,2], method = "moment")
7 fit_skewt = sstdFit(bmwRet[,2])
8
9 q.grid = (1:n) / (n+1)
10 qqplot(bmwRet[,2], qsstd(q.grid, fit_skewt$estimate[1],
11   fit_skewt$estimate[2],
12   fit_skewt$estimate[3], fit_skewt$estimate[4]),
13   ylab = "skewed-t quantiles" )

```

The function `qsstd()` is in the `fGarch` package loaded at line 1. The required package `timeDate` is also loaded and the function `kurtosis()` is in `timeDate`.

Problem 1 *What is the MLE of ν ? Does the t-distribution with this value of ν have a finite skewness and kurtosis?*

Since the kurtosis coefficient based on the fourth central moment is infinite for some distributions, as in Sect. 6.4 we will define a quantile-based kurtosis:

$$\text{quKurt}(F) = \frac{F^{-1}(1 - p_1) - F^{-1}(p_1)}{F^{-1}(1 - p_2) - F^{-1}(p_2)},$$

where F is a CDF and $0 < p_1 < p_2 < 1/2$. Typically, p_1 is close to zero so that the numerator is sensitive to tail weight and p_2 is much larger and measures dispersion in the center of the distribution. Because the numerator and denominator of `quKurt` are each the difference between two quantiles, they are location-free and therefore scale parameters. Moreover, because `quKurt` is a ratio of two scale parameters, it is scale-free and therefore a shape parameter. A typical example would be $p_1 = 0.025$ and $p_2 = 0.25$. `quKurt` is estimated by replacing the population quantiles by sample quantiles.

Problem 2 *Write an R program to plot `quKurt` for the t-distribution as a function of ν . Use $p_1 = 0.025$ and $p_2 = 0.25$. Let ν take values from 1 to 10, incremented by 0.25. If you want to get fancy while labeling the axes, `xlab=expression(nu)` in the call to plot will put a “ ν ” on the x-axis.*

Run the following code, which defines a function to compute `quKurt` and bootstraps this function on the BMW returns. Note that p_1 and p_2 are given default values that are used in the bootstrap and that both model-free and model-based bootstrap samples are taken.

```

quKurt = function(y, p1 = 0.025, p2 = 0.25)
{
  Q = quantile(y, c(p1, p2, 1 - p2, 1 - p1))
  k = (Q[4] - Q[1]) / (Q[3] - Q[2])
  k
}
nboot = 5000
ModelFree_kurt = rep(0, nboot)
ModelBased_kurt = rep(0, nboot)
set.seed("5640")
for (i in 1:nboot)
{
  samp_ModelFree = sample(bmwRet[,2], n, replace = TRUE)
  samp_ModelBased = rsstd(n, fit_skewt$estimate[1],
    fit_skewt$estimate[2],
    fit_skewt$estimate[3], fit_skewt$estimate[4])
  ModelFree_kurt[i] = quKurt(samp_ModelFree)
  ModelBased_kurt[i] = quKurt(samp_ModelBased)
}

```

Problem 3 Plot KDEs of `ModelFree_kurt` and `ModelBased_kurt`. Also, plot side-by-side boxplots of the two samples. Describe any major differences between the model-based and model-free results. Include the plots with your work.

Problem 4 Find 90 % percentile method bootstrap confidence intervals for `quKurt` using the model-based and model-free bootstraps.

Problem 5 BC_a confidence intervals can be constructed using the function `bcanon()` in R's `bootstrap` package. Find a 90 % BC_a confidence interval for `quKurt`. Use 5,000 resamples. Compare the BC_a interval to the model-free percentile interval from Problem 4.

6.5.2 Simulation Study: Bootstrapping the Kurtosis

The sample kurtosis is highly variable because it is based on the 4th moment. As a result, it is challenging to construct an accurate confidence interval for the kurtosis. In this section, five bootstrap confidence intervals for the kurtosis will be compared. The comparisons will be on widths of the intervals, where smaller is better, and actual coverage probabilities, where closer to nominal is better.

Run the following code. Warning: this simulation experiment takes a while to run, e.g., 5 to 10 minutes, and will have only moderate accuracy. To increase the accuracy, you might wish to increase `niter` and `nboot` and run the experiment over a longer period, even overnight.

```

library(bootstrap)
Kurtosis = function(x) mean(((x - mean(x)) / sd(x))^4)
set.seed(3751)
niter = 500
nboot = 400
n = 50
nu = 10
trueKurtosis = 3 + 6 / (nu - 4)
correct = matrix(nrow = niter, ncol = 5)
width = matrix(nrow = niter, ncol = 5)
error = matrix(nrow = niter, ncol = 1)
t1 = proc.time()
for (i in 1:niter){
  y = rt(n,nu)
  int1 = boott(y, Kurtosis, nboot,
    nbootsd = 50)$confpoints[c(3, 9)]
  width[i,1] = int1[2] - int1[1]
  correct[i,1] = as.numeric((int1[1] < trueKurtosis) &
    (trueKurtosis < int1[2]))
  int2 = bcanon(y, nboot, Kurtosis)$confpoints[c(1, 8), 2]
  width[i,2] = int2[2] - int2[1]
  correct[i,2] = as.numeric((int2[1] < trueKurtosis) &
    (trueKurtosis < int2[2]))
  boot = bootstrap(y, nboot, Kurtosis)$thetastar
  int3 = Kurtosis(y) + 1.96 * c(-1, 1) * sd(boot)
  width[i,3] = int3[2] - int3[1]
  correct[i,3] = as.numeric((int3[1] < trueKurtosis) &
    (trueKurtosis < int3[2]))
  int4 = quantile(boot, c(0.025, 0.975))
  width[i,4] = int4[2] - int4[1]
  correct[i,4] = as.numeric((int4[1] < trueKurtosis) &
    (trueKurtosis < int4[2]))
  int5 = 2*Kurtosis(y) - quantile(boot, c(0.975, 0.025))
  width[i,5] = int5[2] - int5[1]
  correct[i,5] = as.numeric((int5[1] < trueKurtosis) &
    (trueKurtosis < int5[2]))
  error[i] = mean(boot) - Kurtosis(y)
}
t2 = proc.time()
(t2 - t1)/60
colMeans(width)
colMeans(correct)
options(digits = 3)
mean(error)
mean(error^2)

```

Problem 6 Which five bootstrap intervals are being used here?

Problem 7 What is the value of B here?

Problem 8 How many simulations are used?

Problem 9 What are the estimates of bias?

Problem 10 What is the estimated MSE?

Problem 11 Estimate the actual coverage probability of the BC_a and bootstrap- t intervals. (Because this is a simulation experiment, it is subject to Monte Carlo errors, so the coverage probability is only estimated.)

Problem 12 Find a 95 % confidence interval for the actual coverage probability of the BC_a interval?

Problem 13 Which interval is most accurate? Would you consider any of the intervals as highly accurate?

Problem 14 How much clock time did the entire simulation take?

As mentioned, kurtosis is difficult to estimate because it is based on the 4th moment and a quantile-based measure of tailweight might be a better alternative. The next problem investigates this conjecture.

Problem 15 Repeat the simulation experiment with kurtosis replaced by `quKurt()` defined in Sect. 6.5.1 Which interval is most accurate now? Would you consider any of the intervals as highly accurate?

6.6 Exercises

1. To estimate the risk of a stock, a sample of 50 log returns was taken and s was 0.31. To get a confidence interval for σ , 10,000 resamples were taken. Let $s_{b,\text{boot}}$ be the sample standard deviation of the b th resample. The 10,000 values of $s_{b,\text{boot}}/s$ were sorted and the table below contains selected values of $s_{b,\text{boot}}/s$ ranked from smallest to largest (so rank 1 is the smallest and so forth).

Rank	Value of $s_{b,\text{boot}}/s$
250	0.52
500	0.71
1,000	0.85
9,000	1.34
9,500	1.67
9,750	2.19

Find a 90% confidence interval for σ .

2. In the following R program, resampling was used to estimate the bias and variance of the sample correlation between the variables in the vectors x and y .

```
samplecor = cor(x, y)
n = length(x)
nboot = 5000
resamplecor = rep(0, nboot)
for (b in (1:nboot))
{
  ind = sample(1:n, replace = TRUE)
  resamplecor[b] = cor(x[ind], y[ind])
}
samplecor
mean(resamplecor)
sd(resamplecor)
```

The output is

```
> n
[1] 20
> samplecor
[1] 0.69119
> mean(resamplecor)
[1] 0.68431
> sd(resamplecor)
[1] 0.11293
```

- (a) Estimate the bias of the sample correlation coefficient.
 (b) Estimate the standard deviation of the sample correlation coefficient.
 (c) Estimate the MSE of the sample correlation coefficient.
 (d) What fraction of the MSE is due to bias? How serious is the bias?
 Should something be done to reduce the bias? Explain your answer.
3. The following R code was used to bootstrap the sample standard deviation.

```
( code to read the variable x )
sampleSD = sd(x)
n = length(x)
nboot = 15000
resampleSD = rep(0, nboot)
```

```

for (b in (1:nboot))
{
  resampleSD[b] = sd(sample(x, replace = TRUE))
}
options(digits = 4)
sampleSD
mean(resampleSD)
sd(resampleSD)

```

The output is

```

> sampleSD
[1] 1.323
> mean(resampleSD)
[1] 1.283
> sd(resampleSD)
[1] 0.2386

```

- (a) Estimate the bias of the sample standard deviation of x .
- (b) Estimate the mean squared error of the sample standard deviation of x .

References

- Chernick, M. R. (2007) *Bootstrap Methods: A Guide for Practitioners and Researchers*, 2nd ed., Wiley-Interscience, Hoboken, NJ.
- Davison, A. C., and Hinkley, D. V. (1997) *Bootstrap Methods and Their Applications*, Cambridge University Press, Cambridge.
- Efron, B. (1979) Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7, 1–26.
- Efron, B., and Tibshirani, R. (1993) *An Introduction to the Bootstrap*, Chapman & Hall, New York.
- Good, P. I. (2005) *Resampling Methods: A Practical Guide to Data Analysis*, 3rd ed., Birkhauser, Boston.

Multivariate Statistical Models

7.1 Introduction

Often we are not interested merely in a single random variable but rather in the joint behavior of several random variables, for example, returns on several assets and a market index. Multivariate distributions describe such joint behavior. This chapter is an introduction to the use of multivariate distributions for modeling financial markets data. Readers with little prior knowledge of multivariate distributions may benefit from reviewing Appendices A.12–A.14 before reading this chapter.

7.2 Covariance and Correlation Matrices

Let $\mathbf{Y} = (Y_1, \dots, Y_d)^\top$ be a random vector. We define the expectation vector of \mathbf{Y} to be

$$E(\mathbf{Y}) = \begin{pmatrix} E(Y_1) \\ \vdots \\ E(Y_d) \end{pmatrix}.$$

The *covariance matrix* of \mathbf{Y} is the matrix whose (i, j) th entry is $\text{Cov}(Y_i, Y_j)$ for $i, j = 1, \dots, N$. Since $\text{Cov}(Y_i, Y_i) = \text{Var}(Y_i)$, the covariance matrix is

$$\text{COV}(\mathbf{Y}) = \begin{pmatrix} \text{Var}(Y_1) & \text{Cov}(Y_1, Y_2) & \cdots & \text{Cov}(Y_1, Y_d) \\ \text{Cov}(Y_2, Y_1) & \text{Var}(Y_2) & \cdots & \text{Cov}(Y_2, Y_d) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(Y_d, Y_1) & \text{Cov}(Y_d, Y_2) & \cdots & \text{Var}(Y_d) \end{pmatrix}.$$

Similarly, the *correlation matrix* of \mathbf{Y} , denoted $\text{CORR}(\mathbf{Y})$, has i, j th element $\rho_{Y_i Y_j}$. Because $\text{Corr}(Y_i, Y_i) = 1$ for all i , the diagonal elements of a correlation

matrix are all equal to 1. Note the use of “COV” and “CORR” to denote matrices and “Cov” and “Corr” to denote scalars.

The covariance matrix can be written as

$$\text{COV}(\mathbf{Y}) = E \left[\{\mathbf{Y} - E(\mathbf{Y})\} \{\mathbf{Y} - E(\mathbf{Y})\}^T \right]. \quad (7.1)$$

There are simple relationships between the covariance and correlation matrices. Let $\mathbf{S} = \text{diag}(\sigma_{Y_1}, \dots, \sigma_{Y_d})$, where σ_{Y_i} is the standard deviation of Y_i . Then

$$\text{CORR}(\mathbf{Y}) = \mathbf{S}^{-1} \text{COV}(\mathbf{Y}) \mathbf{S}^{-1} \quad (7.2)$$

and, equivalently,

$$\text{COV}(\mathbf{Y}) = \mathbf{S} \text{CORR}(\mathbf{Y}) \mathbf{S}. \quad (7.3)$$

The *sample covariance* and *correlation matrices* replace $\text{Cov}(Y_i, Y_j)$ and $\rho_{Y_i Y_j}$ by their estimates given by (A.29) and (A.30).

A *standardized* variable is obtained by subtracting the variable’s mean and dividing the difference by the variable’s standard deviation. After standardization, a variable has a mean equal to 0 and a standard deviation equal to 1. The covariance matrix of standardized variables equals the correlation matrix of original variables, which is also the correlation matrix of the standardized variables.

Example 7.1. CRSPday covariances and correlations

This example uses the CRSPday data set in R’s `Ecdat` package. There are four variables, daily returns from January 3, 1969, to December 31, 1998, on three stocks, GE, IBM, and Mobil, and on the CRSP value-weighted index, including dividends. CRSP is the Center for Research in Security Prices at the University of Chicago. The sample covariance matrix for these four series is

	ge	ibm	mobil	crsp
ge	1.88e-04	8.01e-05	5.27e-05	7.61e-05
ibm	8.01e-05	3.06e-04	3.59e-05	6.60e-05
mobil	5.27e-05	3.59e-05	1.67e-04	4.31e-05
crsp	7.61e-05	6.60e-05	4.31e-05	6.02e-05

It is difficult to get much information just by inspecting the covariance matrix. The covariance between two random variables depends on their variances as well as the strength of the linear relationship between them. Covariance matrices are extremely important as input to, for example, a portfolio analysis, but to understand the relationship between variables, it is much better to examine their sample correlation matrix. The sample correlation matrix in this example is

```

      ge   ibm mobil crsp
ge    1.000 0.334 0.297 0.715
ibm   0.334 1.000 0.159 0.486
mobil 0.297 0.159 1.000 0.429
crsp  0.715 0.486 0.429 1.000

```

We can see that all sample correlations are positive and the largest correlations are between `crsp` and the individual stocks. GE is the stock most highly correlated with `crsp`. The correlations between individual stocks and a market index such as `crsp` are a key component of finance theory, especially the Capital Asset Pricing Model (CAPM) introduced in Chap. 17. \square

7.3 Linear Functions of Random Variables

Often we are interested in finding the expectation and variance of a linear combination (weighted average) of random variables. For example, consider returns on a set of assets. A *portfolio* is simply a weighted average of the assets with weights that sum to one. The weights specify what fractions of the total investment are allocated to the assets. For example, if a portfolio consists of 200 shares of Stock 1 selling at \$88/share and 150 shares of Stock 2 selling at \$67/share, then the weights are

$$w_1 = \frac{(200)(88)}{(200)(88) + (150)(67)} = 0.637 \quad \text{and} \quad w_2 = 1 - w_1 = 0.363. \quad (7.4)$$

Because the return on a portfolio is a linear combination of the returns on the individual assets in the portfolio, the material in this section is used extensively in the portfolio theory of Chaps. 16 and 17.

First, we look at a linear function of a single random variable. If Y is a random variable and a and b are constants, then

$$E(aY + b) = aE(Y) + b.$$

Also,

$$\text{Var}(aY + b) = a^2\text{Var}(Y) \quad \text{and} \quad \sigma_{aY+b} = |a|\sigma_Y.$$

Next, we consider linear combinations of two random variables. If X and Y are random variables and w_1 and w_2 are constants, then

$$E(w_1X + w_2Y) = w_1E(X) + w_2E(Y),$$

and

$$\text{Var}(w_1X + w_2Y) = w_1^2\text{Var}(X) + 2w_1w_2\text{Cov}(X, Y) + w_2^2\text{Var}(Y). \quad (7.5)$$

Check that (7.5) can be reexpressed as

$$\text{Var}(w_1X + w_2Y) = (w_1 \ w_2) \begin{pmatrix} \text{Var}(X) & \text{Cov}(X, Y) \\ \text{Cov}(X, Y) & \text{Var}(Y) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}. \quad (7.6)$$

Although formula (7.6) may seem unnecessarily complicated, we will show that this equation generalizes in an elegant way to more than two random variables; see (7.7) below. Notice that the matrix in (7.6) is the covariance matrix of the random vector $(X \ Y)^T$.

Let $\mathbf{w} = (w_1, \dots, w_d)^T$ be a vector of weights and let $\mathbf{Y} = (Y_1, \dots, Y_d)$ be a random vector. Then

$$\mathbf{w}^T \mathbf{Y} = \sum_{i=1}^N w_i Y_i$$

is a weighted average of the components of \mathbf{Y} . One can easily show that

$$E(\mathbf{w}^T \mathbf{Y}) = \mathbf{w}^T \{E(\mathbf{Y})\}$$

and

$$\text{Var}(\mathbf{w}^T \mathbf{Y}) = \sum_{i=1}^N \sum_{j=1}^N w_i w_j \text{Cov}(Y_i, Y_j).$$

This last result can be expressed more succinctly using vector/matrix notation:

$$\text{Var}(\mathbf{w}^T \mathbf{Y}) = \mathbf{w}^T \text{COV}(\mathbf{Y}) \mathbf{w}. \quad (7.7)$$

Example 7.2. The variance of a linear combination of correlated random variables

Suppose that $\mathbf{Y} = (Y_1 \ Y_2 \ Y_3)^T$, $\text{Var}(Y_1) = 2$, $\text{Var}(Y_2) = 3$, $\text{Var}(Y_3) = 5$, $\rho_{Y_1, Y_2} = 0.6$, and that Y_1 and Y_2 are independent of Y_3 . Find $\text{Var}(Y_1 + Y_2 + 1/2 Y_3)$.

Answer: The covariance between Y_1 and Y_3 is 0 by independence, and the same is true of Y_2 and Y_3 . The covariance between Y_1 and Y_2 is $(0.6)\sqrt{(2)(3)} = 1.47$. Therefore,

$$\text{COV}(\mathbf{Y}) = \begin{pmatrix} 2 & 1.47 & 0 \\ 1.47 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix},$$

and by (7.7),

$$\begin{aligned} \text{Var}(Y_1 + Y_2 + Y_3/2) &= (1 \ 1 \ \frac{1}{2}) \begin{pmatrix} 2 & 1.47 & 0 \\ 1.47 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \frac{1}{2} \end{pmatrix} \\ &= (1 \ 1 \ \frac{1}{2}) \begin{pmatrix} 3.47 \\ 4.47 \\ 2.5 \end{pmatrix} \\ &= 9.19. \end{aligned}$$

□

An important property of covariance and correlation matrices is that they are symmetric and positive semidefinite. A matrix \mathbf{A} is said to be positive semidefinite (definite) if $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ (> 0) for all vectors $\mathbf{x} \neq 0$. By (7.7), any covariance matrix must be positive semidefinite, because otherwise there would exist a random variable with a negative variance, a contradiction. A nonsingular covariance matrix is positive definite. A covariance matrix must be symmetric because $\text{Cov}(Y_i, Y_j) = \text{Cov}(Y_j, Y_i)$ for every i and j . Since a correlation matrix is the covariance matrix of standardized variables, it is also symmetric and positive semidefinite.

7.3.1 Two or More Linear Combinations of Random Variables

More generally, suppose that $\mathbf{w}_1^\top \mathbf{Y}$ and $\mathbf{w}_2^\top \mathbf{Y}$ are two weighted averages of the components of \mathbf{Y} , e.g., returns on two different portfolios. Then

$$\text{Cov}(\mathbf{w}_1^\top \mathbf{Y}, \mathbf{w}_2^\top \mathbf{Y}) = \mathbf{w}_1^\top \text{COV}(\mathbf{Y}) \mathbf{w}_2 = \mathbf{w}_2^\top \text{COV}(\mathbf{Y}) \mathbf{w}_1. \quad (7.8)$$

Example 7.3. (Example 7.2 continued)

Suppose that the random vector $\mathbf{Y} = (Y_1, Y_2, Y_3)^\top$ has the mean vector and covariance matrix used in the previous example and contains the returns on three assets. Find the covariance between a portfolio that allocates 1/3 to each of the three assets and a second portfolio that allocates 1/2 to each of the first two assets. That is, find the covariance between $(Y_1 + Y_2 + Y_3)/3$ and $(Y_1 + Y_2)/2$.

Answer: Let

$$\mathbf{w}_1 = \left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3} \right)^\top$$

and

$$\mathbf{w}_2 = \left(\frac{1}{2} \quad \frac{1}{2} \quad 0 \right)^\top.$$

Then

$$\begin{aligned} \text{Cov} \left\{ \frac{Y_1 + Y_2}{2}, \frac{Y_1 + Y_2 + Y_3}{3} \right\} &= \mathbf{w}_1^\top \text{COV}(\mathbf{Y}) \mathbf{w}_2 \\ &= (1/3 \quad 1/3 \quad 1/3) \begin{pmatrix} 2 & 1.47 & 0 \\ 1.47 & 3 & 0 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix} \\ &= (1.157 \quad 1.490 \quad 1.667) \begin{pmatrix} 1/2 \\ 1/2 \\ 0 \end{pmatrix} \\ &= 1.323. \end{aligned}$$

□

Let \mathbf{W} be a nonrandom $d \times q$ matrix so that $\mathbf{W}^\top \mathbf{Y}$ is a random vector of q linear combinations of \mathbf{Y} . Then (7.7) can be generalized to

$$\text{COV}(\mathbf{W}^\top \mathbf{Y}) = \mathbf{W}^\top \text{COV}(\mathbf{Y}) \mathbf{W}. \quad (7.9)$$

Let \mathbf{Y}_1 and \mathbf{Y}_2 be two random vectors of dimensions n_1 and n_2 , respectively. Then $\boldsymbol{\Sigma}_{\mathbf{Y}_1, \mathbf{Y}_2} = \text{COV}(\mathbf{Y}_1, \mathbf{Y}_2)$ is defined as the $n_1 \times n_2$ matrix whose i, j th element is the covariance between the i th component of \mathbf{Y}_1 and the j th component of \mathbf{Y}_2 , that is, $\boldsymbol{\Sigma}_{\mathbf{Y}_1, \mathbf{Y}_2}$ is the matrix of covariances between the random vectors \mathbf{Y}_1 and \mathbf{Y}_2 .

It is not difficult to show that

$$\text{Cov}(\mathbf{w}_1^\top \mathbf{Y}_1, \mathbf{w}_2^\top \mathbf{Y}_2) = \mathbf{w}_1^\top \text{COV}(\mathbf{Y}_1, \mathbf{Y}_2) \mathbf{w}_2, \quad (7.10)$$

for constant vectors \mathbf{w}_1 and \mathbf{w}_2 of lengths n_1 and n_2 .

7.3.2 Independence and Variances of Sums

If Y_1, \dots, Y_d are independent, or at least uncorrelated, then

$$\text{Var}(\mathbf{w}^\top \mathbf{Y}) = \text{Var}\left(\sum_{i=1}^n w_i Y_i\right) = \sum_{i=1}^n w_i^2 \text{Var}(Y_i). \quad (7.11)$$

When $\mathbf{w}^\top = (1/n, \dots, 1/n)$ so that $\mathbf{w}^\top \mathbf{Y} = \bar{Y}$, then we obtain that

$$\text{Var}(\bar{Y}) = \frac{1}{n^2} \sum_{i=1}^n \text{Var}(Y_i). \quad (7.12)$$

In particular, if $\text{Var}(Y_i) = \sigma^2$ for all i , then we obtain the well-known result that if Y_1, \dots, Y_d are uncorrelated and have a constant variance σ^2 , then

$$\text{Var}(\bar{Y}) = \frac{\sigma^2}{n}. \quad (7.13)$$

Another useful fact that follows from (7.11) is that if Y_1 and Y_2 are uncorrelated, then

$$\text{Var}(Y_1 - Y_2) = \text{Var}(Y_1) + \text{Var}(Y_2). \quad (7.14)$$

7.4 Scatterplot Matrices

A correlation coefficient is only a summary of the linear relationship between variables. Interesting features, such as nonlinearity or the joint behavior of extreme values, remain hidden when only correlations are examined. A solution to this problem is the so-called scatterplot matrix, which is a matrix

of scatterplots, one for each pair of variables. A scatterplot matrix can be created easily with modern statistical software such as R. Figure 7.1 shows a scatterplot matrix for the CRSPday data set.

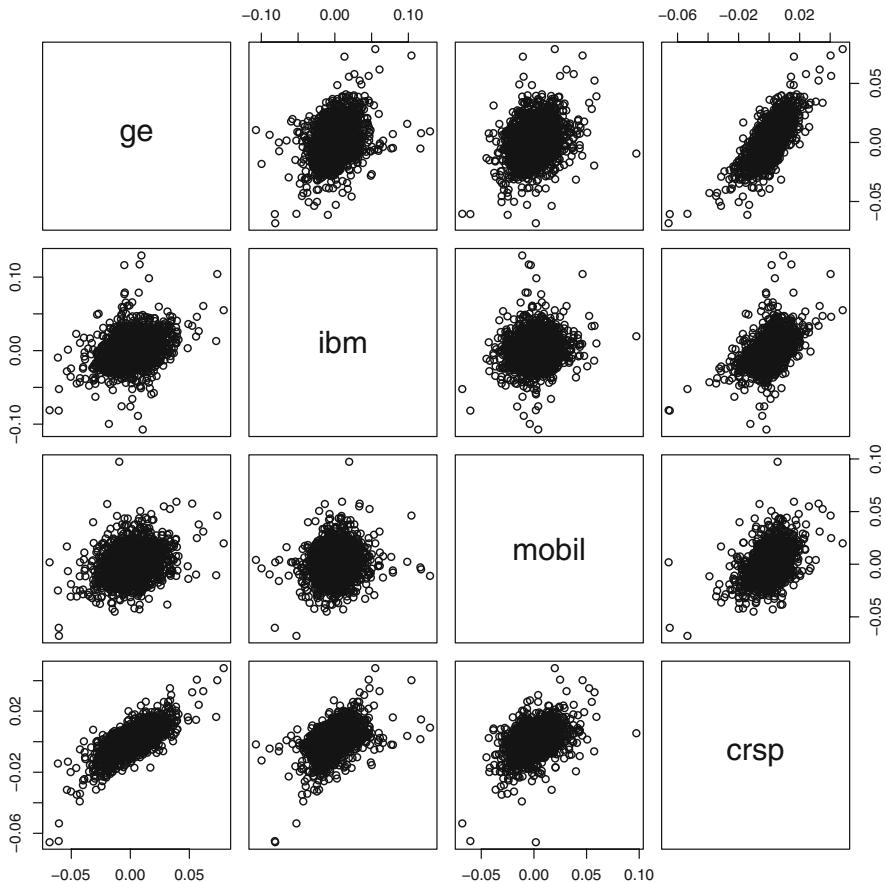


Fig. 7.1. Scatterplot matrix for the CRSPday data set.

One sees little evidence of nonlinear relationships in Fig. 7.1. This lack of nonlinearities is typical of returns on equities, but it should not be taken for granted—instead, one should always look at the scatterplot matrix. The strong linear association between GE and `crsp`, which was suggested before by their high correlation coefficient, can be seen also in their scatterplot.

A portfolio is riskier if large negative returns on its assets tend to occur together on the same days. To investigate whether extreme values tend to cluster in this way, one should look at the scatterplots. In the scatterplot for IBM and Mobil, extreme returns for one stock do not tend to occur on the same days as extreme returns on the other stock; this can be seen by noticing that the outliers tend to fall along the x - and y -axes. The extreme-value behavior

is different with GE and `crsp`, where extreme values are more likely to occur together; note that the outliers have a tendency to occur together, that is, in the upper-right and lower-left corners, rather than being concentrated along the axes. The IBM and Mobil scatterplot is said to show *tail independence*. In contrast, the GE and `crsp` scatterplot is said to show *tail dependence*. Tail dependence is explored further in Chap. 8.

7.5 The Multivariate Normal Distribution

In Chap. 5 we saw the importance of having parametric families of univariate distributions as statistical models. Parametric families of multivariate distributions are equally useful, and the multivariate normal family is the best known of them.

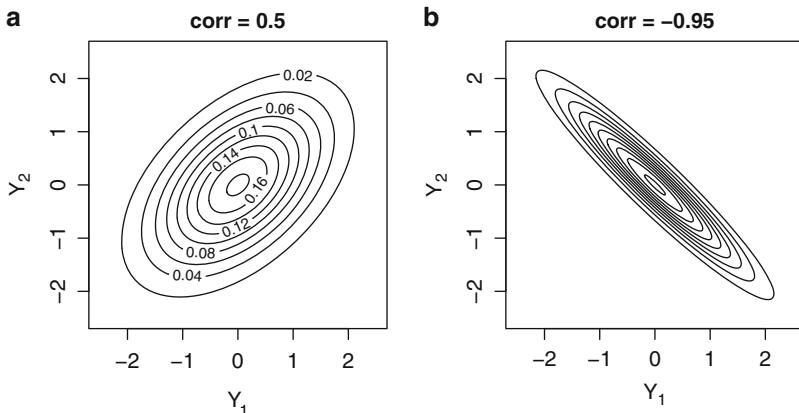


Fig. 7.2. Contour plots of a bivariate normal densities with $N(0, 1)$ marginal distributions and correlations of 0.5 or -0.95.

The random vector $\mathbf{Y} = (Y_1, \dots, Y_d)^\top$ has a *d*-dimensional *multivariate normal distribution* with mean vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d)^\top$ and covariance matrix $\boldsymbol{\Sigma}$ if its probability density function is

$$\phi_d(\mathbf{y} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left[\frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \right] \exp \left\{ -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}) \right\}, \quad (7.15)$$

where $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$. The quantity in square brackets is a constant that normalizes the density so that it integrates to 1. The density depends on \mathbf{y} only through $(\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})$, and so the density is constant on each ellipse $\{\mathbf{y} : (\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}) = c\}$. Here $c > 0$ is a fixed constant that determines the size of the ellipse, with larger values of c giving larger ellipses, each centered at $\boldsymbol{\mu}$. Such densities are called *elliptically contoured*.

Figure 7.2 has contour plots of bivariate normal densities. Both Y_1 and Y_2 are $N(0, 1)$ and the correlation between Y_1 and Y_2 is 0.5 in panel (a) or -0.95 in panel (b). Notice how the orientations of the contours depend on the sign and magnitude of the correlation. In panel (a) we can see that the height of the density is constant on ellipses and decreases with the distance from the mean, which is $(0, 0)$. The same behavior occurs in panel (b), but, because of the high correlation, the contours are so close together that it was not possible to label them.

If $\mathbf{Y} = (Y_1, \dots, Y_d)^T$ has a multivariate normal distribution, then for every set of constants $\mathbf{c} = (c_1, \dots, c_d)^T$, the weighted average (linear combination) $\mathbf{c}^T \mathbf{Y} = c_1 Y_1 + \dots + c_d Y_d$ has a normal distribution with mean $\mathbf{c}^T \boldsymbol{\mu}$ and variance $\mathbf{c}^T \boldsymbol{\Sigma} \mathbf{c}$. In particular, the marginal distribution of Y_i is $N(\mu_i, \sigma_i^2)$, where σ_i^2 is the i th diagonal element of $\boldsymbol{\Sigma}$ —to see this, take $c_i = 1$ and $c_j = 0$ for $j \neq i$.

The assumption of multivariate normality facilitates many useful probability calculations. If the returns on a set of assets have a multivariate normal distribution, then the return on any portfolio formed from these assets will be normally distributed. This is because the return on the portfolio is the weighted average of the returns on the assets. Therefore, the normal distribution could be used, for example, to find the probability of a loss of some size of interest, say, 10% or more, on the portfolio. Such calculations have important applications in finding a value-at-risk; see Chap. 19.

Unfortunately, we saw in Chap. 5 that often individual returns are not normally distributed, which implies that a vector of returns will not have a multivariate normal distribution. In Sect. 7.6 we will look at an important class of heavy-tailed multivariate distributions.

7.6 The Multivariate t -Distribution

We have seen that the univariate t -distribution is a good model for the returns of individual assets. Therefore, it is desirable to have a model for vectors of returns such that the univariate marginals are t -distributed. The multivariate t -distribution has this property. The random vector \mathbf{Y} has a multivariate $t_\nu(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ distribution if

$$\mathbf{Y} = \boldsymbol{\mu} + \sqrt{\frac{\nu}{W}} \mathbf{Z}, \quad (7.16)$$

where W is chi-squared distributed with ν degrees of freedom, \mathbf{Z} is $N_d(0, \boldsymbol{\Lambda})$ distributed, and W and \mathbf{Z} are independent. Thus, the multivariate t -distribution is a continuous scale mixture of multivariate normal distributions. Extreme values of \mathbf{Y} tend to occur when W is near zero. Since $W^{-1/2}$ multiplies all components of \mathbf{Z} , outliers in one component tend to occur with outliers in other components, that is, there is tail dependence.

For $\nu > 1$, $\boldsymbol{\mu}$ is the mean vector of \mathbf{Y} . For $0 < \nu \leq 1$, the expectation of \mathbf{Y} does not exist, but $\boldsymbol{\mu}$ can still be regarded as the “center” of the distribution

of \mathbf{Y} because, for any value of ν , the vector $\boldsymbol{\mu}$ contains the medians of the components of \mathbf{Y} and the contours of the density of \mathbf{Y} are ellipses centered at $\boldsymbol{\mu}$. Also, $\boldsymbol{\mu}$ is the mode of the distribution, that is, the location where the density is maximized.

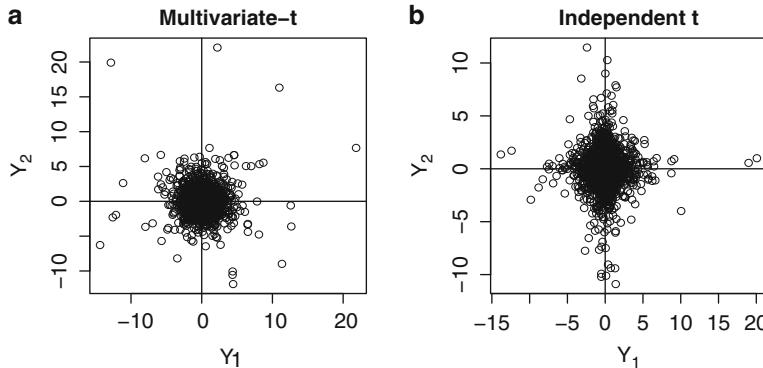


Fig. 7.3. (a) Plot of a random sample from a bivariate t -distribution with $\nu = 3$, $\boldsymbol{\mu} = (0 \ 0)^T$ and identity covariate matrix. (b) Plot of a random sample of pairs of independent $t_3(0, 1)$ random variables. Both sample sizes are 2,500.

For $\nu > 2$, the covariance matrix of \mathbf{Y} exists and is

$$\boldsymbol{\Sigma} = \frac{\nu}{\nu - 2} \boldsymbol{\Lambda}. \quad (7.17)$$

We will call $\boldsymbol{\Lambda}$ the *scale matrix*. The scale matrix exists for all values of ν . Since the covariance matrix $\boldsymbol{\Sigma}$ of \mathbf{Y} is just a multiple of the covariance matrix $\boldsymbol{\Lambda}$ of \mathbf{Z} , \mathbf{Y} and \mathbf{Z} have the same correlation matrices, assuming $\nu > 2$ so that the correlation matrix of \mathbf{Y} exists. If $\Sigma_{i,j} = 0$, then Y_i and Y_j are uncorrelated, but they are dependent, nonetheless, because of the tail dependence. Tail dependence is illustrated in Fig. 7.3, where panel (a) is a plot of 2500 observations from an uncorrelated bivariate t -distribution with marginal distributions that are $t_3(0, 1)$. For comparison, panel (b) is a plot of 2500 observations of pairs of independent $t_3(0, 1)$ random variables—these pairs do not have a bivariate t -distribution. Notice that in (b), outliers in Y_1 are not associated with outliers in Y_2 , since the outliers are concentrated near the x - and y -axes. In contrast, outliers in (a) are distributed uniformly in all directions. The univariate marginal distributions are the same in (a) and (b).

Tail dependence can be expected in equity returns. For example, on Black Monday, almost all equities had extremely large negative returns. Of course, Black Monday was an extreme, even among extreme events. We would not want to reach any general conclusions based upon Black Monday alone. However, in Fig. 7.1, we see little evidence that outliers are concentrated along the axes, with the possible exception of the scatterplot for IBM and Mobil. As another example of dependencies among stock returns, Fig. 7.4 contains a

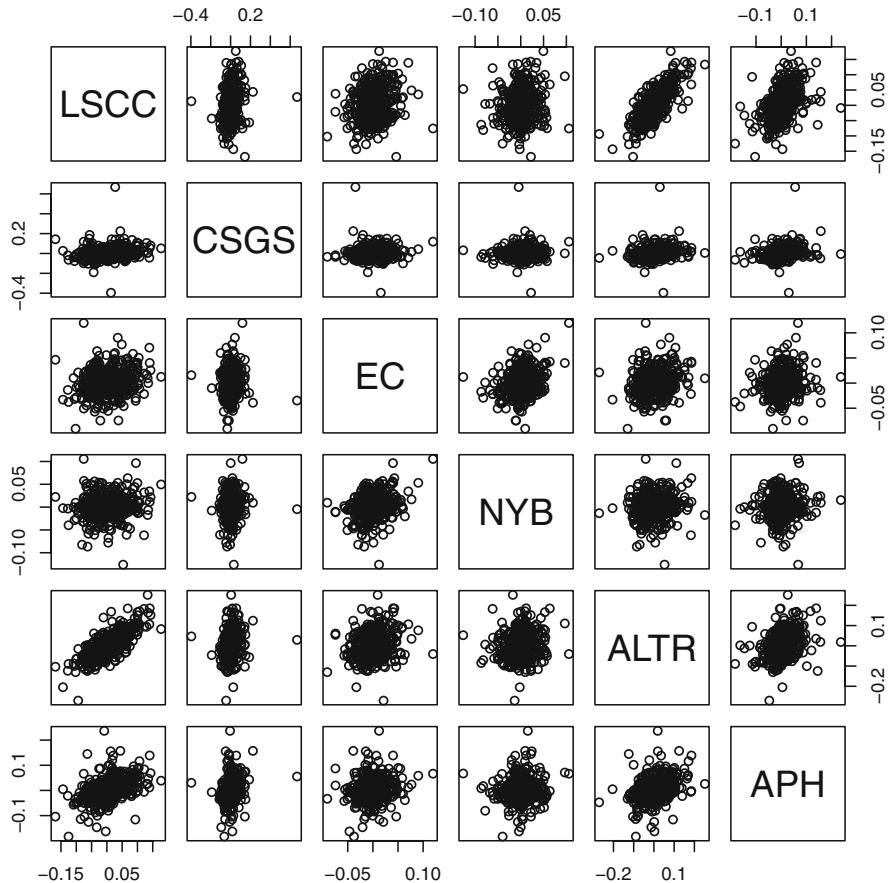


Fig. 7.4. Scatterplot matrix of 500 daily returns on six midcap stocks in R’s `midcapD.ts` data set.

scatterplot matrix of returns on six midcap stocks in the `midcapD.ts` data set. Again, tail dependence can be seen. This suggests that tail dependence is common among equity returns and the multivariate t -distribution is a promising model for them.

7.6.1 Using the t -Distribution in Portfolio Analysis

If Y has a $t_\nu(\mu, \Lambda)$ distribution, which we recall has covariance matrix $\Sigma = \{\nu/(\nu - 2)\}\Lambda$, and w is a vector of weights, then $w^\top Y$ has a univariate t -distribution with mean $w^\top \mu$ and variance $\{\nu/(\nu - 2)\}w^\top \Lambda w = w^\top \Sigma w$. This fact can be useful when computing risk measures for a portfolio. If the returns on the assets have a multivariate t -distribution, then the return on the portfolio will have a univariate t -distribution. We will make use of this result in Chap. 19.

7.7 Fitting the Multivariate t -Distribution by Maximum Likelihood

To estimate the parameters of a multivariate t -distribution, one can use the function `cov.trob` in R's `MASS` package. This function computes the maximum likelihood estimates of μ and Λ with ν fixed. To estimate ν , one computes the profile log-likelihood for ν and finds the value, $\hat{\nu}$ of ν that maximizes the profile log-likelihood. Then the MLEs of μ and Λ are the estimates from `cov.trob` with ν fixed at $\hat{\nu}$.

Example 7.4. Fitting the CRSPday data

This example uses the data set `CRSPday` analyzed earlier in Example 7.1. Recall that there are four variables, returns on GE, IBM, Mobil, and the CRSP index. The profile log-likelihood is plotted in Fig. 7.5. In that figure, one can see that the MLE of ν is 5.94, and there is relatively little uncertainty about this parameter's value—the 95 % profile likelihood confidence interval is (5.41, 6.55). The code to create this figure is below.

```
library(mnormt)
library(MASS)
data(CRSPday, package = "Ecdat")
```

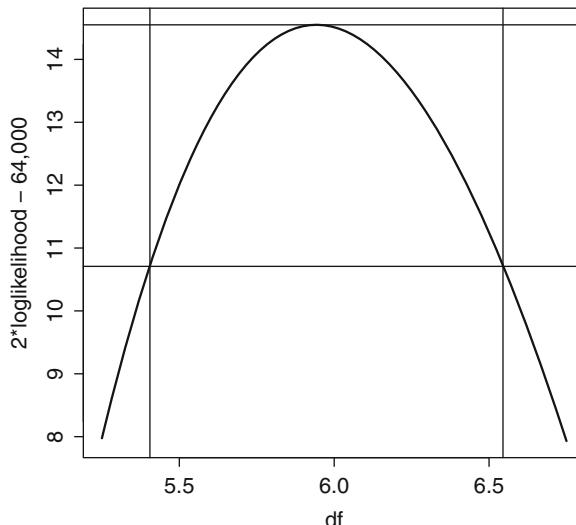


Fig. 7.5. CRSPday data. A profile likelihood confidence interval for ν . The solid curve is $2L_{\max}(\nu)$, where $L_{\max}(\nu)$ is the profile likelihood minus 32,000. 32,000 was subtracted from the profile likelihood to simplify the labeling of the y-axis. The horizontal line intersects the y-axis at $2L_{\max}(\hat{\nu}) - \chi^2_{\alpha,1}$, where $\hat{\nu}$ is the MLE and $\alpha = 0.05$. All values of ν such that $2L_{\max}(\nu)$ is above the horizontal line are in the profile likelihood 95 % confidence interval. The two vertical lines intersect the x-axis at 5.41 and 6.55, the endpoints of the confidence interval.

```

dat =CRSPday[, 4:7]
df = seq(5.25, 6.75, 0.01)
n = length(df)
loglik = rep(0,n)
for(i in 1:n){
  fit = cov.trob(dat,nu=df)
  loglik[i] = sum(log(dmt(dat, mean=fit$center,
    S = fit$cov, df = df[i])))
}
aic_t = -max(2 * loglik) + 2 * (4 + 10 + 1) + 64000
z1 = (2 * loglik > 2 * max(loglik) - qchisq(0.95, 1))
plot(df, 2 * loglik - 64000, type = "l", cex.axis = 1.5,
  cex.lab = 1.5, ylab = "2 * loglikelihood - 64,000", lwd = 2)
abline(h = 2 * max(loglik) - qchisq(0.95, 1) - 64000)
abline(h = 2 * max(loglik) - 64000)
abline(v = (df[16] + df[17]) / 2)
abline(v = (df[130] + df[131]) / 2)

```

AIC for this model is 15.45 plus 64,000. Here AIC values are expressed as deviations from 64,000 to keep these values small. This is helpful when comparing two or more models via AIC. Subtracting the same constant from all AIC values, of course, has no effect on model comparisons.

The maximum likelihood estimates of the mean vector and the correlation matrix are called `$center` and `$cor`, respectively, in the following output:

```

$center
[1] 0.0009424 0.0004481 0.0006883 0.0007693

$cor
 [,1]   [,2]   [,3]   [,4]
[1,] 1.0000 0.3192 0.2845 0.6765
[2,] 0.3192 1.0000 0.1584 0.4698
[3,] 0.2845 0.1584 1.0000 0.4301
[4,] 0.6765 0.4698 0.4301 1.0000

```

These estimates were computed using `cov.trob` with ν fixed at 6.

When the data are *t*-distributed, the maximum likelihood estimates are superior to the sample mean and covariance matrix in several respects—the MLE is less variable and it is less sensitive to outliers. However, in this example, the maximum likelihood estimates are similar to the sample mean and correlation matrix. For example, the sample correlation matrix is

```

      ge     ibm    mobil    crsp
ge     1.0000 0.3336 0.2972 0.7148
ibm    0.3336 1.0000 0.1587 0.4864
mobil  0.2972 0.1587 1.0000 0.4294
crsp   0.7148 0.4864 0.4294 1.0000

```



7.8 Elliptically Contoured Densities

The multivariate normal and t -distributions have *elliptically contoured* densities, a property that will be discussed in this section. A d -variate multivariate density f is elliptically contoured if it can be expressed as

$$f(\mathbf{y}) = |\Lambda|^{-1/2} g \{ (\mathbf{y} - \boldsymbol{\mu})^\top \Lambda^{-1} (\mathbf{y} - \boldsymbol{\mu}) \}, \quad (7.18)$$

where g is a nonnegative-valued function such that $1 = \int_{\mathbb{R}^d} g(\|\mathbf{y}\|^2) d\mathbf{y}$, $\boldsymbol{\mu}$ is a $d \times 1$ vector, and Λ is a $d \times d$ symmetric, positive definite matrix. Usually, $g(x)$ is a decreasing function of $x \geq 0$, and we will assume this is true. We will also assume the finiteness of second moments, in which case $\boldsymbol{\mu}$ is the mean vector and the covariance matrix Σ is a scalar multiple of Λ .

For each fixed $c > 0$,

$$\mathcal{E}(c) = \{ \mathbf{y} : (\mathbf{y} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu}) = c \}$$

is an ellipse centered at $\boldsymbol{\mu}$, and if $c_1 > c_2$, then $\mathcal{E}(c_1)$ is inside $\mathcal{E}(c_2)$ because g is decreasing. The contours of f are concentric ellipses as can be seen in Fig. 7.6.

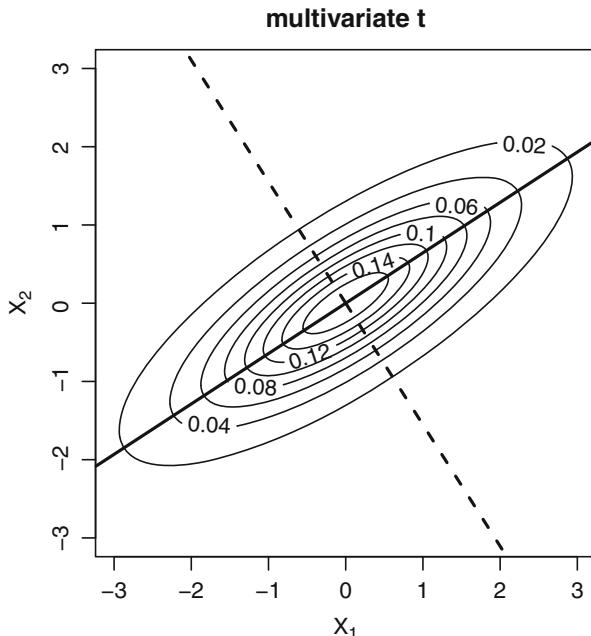


Fig. 7.6. Contour plot of a multivariate t_4 -density with $\boldsymbol{\mu} = (0, 0)^\top$, $\sigma_1^2 = 2$, $\sigma_2^2 = 1$, and $\sigma_{12} = 1.1$.

That figure shows the contours of the bivariate t_4 -density with $\mu = (0, 0)^\top$ and

$$\Sigma = \begin{pmatrix} 2 & 1.1 \\ 1.1 & 1 \end{pmatrix}.$$

The major axis of the ellipses is a solid line and the minor axis is a dashed line.

How can the axes be found? From Appendix A.20, we know that Σ has an *eigenvalue-eigenvector decomposition*

$$\Sigma = O \operatorname{diag}(\lambda_i) O^\top,$$

where O is an orthogonal matrix whose columns are the eigenvectors of Σ and $\lambda_1, \dots, \lambda_d$ are the eigenvalues of Σ .

The columns of O determine the axes of the ellipse $\mathcal{E}(c)$. The decomposition can be found in R using the function `eigen()` and, for the matrix Σ in the example, the decomposition is

```
$values
[1] 2.708 0.292
```

which gives the eigenvalues, and

```
$vectors
[,1]   [,2]
[1,] -0.841  0.541
[2,] -0.541 -0.841
```

which has the corresponding eigenvectors as columns; e.g., $(-0.841, -0.541)$ is an eigenvector with eigenvalue 2.708. The eigenvectors are normalized so have norm equal to 1. Nonetheless, the eigenvectors are only determined up to a sign change, so the first eigenvector could be taken as $(-0.841, -0.541)$, as in the R output, or $(0.841, 0.541)$.

If o_i is the i th column of O , the i th axis of $\mathcal{E}(c)$ goes through the points μ and $\mu + o_i$. Therefore, this axis is the line

$$\{\mu + k o_i : -\infty < k < \infty\}.$$

Because O is an orthogonal matrix, the axes are mutually perpendicular. The axes can be ordered according to the size of the corresponding eigenvalues. In the bivariate case the axis associated with the largest (smallest) eigenvalue is the major (minor) axis. We are assuming that there are no ties among the eigenvalues.

Since $\mu = 0$, in our example the major axis is $k(0.841, 0.541)$, $-\infty < k < \infty$, and the minor axis is $k(0.541, -0.841)$, $-\infty < k < \infty$.

When there are ties among the eigenvalues, the eigenvectors are not unique and the analysis is somewhat more complicated and will not be discussed in detail. Instead two examples will be given. In the bivariate case if $\Sigma = I$, the contours are circles and there is no unique choice of the axes—any pair of perpendicular vectors will do. As a trivariate example, if $\Sigma = \operatorname{diag}(1, 1, 3)$, then the first principle axis is $(0, 0, 1)$ with eigenvalue 3. The second and third

principal axis can be any perpendicular pair of vectors with third coordinates equal to 0. The `eigen()` function in R returns $(0,1,0)$ and $(1,0,0)$ as the second and third axes.

7.9 The Multivariate Skewed t -Distributions

Azzalini and Capitanio (2003) have proposed a skewed extension of the multivariate t -distribution. The univariate special case was discussed in Sect. 5.7. In the multivariate case, in addition to the shape parameter ν determining tail weight, the skewed t -distribution has a vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)^T$ of shape parameters determining the amounts of skewness in the components of the distribution. If \mathbf{Y} has a skewed t -distribution, then Y_i is left-skewed, symmetric, or right-skewed depending on whether $\alpha_i < 0$, $\alpha_i = 0$, or $\alpha_i > 0$.

Figure 7.7 is a contour plot of a bivariate skewed t -distribution with $\boldsymbol{\alpha} = (-1, 0.25)^T$ and $df = 4$. Notice that, because α_1 is reasonably large and negative, Y_1 has a considerable amount of left skewness, as can be seen in the contours, which are more widely spaced on the left side of the plot compared to the right. Also, Y_2 shows a lesser amount of right skewness, since the contours on top are slightly more widely spaced than on the bottom. This feature is to be expected since α_2 is positive but with a relatively small absolute value.

multivariate skewed t: $\alpha_1 = -1, \alpha_2 = 0.25$

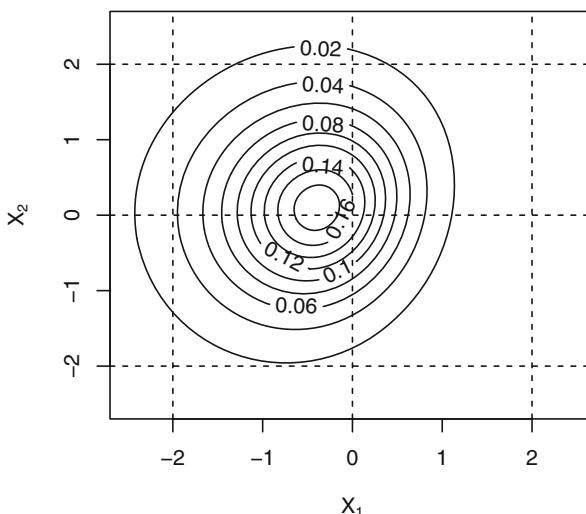


Fig. 7.7. Contours of a bivariate skewed t -density. The contours are more widely spaced on the left compared to the right because X_1 is left-skewed. Similarly, the contours are more widely spaced on the top compared to the bottom because X_2 is right-skewed, but the skewness of X_2 is relatively small and less easy to see.

Example 7.5. Fitting the skewed t -distribution to CRSPday

We now fit the skewed t -model to the CRSPday data set using the function `mst.mple()` in R's `sn` package. This function maximizes the likelihood over all parameters, so there is no need to use the profile likelihood as with `cov.trob()`. The code is below.

```
library(sn)
data(CRSPday, package = "Ecdat")
dat = CRSPday[, 4:7]
fit = mst.mple(y = dat, penalty = NULL)
aic_skewt = -2 * fit$logL + 64000 + 2 * (4 + 10 + 4 + 1)
dp2cp(fit$dp, "st")
aic_skewt
```

The CP estimates are as follows.

```
> dp2cp(fit$dp, "st")
$beta
      ge          ibm         mobil        crsp
[1,] 0.0009459182 0.0004521179 0.0006917701 0.0007722816

$var.cov
      ge          ibm         mobil        crsp
ge  1.899520e-04 7.252242e-05 5.185778e-05 6.957078e-05
ibm 7.252242e-05 2.743354e-04 3.492763e-05 5.771567e-05
mobil 5.185778e-05 3.492763e-05 1.749708e-04 4.238468e-05
crsp 6.957078e-05 5.771567e-05 4.238468e-05 5.565159e-05

$gamma1
      ge          ibm         mobil        crsp
0.0010609438 0.0012389968 0.0007125122 0.0009920253

$gamma2M
[1] 25.24996
```

Here `$beta` is the estimate of the means, `$var.cov` is the estimate of covariance matrix, `$gamma1` is the estimate of skewnesses, and `$gamma2M` estimates the common kurtosis of the four marginal distributions. The DP estimates are in `fit$dp` but are of less interest so are not included here.

AIC for the skewed t -model is 23.47885 (plus 64,000), larger than 15.45, the AIC found in Example 7.4 for the symmetric t -model. This result suggests that the symmetric t -model is adequate for this data set.

In summary, the CRSPday data are well fit by a symmetric t -distribution and no need was found for using a skewed t -distribution. Also, in the normal plots of the four variables in Fig. 7.8, heavy tails are evident but there are no signs of serious skewness. Although this might be viewed as a negative result, since we have not found an improvement in fit by going to the more

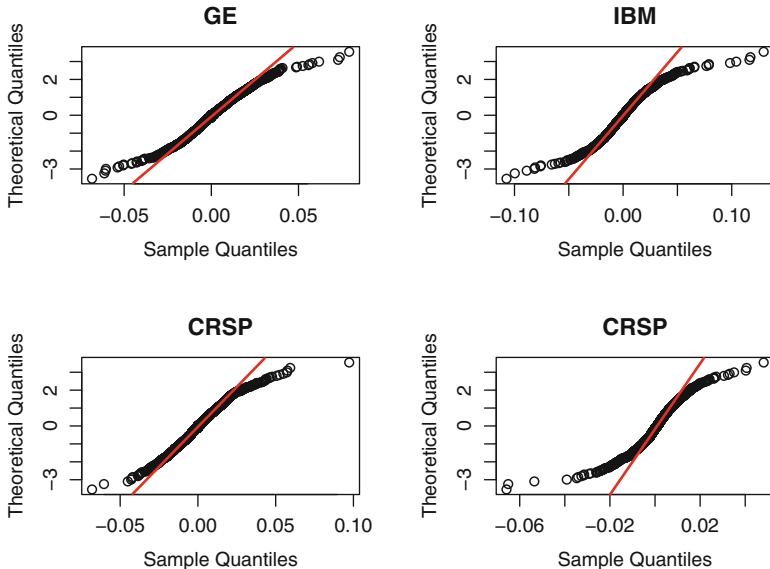


Fig. 7.8. Normal plots of the four returns series in the CRSPday data set. The reference lines go through the first and third quartiles.

flexible skewed t -distribution, the result does give us more confidence that the symmetric t -distribution is suitable for modeling this data set. \square

7.10 The Fisher Information Matrix

In the discussion of Fisher information in Sect. 5.10, θ was assumed to be one-dimensional. If θ is an m -dimensional parameter vector, then the Fisher information matrix is an $m \times m$ square matrix, \mathcal{I} , and is equal the matrix of expected second-order partial derivatives of $-\log\{L(\theta)\}$.¹ In other words, the i, j th entry of the Fisher information matrix is

$$\mathcal{I}_{ij}(\theta) = -E \left[\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log\{L(\theta)\} \right]. \quad (7.19)$$

The standard errors are the square roots of the diagonal entries of the inverse of the Fisher information matrix. Thus, the standard error for θ_i is

$$s_{\hat{\theta}_i} = \sqrt{\{\mathcal{I}(\hat{\theta})^{-1}\}_{ii}}. \quad (7.20)$$

¹ The matrix of second partial derivatives of a function is called its *Hessian matrix*, so the Fisher information matrix is the expectation of the Hessian of the negative log-likelihood.

In the case of a single parameter, (7.20) reduces to (5.19). The central limit theorem for the MLE in Sect. 5.10 generalizes to the following multivariate version.

Result 7.6. *Under suitable assumptions, for large enough sample sizes, the maximum likelihood estimator is approximately normally distributed with mean equal to the true parameter vector and with covariance matrix equal to the inverse of the Fisher information matrix.*

Computation of the expectation in $\mathcal{I}(\boldsymbol{\theta})$ can be challenging. Programming the second derivatives can be difficult as well, especially for complex models. In practice, the observed Fisher information matrix, whose i, j th element is

$$\mathcal{I}_{ij}^{\text{obs}}(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \log\{L(\boldsymbol{\theta})\} \quad (7.21)$$

is often used. The observed Fisher information matrix is, of course, the multivariate analog of (5.21). Using observed information obviates the need to compute the expectation. Moreover, the Hessian matrix can be computed numerically by finite differences, for example, using R's `fdHess()` function in the `nlme` package. Also, as demonstrated in several examples in Chap. 5, if `hessian=TRUE` in the call to `optim()`, then the Hessian matrix is returned when the negative log-likelihood is minimized by that function.

Inverting the observed Fisher information matrix computed by finite differences is the most commonly used method for obtaining standard errors. The advantage of this approach is that only the computation of the log-likelihood is necessary, and of course this computation is necessary simply to compute the MLE.

The key point is that there is an explicit method of calculating standard errors for maximum likelihood estimators. The calculation of standard errors of maximum likelihood estimators by computing and then inverting the observed Fisher information matrix is routinely programmed into statistical software, e.g., by the R function `fitdistr()` used to fit univariate distributions.

7.11 Bootstrapping Multivariate Data

When resampling multivariate data, the dependencies within the observation vectors need to be preserved. Let the vectors $\mathbf{Y}_1, \dots, \mathbf{Y}_n$ be an i.i.d. sample of multivariate data. In model-free resampling, the vectors $\mathbf{Y}_1, \dots, \mathbf{Y}_n$ are sampled with replacement. There is no resampling of the components within a vector. Resampling within vectors would make their components mutually independent and would not mimic the actual data where the components are dependent. Stated differently, if the data are in a spreadsheet (or matrix) with rows corresponding to observations and columns to variables, then one samples entire rows.

Model-based resampling simulates vectors from the multivariate distribution of the \mathbf{Y}_i , for example, from a multivariate t -distribution with the mean vector, covariance matrix, and degrees of freedom equal to the MLEs.

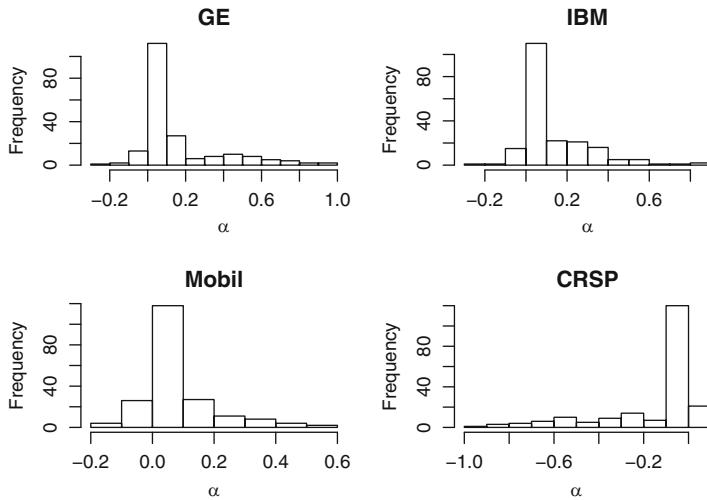


Fig. 7.9. Histograms of 200 bootstrapped values of $\hat{\alpha}$ for each of the returns series in the CRSPday data set.

Example 7.7. Bootstrapping the skewed t fit to CRSPday

In Example 7.5 the skewed t -model was fit to the CRSPday data. This example continues that analysis by bootstrapping the estimator of α for each of the four returns series. Histograms of 200 bootstrap values of $\hat{\alpha}$ are found in Fig. 7.9. Bootstrap percentile 95 % confidence intervals include 0 for all four stocks, so there is no strong evidence of skewness in any of the returns series.

Despite the large sample size of 2,528, the estimators of α do not appear to be normally distributed. We can see in Fig. 7.9 that they are right-skewed for the three stocks and left-skewed for the CRSP returns. The distribution of $\hat{\alpha}$ also appears heavy-tailed. The excess kurtosis coefficient of the 200 bootstrap values of $\hat{\alpha}$ is 2.38, 1.33, 3.18, and 2.38 for the four series.

The central limit theorem for the MLE guarantees that $\hat{\alpha}$ is nearly normally distributed for sufficiently large samples, but this theorem does not tell us how large the sample size must be. Fortunately, the sample size needed for near normality is often small, but there are exceptions. We see in this example that in such cases the sample size must be very large indeed since 2,528 is not large enough. This is a major reason for preferring to construct confidence intervals using the bootstrap rather than a normal approximation.

A bootstrap sample of the returns was drawn with the following R code. The returns are in the matrix `dat` and `yboot` is a bootstrap sample chosen by taking a random sample of the rows of `dat`, with replacement of course.

```
yboot = dat[sample((1:n), n, replace = TRUE), ]
```

□

7.12 Bibliographic Notes

The multivariate central limit theorem for the MLE is stated precisely and proved in textbooks on asymptotic theory such as Lehmann (1999) and van der Vaart (1998). The multivariate skewed t -distribution is in Azzalini and Capitanio (2003) and Azzalini (2014).

7.13 R Lab

7.13.1 Equity Returns

This section uses the data set `berndtInvest` on the book's web site and taken originally from R's `fEcofin` package. This data set contains monthly returns from January 1, 1987, to December 1, 1987, on 16 equities. There are 18 columns. The first column is the date and the last is the risk-free rate.

In the lab we will only use the first four equities. The following code computes the sample covariance and correlation matrices for these returns.

```
berndtInvest = read.csv("berndtInvest.csv")
Berndt = as.matrix(berndtInvest[, 2:5])
cov(Berndt)
cor(Berndt)
```

If you wish, you can also plot a scatterplot matrix with the following R code.

```
pairs(Berndt)
```

Problem 1 Suppose the four variables being used are denoted by X_1, \dots, X_4 . Use the sample covariance matrix to estimate the variance of $0.5X_1 + 0.3X_2 + 0.2X_3$. (Useful R facts: “`t(a)`” is the transpose of a vector or matrix `a` and “`a %*% b`” is the matrix product of `a` and `b`.)

Fit a multivariate- t model to the data using the function `cov.trob` in the `MASS` package. This function computes the MLE of the mean and covariance matrix with a fixed value of ν . To find the MLE of ν , the following code computes the profile log-likelihood for ν .

```

library(MASS) # needed for cov.trob
library(mnormt) # needed for dmt
df = seq(2.5, 8, 0.01)
n = length(df)
loglik_profile = rep(0, n)
for(i in 1:n)
{
  fit = cov.trob(Berndt, nu = df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow = 4)
  loglik_profile[i] = sum(log(dmt(Berndt, mean = fit$center,
    S= f it$cov, df = df[i])))
}

```

Problem 2 Using the results produced by the code above, find the MLE of ν and a 90 % profile likelihood confidence interval for ν . Include your R code with your work. Also, plot the profile log-likelihood and indicate the MLE and the confidence interval on the plot.

Section 7.13.3 demonstrates how the MLE for a multivariate t -model can be fit directly with the `optim` function, rather than `profile likelihood`.

7.13.2 Simulating Multivariate t -Distributions

The following code generates and plots four bivariate samples. Each sample has univariate marginals that are standard t_3 -distributions. However, the dependencies are different.

```

library(MASS) # need for mvtnorm
par(mfrow=c(1,4))
N = 2500
nu = 3

set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1), nrow = 2)
x= mvtnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w, w)
plot(x, main = "(a)")

set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1),nrow = 2)
x= mvtnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w1, w2)
plot(x, main = "(b)")

```

```

set.seed(5640)
cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvtnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w1, w2)
plot(x, main = "(c)")

set.seed(5640)
cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvtnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x = x * cbind(w, w)
plot(x, main = "(d)")

```

Note the use of these R commands: `set.seed` to set the seed of the random number generator, `mvtnorm` to generate multivariate normally distributed vectors, `rchisq` to generate χ^2 -distributed random numbers, `cbind` to bind together vectors as the columns of a matrix, and `matrix` to create a matrix from a vector. In R, “`a * b`” is elementwise multiplication of same-size matrices `a` and `b`, and “`a %*% b`” is matrix multiplication of conforming matrices `a` and `b`.

Problem 3 Which sample has independent variates? Explain your answer.

Problem 4 Which sample has variates that are correlated but do not have tail dependence? Explain your answer.

Problem 5 Which sample has variates that are uncorrelated but with tail dependence? Explain your answer.

Problem 6* Suppose that (X, Y) are the returns on two assets and have a multivariate t -distribution with degrees of freedom, mean vector, and covariance matrix

$$\nu = 5, \quad \mu = \begin{pmatrix} 0.001 \\ 0.002 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 0.10 & 0.03 \\ 0.03 & 0.15 \end{pmatrix}.$$

Then $R = (X + Y)/2$ is the return on an equally weighted portfolio of the two assets.

- (a) What is the distribution of R ?
- (b) Write an R program to generate a random sample of size 10,000 from the distribution of R . Your program should also compute the 0.01 upper quantile of this sample and the sample average of all returns that exceed this quantile. This quantile and average will be useful later when we study risk analysis.

7.13.3 Fitting a Bivariate t -Distribution

When you run the R code that follows this paragraph, you will compute the MLE for a bivariate t -distribution fit to CRSP returns data. A challenge when fitting a multivariate distribution is enforcing the constraint that the scale matrix (or the covariance matrix) must be positive definite. One way to meet this challenge is to let the scale matrix be $A^T A$, where A is an upper triangular matrix. (It is easy to show that $A^T A$ is positive semidefinite if A is any square matrix. Because a scale or covariance matrix is symmetric, only the entries on and above the main diagonal are free parameters. In order for A to have the same number of free parameters as the covariance matrix, we restrict A to be upper triangular.)

```
library(mnormt)
data(CRSPday, package = "Ecdat")
Y = CRSPday[, c(5, 7)]
loglik = function(par)
{
  mu = par[1:2]
  A = matrix(c(par[3], par[4], 0, par[5]), nrow = 2, byrow = T)
  scale_matrix = t(A) %*% A
  df = par[6]
  -sum(log(dmt(Y, mean = mu, S = scale_matrix, df = df)))
}
A = chol(cov(Y))
start = as.vector(c(apply(Y, 2, mean),
  A[1, 1], A[1, 2], A[2, 2], 4))
fit_mvt = optim(start, loglik, method = "L-BFGS-B",
  lower = c(-0.02, -0.02, -0.1, -0.1, -0.1, 2),
  upper = c(0.02, 0.02, 0.1, 0.1, 0.1, 15), hessian = T)
```

Problem 7* Let $\boldsymbol{\theta} = (\mu_1, \mu_2, A_{1,1}, A_{1,2}, A_{2,2}, \nu)$, where μ_j is the mean of the j th variable, $A_{1,1}$, $A_{1,2}$, and $A_{2,2}$ are the nonzero elements of A , and ν is the degrees-of-freedom parameter.

- What does the code $A = \text{chol}(\text{cov}(Y))$ do?
- Find $\hat{\boldsymbol{\theta}}_{\text{ML}}$, the MLE of $\boldsymbol{\theta}$.
- Find the Fisher information matrix for $\boldsymbol{\theta}$. (Hint: The Hessian is part of the object `fit_mvt`. Also, the R function `solve` will invert a matrix.)
- Find the standard errors of the components of $\hat{\boldsymbol{\theta}}_{\text{ML}}$ using the Fisher information matrix.
- Find the MLE of the covariance matrix of the returns.
- Find the MLE of ρ , the correlation between the two returns (Y_1 and Y_2).

7.14 Exercises

1. Suppose that $E(X) = 1$, $E(Y) = 1.5$, $\text{Var}(X) = 2$, $\text{Var}(Y) = 2.7$, and $\text{Cov}(X, Y) = 0.8$.
 - (a) What are $E(0.2X + 0.8Y)$ and $\text{Var}(0.2X + 0.8Y)$?
 - (b) For what value of w is $\text{Var}\{wX + (1-w)Y\}$ minimized? Suppose that X is the return on one asset and Y is the return on a second asset. Why would it be useful to minimize $\text{Var}\{wX + (1-w)Y\}$?
2. Let X_1 , X_2 , Y_1 , and Y_2 be random variables.
 - (a) Show that $\text{Cov}(X_1 + X_2, Y_1 + Y_2) = \text{Cov}(X_1, Y_1) + \text{Cov}(X_1, Y_2) + \text{Cov}(X_2, Y_1) + \text{Cov}(X_2, Y_2)$.
 - (b) Generalize part (a) to an arbitrary number of X_i s and Y_i s.
3. Verify formulas (A.24)–(A.27).
4. (a) Show that

$$E\{X - E(X)\} = 0$$

for any random variable X .

- (b) Use the result in part (a) and Eq. (A.31) to show that if two random variables are independent then they are uncorrelated.
5. Show that if X is uniformly distributed on $[-a, a]$ for any $a > 0$ and if $Y = X^2$, then X and Y are uncorrelated but they are not independent.
6. Verify the following results that were stated in Sect. 7.3:

$$E(\mathbf{w}^\top \mathbf{X}) = \mathbf{w}^\top \{E(\mathbf{X})\}$$

and

$$\begin{aligned} \text{Var}(\mathbf{w}^\top \mathbf{X}) &= \sum_{i=1}^N \sum_{j=1}^N w_i w_j \text{Cov}(X_i, X_j) \\ &= \text{Var}(\mathbf{w}^\top \mathbf{X}) \mathbf{w}^\top \text{COV}(\mathbf{X}) \mathbf{w}. \end{aligned}$$

7. Suppose $\mathbf{Y} = (Y_1, Y_2, Y_3)$ has covariance matrix

$$\text{COV}(\mathbf{Y}) = \begin{pmatrix} 1.0 & 0.9 & a \\ 0.9 & 1.0 & 0.9 \\ a & 0.9 & 1.0 \end{pmatrix}$$

for some unknown value a . Use Eq. (7.7) and the fact that the variance of a random variable is always ≥ 0 to show that a cannot equal 0.

References

- Azzalini, A. (2014) *The Skew-Normal and Related Families (Institute of Mathematical Statistics Monographs, Book 3)*, Cambridge University Press.
- Azzalini, A., and Capitanio, A. (2003) Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t distribution. *Journal of the Royal Statistics Society, Series B*, **65**, 367–389.
- Lehmann, E. L. (1999) *Elements of Large-Sample Theory*, Springer-Verlag, New York.
- van der Vaart, A. W. (1998) *Asymptotic Statistics*, Cambridge University Press, Cambridge.

Copulas

8.1 Introduction

Copulas are a popular framework for both defining multivariate distributions and modeling multivariate data. A copula characterizes the dependence—and only the dependence—between the components of a multivariate distribution; they can be combined with any set of univariate marginal distributions to form a full joint distribution. Consequently, the use of copulas allows us to take advantage of the wide variety of univariate models that are available.

The primary financial application of copula models is risk assessment and management of portfolios that contain assets which exhibit co-movements in extreme behavior. For example, a pair of assets may have weakly correlated returns, but their largest losses may tend to occur in the same periods. They are commonly applied to portfolios of loans, bonds, and collateralized debt obligations (CDOs). Their misapplication in finance is also well-documented, as referenced in Sect. 8.8.

A *copula* is a multivariate CDF whose univariate marginal distributions are all $\text{Uniform}(0,1)$. Suppose that $\mathbf{Y} = (Y_1, \dots, Y_d)$ has a multivariate CDF $F_{\mathbf{Y}}$ with continuous marginal univariate CDFs F_{Y_1}, \dots, F_{Y_d} . Then, by Eq. (A.9) in Appendix A.9.2, each of $F_{Y_1}(Y_1), \dots, F_{Y_d}(Y_d)$ is distributed $\text{Uniform}(0,1)$. Therefore, the CDF of $\{F_{Y_1}(Y_1), \dots, F_{Y_d}(Y_d)\}$ is a copula. This CDF is called the copula of \mathbf{Y} and denoted by C_Y . C_Y contains all information about dependencies among the components of \mathbf{Y} but has no information about the marginal CDFs of \mathbf{Y} .

It is easy to find a formula for C_Y . To avoid technical issues, in this section we will assume that all random variables have continuous, strictly increasing CDFs. More precisely, the CDFs are assumed to be increasing on their support. For example, the standard exponential CDF

$$F(y) = \begin{cases} 1 - e^{-y}, & y \geq 0, \\ 0, & y < 0, \end{cases}$$

has support $[0, \infty)$ and is strictly increasing on that set. The assumption that the CDF is continuous and strictly increasing is reasonable in many financial applications, but it is avoided in more mathematically advanced texts; see Sect. 8.8.

Since C_Y is the CDF of $\{F_{Y_1}(Y_1), \dots, F_{Y_d}(Y_d)\}$, by the definition of a CDF we have

$$\begin{aligned} C_Y(u_1, \dots, u_d) &= P\{F_{Y_1}(Y_1) \leq u_1, \dots, F_{Y_d}(Y_d) \leq u_d\} \\ &= P\{Y_1 \leq F_{Y_1}^{-1}(u_1), \dots, Y_d \leq F_{Y_d}^{-1}(u_d)\} \\ &= F_Y\{F_{Y_1}^{-1}(u_1), \dots, F_{Y_d}^{-1}(u_d)\}. \end{aligned} \quad (8.1)$$

Next, letting $u_j = F_{Y_j}(y_j)$, for $j = 1, \dots, d$, in (8.1) we see that

$$F_Y(y_1, \dots, y_d) = C_Y\{F_{Y_1}(y_1), \dots, F_{Y_d}(y_d)\}. \quad (8.2)$$

Equation (8.2) is part of a famous theorem due to Sklar which states that the joint CDF F_Y can be decomposed into the copula C_Y , which contains all information about the dependencies among (Y_1, \dots, Y_d) , and the univariate marginal CDFs F_{Y_1}, \dots, F_{Y_d} , which contain all information about the univariate marginal distributions.

Let

$$c_Y(u_1, \dots, u_d) = \frac{\partial^d}{\partial u_1 \cdots \partial u_d} C_Y(u_1, \dots, u_d) \quad (8.3)$$

be the density associated with C_Y . By differentiating (8.2), we find that the density of \mathbf{Y} is equal to

$$f_Y(y_1, \dots, y_d) = c_Y\{F_{Y_1}(y_1), \dots, F_{Y_d}(y_d)\} f_{Y_1}(y_1) \cdots f_{Y_d}(y_d), \quad (8.4)$$

in which f_{Y_1}, \dots, f_{Y_d} are the univariate marginal densities of Y_1, \dots, Y_d , respectively.

One important property of copulas is that they are invariant to strictly increasing transformations of the component variables. More precisely, suppose that g_j is strictly increasing and $X_j = g_j(Y_j)$ for $j = 1, \dots, d$. Then $\mathbf{X} = (X_1, \dots, X_d)$ and \mathbf{Y} have the same copulas. To see this, first note that the CDF of \mathbf{X} is

$$\begin{aligned} F_X(x_1, \dots, x_d) &= P\{g_1(Y_1) \leq x_1, \dots, g_d(Y_d) \leq x_d\} \\ &= P\{Y_1 \leq g_1^{-1}(x_1), \dots, Y_d \leq g_d^{-1}(x_d)\} \\ &= F_Y\{g_1^{-1}(x_1), \dots, g_d^{-1}(x_d)\}, \end{aligned} \quad (8.5)$$

and therefore the CDF of X_j is

$$F_{X_j}(x_j) = F_{Y_j}\{g_j^{-1}(x_j)\}.$$

Consequently,

$$F_{X_j}^{-1}(u_j) = g_j \left\{ F_{Y_j}^{-1}(u_j) \right\}$$

and

$$g_j^{-1} \left\{ F_{X_j}^{-1}(u_j) \right\} = F_{Y_j}^{-1}(u_j), \quad (8.6)$$

and by applying (8.1) to \mathbf{X} , followed by (8.5), (8.6), and then applying (8.1) to \mathbf{Y} , we conclude that the copula of \mathbf{X} is

$$\begin{aligned} C_X(u_1, \dots, u_d) &= F_X \left\{ F_{X_1}^{-1}(u_1), \dots, F_{X_d}^{-1}(u_d) \right\} \\ &= F_Y \left[g_1^{-1} \left\{ F_{X_1}^{-1}(u_1) \right\}, \dots, g_d^{-1} \left\{ F_{X_d}^{-1}(u_d) \right\} \right] \\ &= F_Y \left\{ F_{Y_1}^{-1}(u_1), \dots, F_{Y_d}^{-1}(u_d) \right\} \\ &= C_Y(u_1, \dots, u_d). \end{aligned}$$

8.2 Special Copulas

All d -dimensional copula functions C have domain $[0, 1]^d$ and range $[0, 1]$. There are three copulas of special interest because they represent independence and two extremes of dependence.

The d -dimensional *independence copula* C_0 is the CDF of d mutually independent Uniform(0,1) random variables. It equals

$$C_0(u_1, \dots, u_d) = u_1 \cdots u_d, \quad (8.7)$$

and the associated density is uniform on $[0, 1]^d$; that is, $c_0(u_1, \dots, u_d) = 1$ on $[0, 1]^d$, and zero elsewhere.

The d -dimensional *co-monotonicity copula* C_+ characterizes perfect positive dependence. Let U be Uniform(0,1). Then, the co-monotonicity copula is the CDF of $\mathbf{U} = (U, \dots, U)$; that is, \mathbf{U} contains d copies of U so that all of the components of \mathbf{U} are equal. Thus,

$$\begin{aligned} C_+(u_1, \dots, u_d) &= P(U \leq u_1, \dots, U \leq u_d) \\ &= P\{U \leq \min(u_1, \dots, u_d)\} = \min(u_1, \dots, u_d). \end{aligned}$$

The co-monotonicity copula is also an upper bound for all copula functions: $C(u_1, \dots, u_d) \leq C_+(u_1, \dots, u_d)$, for all $(u_1, \dots, u_d) \in [0, 1]^d$.

The two-dimensional *counter-monotonicity copula* C_- is defined as the CDF of $(U, 1 - U)$, which has perfect negative dependence. Therefore,

$$\begin{aligned} C_-(u_1, u_2) &= P(U \leq u_1, 1 - U \leq u_2) \\ &= P(1 - u_2 \leq U \leq u_1) = \max(u_1 + u_2 - 1, 0). \end{aligned} \quad (8.8)$$

It is easy to derive the last equality in (8.8). If $1 - u_2 > u_1$, then the event $\{1 - u_2 \leq U \leq u_1\}$ is impossible, so the probability is 0. Otherwise, the

probability is the length of the interval $(1 - u_2, u_1)$, which is $u_1 + u_2 - 1$. All two-dimensional copula functions are bounded below by (8.8). It is not possible to have a counter-monotonicity copula with $d > 2$. If, for example, U_1 is counter-monotonic to U_2 and U_2 is counter-monotonic to U_3 , then U_1 and U_3 will be co-monotonic, not counter-monotonic. However, a lower bound for all copula functions is: $\max(u_1 + \dots + u_d + 1 - d, 0) \leq C(u_1, \dots, u_d)$, for all $(u_1, \dots, u_d) \in [0, 1]^d$. This lower bound is obtainable only point-wise, but it is not itself a copula function for $d > 2$.

To use copulas to model multivariate dependencies, we next consider parametric families of copulas.

8.3 Gaussian and t -Copulas

Multivariate normal and multivariate t -distributions offer a convenient way to generate families of copulas. Let $\mathbf{Y} = (Y_1, \dots, Y_d)$ have a multivariate normal distribution. Since C_Y depends only on the dependencies within \mathbf{Y} , not the univariate marginal distributions, C_Y depends only on the $d \times d$ correlation matrix of \mathbf{Y} , which will be denoted by Ω . Therefore, there is a one-to-one correspondence between correlation matrices and Gaussian copulas. The Gaussian copula¹ with correlation matrix Ω will be denoted $C_{\text{Gauss}}(u_1, \dots, u_d | \Omega)$.

If a random vector \mathbf{Y} has a Gaussian copula, then \mathbf{Y} is said to have a *meta-Gaussian distribution*. This does not, of course, mean that \mathbf{Y} has a multivariate Gaussian distribution, since the univariate marginal distributions of \mathbf{Y} could be any distributions at all. A d -dimensional Gaussian copula whose correlation matrix is the identity matrix, so that all correlations are zero, is the d -dimensional independence copula. A Gaussian copula will converge to the co-monotonicity copula C_+ if all correlations in Ω converge to 1. In the bivariate case, as the pair-wise correlation converges to -1 , the copula converges to the counter-monotonicity copula C_- .

Similarly, let $C_t(u_1, \dots, u_d | \Omega, \nu)$ denote the copula of a random vector that has a multivariate t -distribution with tail index² ν and correlation matrix Ω .³ For multivariate t random vectors the tail index ν affects both the univariate marginal distributions and the tail dependence between components, so ν is a parameter of the t -copula C_t . We will see in Sect. 8.6 that ν similarly determines the amount of tail dependence of random vectors that have a t -copula. Such a vector is said to have a *meta-t-distribution*.

¹ Gaussian and normal distributions are synonymous and the Gaussian copula may also be referred to as the normal copula, especially in R functions.

² The tail index parameter for the t -distribution is also commonly referred to as the degrees-of-freedom parameter by its association with the theory of linear regression, and some R functions use the abbreviations `df` or `nu`.

³ There is a minor technical issue here if $\nu \leq 2$. In this case, the t -distribution does not have covariance and correlation matrices. However, it still has a scale matrix and we will assume that the scale matrix is equal to some correlation matrix Ω .

8.4 Archimedean Copulas

An *Archimedean copula* with a strict generator has the form

$$C(u_1, \dots, u_d) = \varphi^{-1}\{\varphi(u_1) + \dots + \varphi(u_d)\}, \quad (8.9)$$

where the generator function φ satisfies the following conditions

1. φ is a continuous, strictly decreasing, and convex function mapping $[0, 1]$ onto $[0, \infty]$,
2. $\varphi(0) = \infty$, and
3. $\varphi(1) = 0$.

A plot of a generator function is shown in Fig. 8.1 to illustrate these properties. It was generated using the `iPsi()` function from R's `copula` package with the following commands.

```

1 library(copula)
2 u = seq(0.000001, 1, length=500)
3 frank = iPsi(copula=archmCopula(family="frank", param=1), u)
4 plot(u, frank, type="l", lwd=3, ylab=expression(phi(u)))
5 abline(h=0) ; abline(v=0)

```

It is possible to relax assumption 2, but then the generator is not called strict and construction of the copula is more complex. The generator function φ is not unique; for example, $a\varphi$, in which a is any positive constant, generates the same copula as φ . The independence copula C_0 is an Archimedean copula with generator function $\varphi(u) = -\log(u)$. There are many families of Archimedean copulas, but we will only look at four, the Frank, Clayton, Gumbel, and Joe copulas.

Notice that in (8.9), the value of $C(u_1, \dots, u_d)$ is unchanged if we permute u_1, \dots, u_d . A distribution with this property is called *exchangeable*. One consequence of exchangeability is that both Kendall's and Spearman's rank correlation introduced later in Sect. 8.5 are the same for all pairs of variables. Archimedean copulas are most useful in the bivariate case or in applications where we expect all pairs to have similar dependencies.

8.4.1 Frank Copula

The Frank copula has generator

$$\varphi_{\text{Fr}}(u|\theta) = -\log\left(\frac{e^{-\theta u} - 1}{e^{-\theta} - 1}\right), \quad -\infty < \theta < \infty.$$

The inverse generator is

$$\varphi_{\text{Fr}}^{-1}(y|\theta) = -\frac{1}{\theta} \log\{e^{-y}(e^{-\theta} - 1) + 1\}.$$

Therefore, by (8.9), the bivariate Frank copula is

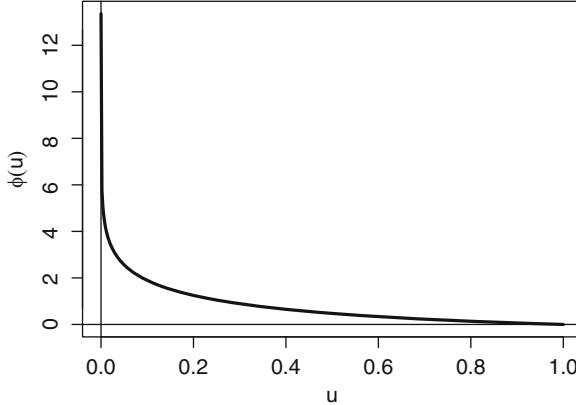


Fig. 8.1. Generator function for the Frank copula with $\theta = 1$.

$$C_{\text{Fr}}(u_1, u_2 | \theta) = -\frac{1}{\theta} \log \left\{ 1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right\}. \quad (8.10)$$

The case $\theta = 0$ requires some care, since plugging this value into (8.10) gives 0/0. Instead, one must evaluate the limit of (8.10) as $\theta \rightarrow 0$. Using the approximations $e^x - 1 \approx x$ and $\log(1+x) \approx x$ as $x \rightarrow 0$, one can show that as $\theta \rightarrow 0$, $C_{\text{Fr}}(u_1, u_2 | \theta) \rightarrow u_1 u_2$, the bivariate independence copula C_0 . Therefore, for $\theta = 0$ we define the Frank copula to be the independence copula.

It is interesting to study the limits of $C_{\text{Fr}}(u_1, u_2 | \theta)$ as $\theta \rightarrow \pm\infty$. As $\theta \rightarrow -\infty$, the bivariate Frank copula converges to the counter-monotonicity copula C_- . To see this, first note that as $\theta \rightarrow -\infty$,

$$C_{\text{Fr}}(u_1, u_2 | \theta) \sim -\frac{1}{\theta} \log \left\{ 1 + e^{-\theta(u_1+u_2-1)} \right\}. \quad (8.11)$$

If $u_1 + u_2 - 1 > 0$, then as $\theta \rightarrow -\infty$, the exponent $-\theta(u_1 + u_2 - 1)$ in (8.11) converges to ∞ and

$$\log \left\{ 1 + e^{-\theta(u_1+u_2-1)} \right\} \sim -\theta(u_1 + u_2 - 1),$$

so that $C_{\text{Fr}}(u_1, u_2 | \theta) \rightarrow u_1 + u_2 - 1$. Similarly, if $u_1 + u_2 - 1 < 0$, then $-\theta(u_1 + u_2 - 1) \rightarrow -\infty$, and $C_{\text{Fr}}(u_1, u_2 | \theta) \rightarrow 0$. Putting these results together, we see that $C_{\text{Fr}}(u_1, u_2 | \theta)$ converges to $\max(0, u_1 + u_2 - 1)$, the counter-monotonicity copula C_- , as $\theta \rightarrow -\infty$.

As $\theta \rightarrow \infty$, $C_{\text{Fr}}(u_1, u_2 | \theta) \rightarrow \min(u_1, u_2)$, the co-monotonicity copula C_+ . Verification of this is left as an exercise for the reader.

Figure 8.2 contains scatterplots of nine bivariate random samples from various Frank copulas, all with a sample size of 200 and with values of θ that give dependencies ranging from strongly negative to strongly positive. Pseudo-random samples may be generated from the copula distributions discussed in this chapter using the `rCopula()` function from R's `copula` package. The convergence to the counter-monotonicity (co-monotonicity) copula as $\theta \rightarrow -\infty (+\infty)$ can be seen in the scatterplots.

```

6 set.seed(5640)
7 theta = c(-100, -50, -10, -1, 0, 5, 20, 50, 500)
8 par(mfrow=c(3,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
9 for(i in 1:9){
10   U = rCopula(n=200,
11                 copula=archmCopula(family="frank", param=theta[i]))
12   plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
13         main=eval(substitute(expression(paste(theta, " = ", j)),
14                           list(j = as.character(theta[i])))))
15 }
```

8.4.2 Clayton Copula

The *Clayton copula*, with generator function $\varphi_{\text{Cl}}(u|\theta) = \frac{1}{\theta}(u^{-\theta} - 1)$, $\theta > 0$, is

$$C_{\text{Cl}}(u_1, \dots, u_d|\theta) = (u_1^{-\theta} + \dots + u_d^{-\theta} + 1 - d)^{-1/\theta}.$$

We define the Clayton copula for $\theta = 0$ as

$$\lim_{\theta \downarrow 0} C_{\text{Cl}}(u_1, \dots, u_d|\theta) = u_1 \cdots u_d$$

which is the independence copula C_0 . There is another way to derive this result. As $\theta \downarrow 0$, l'Hôpital's rule shows that the generator $\frac{1}{\theta}(u^{-\theta} - 1)$ converges to $\varphi_{\text{Cl}}(u|\theta \downarrow 0) = -\log(u)$ with inverse $\varphi_{\text{Cl}}^{-1}(y|\theta \downarrow 0) = \exp(-y)$. Therefore,

$$\begin{aligned} \lim_{\theta \downarrow 0} C_{\text{Cl}}(u_1, \dots, u_d|\theta) &= \varphi_{\text{Cl}}^{-1}\{\varphi_{\text{Cl}}(u_1|\theta \downarrow 0) + \dots + \varphi_{\text{Cl}}(u_d|\theta \downarrow 0)|\theta \downarrow 0\} \\ &= \exp\{-(-\log u_1 - \dots - \log u_d)\} = u_1 \cdots u_d. \end{aligned}$$

It is possible to extend the range of θ to include $-1 \leq \theta < 0$, but then the generator $(u^{-\theta} - 1)/\theta$ is finite at $u = 0$ in violation of assumption 2, of strict generators. Thus, the generator is not strict if $\theta < 0$. As a result, it is necessary to define $C_{\text{Cl}}(u_1, \dots, u_d|\theta)$ to equal 0 for small values of u_i in this case. To appreciate this, consider the bivariate Clayton copula. If $-1 \leq \theta < 0$, then $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$ occurs when u_1 and u_2 are both small. In these cases, $C_{\text{Cl}}(u_1, u_2|\theta)$ is set equal to 0. Therefore, there is no probability in the region $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$. In the limit, as $\theta \rightarrow -1$, there is no probability in the region $u_1 + u_2 < 1$.

As $\theta \rightarrow -1$, the bivariate Clayton copula converges to the counter-monotonicity copula C_- , and as $\theta \rightarrow \infty$, the Clayton copula converges to the co-monotonicity copula C_+ .

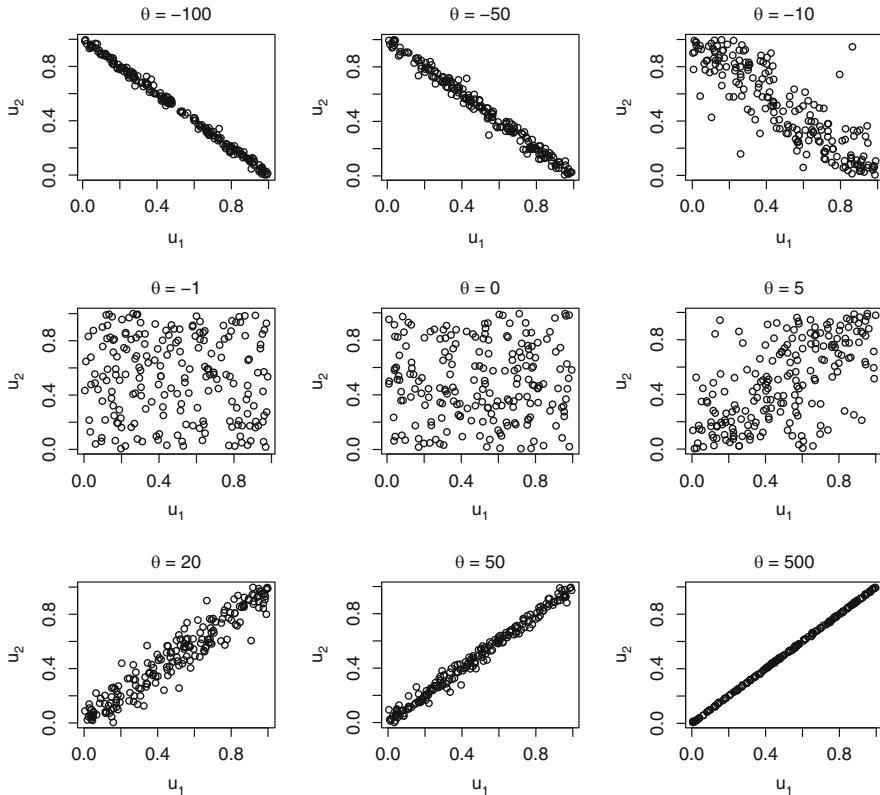


Fig. 8.2. Bivariate random samples of size 200 from various Frank copulas.

Figure 8.3 contains scatterplots of nine bivariate random samples from various Clayton copulas, all with a sample size of 200 and with values of θ that give dependencies ranging from counter-monotonicity to co-monotonicity.

```

16 set.seed(5640)
17 theta = c(-0.98, -0.7, -0.3, -0.1, 0.1, 0.5, 1, 5, 15, 100)
18 par(mfrow=c(3,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
19 for(i in 1:9){
20   U = rCopula(n=200,
21               copula=archmCopula(family="clayton", param=theta[i]))
22   plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
23         main=eval(substitute(expression(paste(theta, " = ", j)),
24         list(j = as.character(theta[i])))))
25 }
```

Comparing Figs. 8.2 and 8.3, we see that the Frank and Clayton copulas are rather different when the amount of dependence is somewhere between these two extremes. In particular, the Clayton copula's exclusion of the region $u_1^{-\theta} + u_2^{-\theta} - 1 < 0$ when $\theta < 0$ is evident, especially in the example with

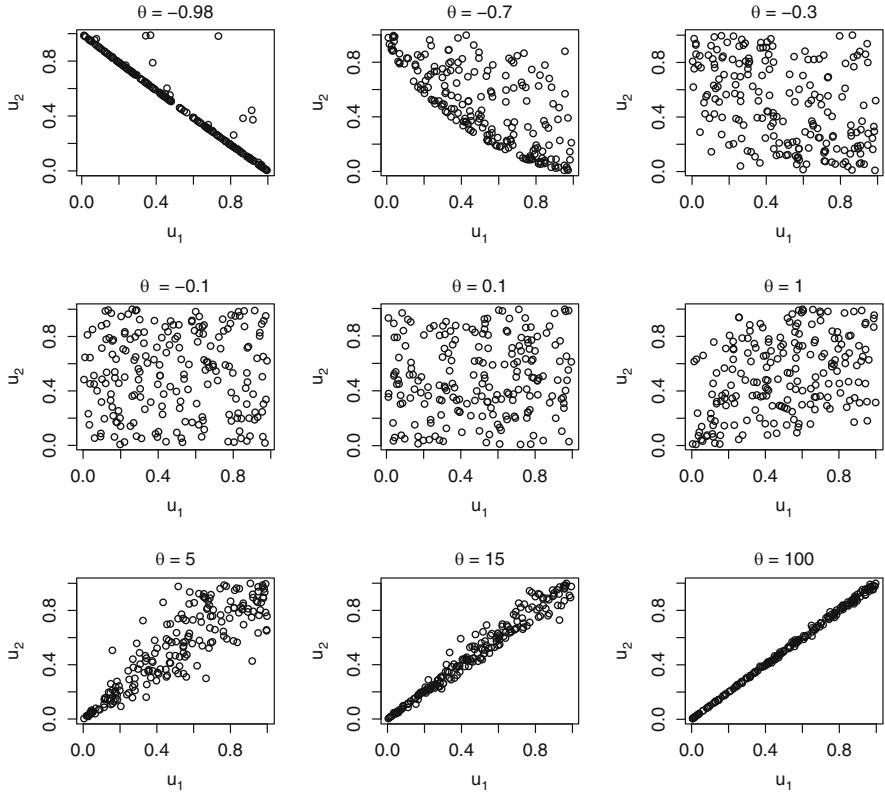


Fig. 8.3. Bivariate random samples of size 200 from various Clayton copulas.

$\theta = -0.7$. In contrast, the Frank copula has positive probability on the entire unit square. The Frank copula is symmetric about the diagonal from $(0, 1)$ to $(1, 0)$, but the Clayton copula does not have this symmetry.

8.4.3 Gumbel Copula

The Gumbel copula has the generator $\varphi_{\text{Gu}}(u|\theta) = (-\log u)^\theta$, $\theta \geq 1$, and consequently is equal to

$$C_{\text{Gu}}(u_1, \dots, u_d|\theta) = \exp \left[-\left\{ (-\log u_1)^\theta + \dots + (-\log u_d)^\theta \right\}^{1/\theta} \right].$$

The Gumbel copula is the independence copula C_0 when $\theta = 1$, and converges to the co-monotonicity copula C_+ as $\theta \rightarrow \infty$, but the Gumbel copula cannot have negative dependence.

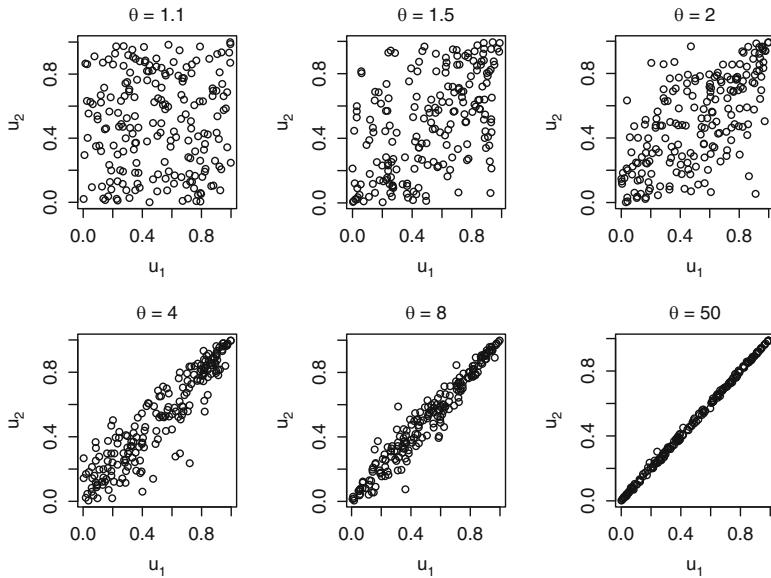


Fig. 8.4. Bivariate random samples of size 200 from various Gumbel copulas.

Figure 8.4 contains scatterplots of six bivariate random samples from various Gumbel copulas, with a sample size of 200 and with values of θ that give dependencies ranging from near independence to strong positive dependence.

```

26 set.seed(5640)
27 theta = c(1.1, 1.5, 2, 4, 8, 50)
28 par(mfrow=c(2,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
29 for(i in 1:6){
30   U = rCopula(n=200,
31               copula=archmCopula(family="gumbel", param=theta[i]))
32   plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
33         main=eval(substitute(expression(paste(theta, " = ", j)),
34         list(j = as.character(theta[i])))))
35 }
```

8.4.4 Joe Copula

The Joe copula is similar to the Gumbel copula; it cannot have negative dependence, but it allows even stronger upper tail dependence and is closer to being the reverse of the Clayton copula in the positive dependence case. The Joe copula has the generator $\varphi_{\text{Joe}}(u|\theta) = -\log\{1 - (1 - u)^\theta\}$, $\theta \geq 1$. In the bivariate case, the Joe copula is equal to

$$C_{\text{Joe}}(u_1, u_2|\theta) = 1 - [(1 - u_1)^\theta + (1 - u_2)^\theta - (1 - u_1)^\theta(1 - u_2)^\theta]^{1/\theta}.$$

The Joe copula is the independence copula C_0 when $\theta = 1$, and converges to the co-monotonicity copula C_+ as $\theta \rightarrow \infty$.

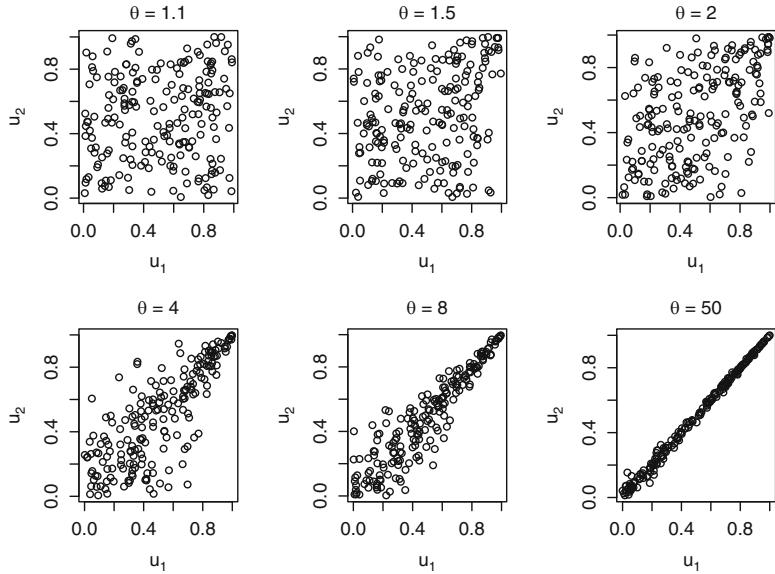


Fig. 8.5. Bivariate random samples of size 200 from various Joe copulas.

Figure 8.5 contains scatterplots of six bivariate random samples from various Joe copulas, with a sample size of 200 and with values of θ that give dependencies ranging from near independence to strong positive dependence.

```

36 set.seed(5640)
37 theta = c(1.1, 1.5, 2, 4, 8, 50)
38 par(mfrow=c(2,3), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
39 for(i in 1:6){
40   U = rCopula(n=200,
41                 copula=archmCopula(family="joe", param=theta[i]))
42   plot(U, xlab=expression(u[1]), ylab=expression(u[2]),
43         main=eval(substitute(expression(paste(theta, " = ", j)),
44         list(j = as.character(theta[i])))))
45 }
```

In applications, it is useful that the different copula families have different properties, since this increases our ability to find a copula that fits the data adequately.

8.5 Rank Correlation

The Pearson correlation coefficient defined by (4.4) is not convenient for fitting copulas to data, since it depends on the univariate marginal distributions as well as the copula. Rank correlation coefficients remedy this problem, since they depend only on the copula.

For each variable, the ranks of that variable are determined by ordering the observations from smallest to largest and giving the smallest rank 1, the next-smallest rank 2, and so forth. In other words, if Y_1, \dots, Y_n is a sample, then the *rank* of Y_i in the sample is equal to 1 if Y_i is the smallest observation, 2 if Y_i is the second smallest, and so forth. More mathematically, the rank of Y_i can also be defined by the formula

$$\text{rank}(Y_i) = \sum_{j=1}^n I(Y_j \leq Y_i), \quad (8.12)$$

which counts the number of observations (including Y_i itself) that are less than or equal to Y_i . A *rank statistic* is a statistic that depends on the data only through the ranks. A key property of ranks is that they are unchanged by strictly monotonic transformations of the variables. In particular, the ranks are unchanged by transforming each variable by its CDF, so the distribution of any rank statistic depends only on the copula of the observations, not on the univariate marginal distributions.

We will be concerned with rank statistics that measure statistical association between pairs of variables. These statistics are called *rank correlations*. There are two rank correlation coefficients in widespread usage, Kendall's tau and Spearman's rho.

8.5.1 Kendall's Tau

Let (Y_1, Y_2) be a bivariate random vector and let (Y_1^*, Y_2^*) be an independent copy of (Y_1, Y_2) . Then (Y_1, Y_2) and (Y_1^*, Y_2^*) are called a *concordant pair* if the ranking of Y_1 relative to Y_1^* is the same as the ranking of Y_2 relative to Y_2^* , that is, either $Y_1 > Y_1^*$ and $Y_2 > Y_2^*$ or $Y_1 < Y_1^*$ and $Y_2 < Y_2^*$. In either case, $(Y_1 - Y_1^*)(Y_2 - Y_2^*) > 0$. Similarly, (Y_1, Y_2) and (Y_1^*, Y_2^*) are called a *discordant pair* if $(Y_1 - Y_1^*)(Y_2 - Y_2^*) < 0$. *Kendall's tau* is the probability of a concordant pair minus the probability of a discordant pair. Therefore, Kendall's tau for (Y_1, Y_2) is

$$\begin{aligned} \rho_\tau(Y_1, Y_2) &= P\{(Y_1 - Y_1^*)(Y_2 - Y_2^*) > 0\} - P\{(Y_1 - Y_1^*)(Y_2 - Y_2^*) < 0\} \\ &= E[\text{sign}\{(Y_1 - Y_1^*)(Y_2 - Y_2^*)\}], \end{aligned} \quad (8.13)$$

where the *sign function* is

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ -1, & x < 0, \\ 0, & x = 0. \end{cases}$$

It is clear from (8.13) that ρ_τ is symmetric in its arguments and takes values in $[-1, 1]$. It is easy to check that if g and h are increasing functions, then

$$\rho_\tau\{g(Y_1), h(Y_2)\} = \rho_\tau(Y_1, Y_2). \quad (8.14)$$

Stated differently, Kendall's tau is invariant to monotonically increasing transformations. If g and h are the marginal CDFs of Y_1 and Y_2 , then the left-hand side of (8.14) is the value of Kendall's tau for a pair of random variables distributed according to the copula of (Y_1, Y_2) . This shows that Kendall's tau depends only on the copula of a bivariate random vector. For a random vector \mathbf{Y} , we define the *Kendall's tau correlation matrix* $\boldsymbol{\Omega}_\tau$ to be the matrix whose (j, k) entry is Kendall's tau for the j th and k th components of \mathbf{Y} , that is $[\boldsymbol{\Omega}_\tau(\mathbf{Y})]_{jk} = \rho_\tau(Y_j, Y_k)$.

If we have a bivariate sample $\mathbf{Y}_{1:n} = \{(Y_{i,1}, Y_{i,2}) : i = 1, \dots, n\}$, then the sample Kendall's tau is

$$\hat{\rho}_\tau(\mathbf{Y}_{1:n}) = \binom{n}{2}^{-1} \sum_{1 \leq i < j \leq n} \text{sign}\{(Y_{i,1} - Y_{j,1})(Y_{i,2} - Y_{j,2})\}. \quad (8.15)$$

Note that $\binom{n}{2}$ is the number of summands in (8.15), so $\hat{\rho}_\tau$ is the average of $\text{sign}\{(Y_{i,1} - Y_{j,1})(Y_{i,2} - Y_{j,2})\}$ across all distinct pairs of observations and is a sample version of (8.13). The sample Kendall's tau correlation matrix is defined analogously to $\boldsymbol{\Omega}_\tau$.

8.5.2 Spearman's Rank Correlation Coefficient

For a sample, Spearman's correlation coefficient is simply the usual Pearson correlation calculated from the marginal ranks of the data. For a distribution (that is, an infinite population rather than a finite sample), both variables are transformed by their univariate marginal CDFs and then the Pearson correlation is computed for the transformed variables. Transforming a random variable by its CDF is analogous to computing the ranks of a variable in a finite sample.

Stated differently, Spearman's rank correlation coefficient, also called *Spearman's rho*, for a bivariate random vector (Y_1, Y_2) will be denoted as $\rho_S(Y_1, Y_2)$ and is defined to be the Pearson correlation coefficient of $\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}$:

$$\rho_S(Y_1, Y_2) = \text{Corr}\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}.$$

Since the joint CDF of $\{F_{Y_1}(Y_1), F_{Y_2}(Y_2)\}$ is the copula of (Y_1, Y_2) , Spearman's rho, like Kendall's tau, depends only on the copula function.

The sample version of Spearman's correlation coefficient can be computed from the ranks of the data and for a bivariate sample $\mathbf{Y}_{1:n} = \{(Y_{i,1}, Y_{i,2}) : i = 1, \dots, n\}$, is

$$\hat{\rho}_S(\mathbf{Y}_{1:n}) = \frac{12}{n(n^2 - 1)} \sum_{i=1}^n \left\{ \text{rank}(Y_{i,1}) - \frac{n+1}{2} \right\} \left\{ \text{rank}(Y_{i,2}) - \frac{n+1}{2} \right\}. \quad (8.16)$$

The set of ranks for any variable is, of course, the integers 1 to n , and hence $(n+1)/2$ is the mean of its ranks. It can be shown that $\hat{\rho}_S(\mathbf{Y}_{1:n})$ is the sample Pearson correlation between the ranks of $\{Y_{i,1}\}$ and the ranks of $\{Y_{i,2}\}$.⁴

If $\mathbf{Y} = (Y_1, \dots, Y_d)$ is a random vector, then the *Spearman's correlation matrix* $\boldsymbol{\Omega}_S$ of \mathbf{Y} is the correlation matrix of $\{F_{Y_1}(Y_1), \dots, F_{Y_d}(Y_d)\}$ and contains the Spearman's correlation coefficients for all pairs of components of \mathbf{Y} , such that $[\boldsymbol{\Omega}_S(\mathbf{Y})]_{jk} = \rho_S(Y_j, Y_k)$, for all $j, k = 1, \dots, d$. The sample Spearman's correlation matrix is defined analogously.

8.6 Tail Dependence

Tail dependence measures association between the extreme values of two random variables and depends only on their copula. We will start with lower tail dependence, which uses extremes in the lower tail. Suppose that $\mathbf{Y} = (Y_1, Y_2)$ is a bivariate random vector with copula C_Y . Then the *coefficient of lower tail dependence* is denoted by λ_ℓ and defined as

$$\lambda_\ell := \lim_{q \downarrow 0} P\{Y_2 \leq F_{Y_2}^{-1}(q) \mid Y_1 \leq F_{Y_1}^{-1}(q)\} \quad (8.17)$$

$$= \lim_{q \downarrow 0} \frac{P\{Y_1 \leq F_{Y_1}^{-1}(q), Y_2 \leq F_{Y_2}^{-1}(q)\}}{P\{Y_1 \leq F_{Y_1}^{-1}(q)\}} \quad (8.18)$$

$$= \lim_{q \downarrow 0} \frac{P\{F_{Y_1}(Y_1) \leq q, F_{Y_2}(Y_2) \leq q\}}{P\{F_{Y_1}(Y_1) \leq q\}} \quad (8.19)$$

$$= \lim_{q \downarrow 0} \frac{C_Y(q, q)}{q}. \quad (8.20)$$

It is helpful to look at these equations individually. As elsewhere in this chapter, for simplicity we are assuming that F_{Y_1} and F_{Y_2} are strictly increasing on their supports and therefore have inverses.

First, (8.17) defines λ_ℓ as the limit as $q \downarrow 0$ of the conditional probability that Y_2 is less than or equal to its q th quantile, given that Y_1 is less than or equal to its q th quantile. Since we are taking a limit as $q \downarrow 0$, we are looking at the extreme left tail. What happens if Y_1 and Y_2 are independent? Then $P(Y_2 \leq y_2 \mid Y_1 \leq y_1) = P(Y_2 \leq y_2)$ for all y_1 and y_2 . Therefore, the conditional probability in (8.17) equals the unconditional probability $P(Y_2 \leq F_{Y_2}^{-1}(q))$ and this probability converges to 0 as $q \downarrow 0$. Therefore, $\lambda_\ell = 0$ implies that in the extreme left tail, Y_1 and Y_2 behave as if they were independent.

Equation (8.18) is just the definition of conditional probability. Equation (8.19) is simply (8.18) after applying the probability transformation to each variable. The numerator in (8.19) is the copula by definition, and the

⁴ If there are ties, then ranks are averaged among tied observations. For example, if there are two observations tied for smallest, then they each get a rank of 1.5. When there are ties, these results must be modified.

denominator in (8.20) is the result of $F_{Y_1}(Y_1)$ being distributed Uniform(0,1); see (A.9).

Deriving formulas for λ_ℓ for Gaussian and t -copulas is a topic best left for more advanced books. Here we give only the results; see Sect. 8.8 for further reading. For any bivariate Gaussian copula C_{Gauss} with $\rho \neq 1$, $\lambda_\ell = 0$, that is, Gaussian copulas do not have tail dependence except in the extreme case of perfect positive correlation. For a bivariate t -copula C_t with tail index ν and correlation ρ ,

$$\lambda_\ell = 2F_{t,\nu+1} \left\{ -\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}} \right\}, \quad (8.21)$$

where $F_{t,\nu+1}$ is the CDF of the t -distribution with tail index $\nu + 1$.

Since $F_{t,\nu+1}(-\infty) = 0$, we see that $\lambda_\ell \rightarrow 0$ as $\nu \rightarrow \infty$, which makes sense since the t -copula converges to a Gaussian copula as $\nu \rightarrow \infty$. Also, $\lambda_\ell \rightarrow 0$ as $\rho \rightarrow -1$, which is also not too surprising, since $\rho = -1$ is perfect *negative* dependence and λ_ℓ measures *positive* tail dependence.

The coefficient of upper tail dependence λ_u is

$$\lambda_u := \lim_{q \uparrow 1} P\{Y_2 \geq F_{Y_2}^{-1}(q) \mid Y_1 \geq F_{Y_1}^{-1}(q)\} \quad (8.22)$$

$$= 2 - \lim_{q \uparrow 1} \frac{1 - C_Y(q, q)}{1 - q}. \quad (8.23)$$

We see that λ_u is defined analogously to λ_ℓ ; λ_u is the limit as $q \uparrow 1$ of the conditional probability that Y_2 is greater than or equal to its q th quantile, given that Y_1 is greater than or equal to its q th quantile. Deriving (8.23) is left as an exercise for the interested reader.

For Gaussian and t -copula, $\lambda_u = \lambda_\ell$, so that $\lambda_u = 0$ for any Gaussian copula and for a t -copula, λ_ℓ is given by the right-hand side of (8.21). Coefficients of tail dependence for t -copulas are plotted in Fig. 8.6. One can see $\lambda_\ell = \lambda_u$ depends strongly on both ρ and ν . For the independence copula C_0 , λ_ℓ and λ_u are both equal to 0, and for the co-monotonicity copula C_+ , both are equal to 1.

```

46 rho = seq(-1,1, by=0.01)
47 df = c(1, 4, 25, 240)
48 x1 = -sqrt((df[1]+1)*(1-rho)/(1+rho))
49 lambda1 = 2*pt(x1,df[1]+1)
50 x4 = -sqrt((df[2]+1)*(1-rho)/(1+rho))
51 lambda4 = 2*pt(x4,df[2]+1)
52 x25 = -sqrt((df[3]+1)*(1-rho)/(1+rho))
53 lambda25 = 2*pt(x25,df[3]+1)
54 x250 = -sqrt((df[4]+1)*(1-rho)/(1+rho))
55 lambda250 = 2*pt(x250,df[4]+1)
56 par(mfrow=c(1,1), lwd=2, cex.axis=1.2, cex.lab=1.2)
57 plot(rho, lambda1, type="l", lty=1, xlab=expression(rho),

```

```

58     ylab=expression(lambda[1]==lambda[u]))
59  lines(rho, lambda4, lty=2)
60  lines(rho, lambda25, lty=3)
61  lines(rho, lambda250, lty=4)
62 legend("topleft", c(expression(nu==1), expression(nu==4),
63                     expression(nu==25), expression(nu==250)), lty=1:4)

```

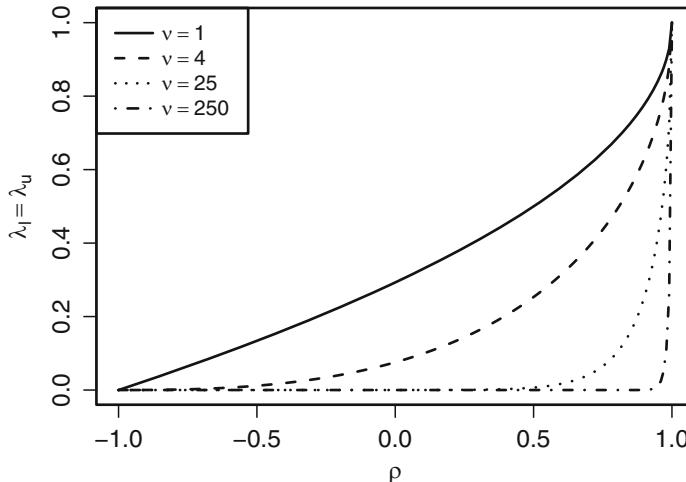


Fig. 8.6. Coefficients of tail dependence for bivariate t -copulas as functions of ρ for $\nu = 1, 4, 25$, and 250 .

Knowing whether or not there is tail dependence is important for risk management. If there are no tail dependencies among the returns on the assets in a portfolio, then there is little risk of simultaneous very negative returns, and the risk of an extreme negative return on the portfolio is low. Conversely, if there are tail dependencies, then the likelihood of extreme negative returns occurring simultaneously on several assets in the portfolio can be high. As such, tail dependencies should be considered when assessing the diversification and risk of any portfolio.

8.7 Calibrating Copulas

Assume that we have an i.i.d. sample $\mathbf{Y}_{1:n} = \{(Y_{i,1}, \dots, Y_{i,d}) : i = 1, \dots, n\}$, and we wish to estimate the copula of \mathbf{Y} and perhaps its univariate marginal distributions as well.

An important task is choosing a copula model. The various copula models differ notably from each other. For example, some have tail dependence

and others do not. The Gumbel copula and Joe copula allow only positive dependence or independence. The Clayton copula with negative dependence excludes the region where both u_1 and u_2 are small. As will be seen in this section, an appropriate copula model can be selected via AIC, and by using graphical techniques.

8.7.1 Maximum Likelihood

Suppose we have parametric models $F_{Y_1}(\cdot | \boldsymbol{\theta}_1), \dots, F_{Y_d}(\cdot | \boldsymbol{\theta}_d)$ for the marginal CDFs as well as a parametric model $c_Y(\cdot | \boldsymbol{\theta}_C)$ for the copula density. By taking logs of (8.4), we find that the log-likelihood is

$$\begin{aligned} \log\{L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\} &= \sum_{i=1}^n \left(\log[c_Y\{F_{Y_1}(Y_{i,1} | \boldsymbol{\theta}_1), \dots, F_{Y_d}(Y_{i,d} | \boldsymbol{\theta}_d)\} | \boldsymbol{\theta}_C] \right. \\ &\quad \left. + \log\{f_{Y_1}(Y_{i,1} | \boldsymbol{\theta}_1)\} + \dots + \log\{f_{Y_d}(Y_{i,d} | \boldsymbol{\theta}_d)\} \right). \end{aligned} \quad (8.24)$$

Maximum likelihood estimation finds the maximum of $\log\{L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\}$ over the entire set of parameters $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)$.

There are two potential problems with maximum likelihood estimation. First, because of the large number of parameters, especially for large values of d , numerically maximizing $\log\{L(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_d, \boldsymbol{\theta}_C)\}$ can be challenging. This difficulty can be ameliorated by the use of starting values that are close to the MLEs. The pseudo-maximum likelihood estimates discussed in the next section are easier to compute than the MLE and can be used either as an alternative to the MLE or as starting values for the MLE.

Second, maximum likelihood estimation requires parametric models for both the copula and the univariate marginal distributions. If any of the univariate marginal distributions are not well fit by a convenient parametric family, this may cause biases in the estimated parameters of both the univariate marginal distributions and the copula. The semiparametric approach to pseudo-maximum likelihood estimation, where the univariate marginal distributions are estimated nonparametrically, provides a remedy to this problem.

8.7.2 Pseudo-Maximum Likelihood

Pseudo-maximum likelihood estimation is a two-step procedure. In the first step, each of the d univariate marginal distribution functions is estimated, one at a time. Let \widehat{F}_{Y_j} be the estimate of the j th univariate marginal CDF, $j = 1, \dots, d$. In the second step,

$$\sum_{i=1}^n \log \left[c_Y \left\{ \widehat{F}_{Y_1}(Y_{i,1}), \dots, \widehat{F}_{Y_d}(Y_{i,d}) | \boldsymbol{\theta}_C \right\} \right] \quad (8.25)$$

is maximized over $\boldsymbol{\theta}_C$. Note that (8.25) is obtained from (8.24) by deleting terms that do not depend on $\boldsymbol{\theta}_C$ and replacing the univariate marginal CDFs

by estimates. By estimating parameters in the univariate marginal distributions and in the copula separately, the pseudo-maximum likelihood approach avoids a high-dimensional optimization.

There are two approaches to the first step, parametric and nonparametric. In the parametric approach, parametric models $F_{Y_1}(\cdot | \boldsymbol{\theta}_1), \dots, F_{Y_d}(\cdot | \boldsymbol{\theta}_d)$ for the univariate marginal CDFs are assumed as in maximum likelihood estimation. The data $Y_{1,j}, \dots, Y_{n,j}$ for the j th variate are used to estimate $\boldsymbol{\theta}_j$, usually by maximum likelihood as discussed in Chap. 5. Then, $\widehat{F}_{Y_j}(\cdot) = F_{Y_j}(\cdot | \boldsymbol{\theta}_j)$. In the nonparametric approach, \widehat{F}_{Y_j} is estimated by the empirical CDF of $Y_{1,j}, \dots, Y_{n,j}$, except that the divisor n in (4.2) is replaced by $n + 1$ so that

$$\widehat{F}_{Y_j}(y) = \frac{\sum_{i=1}^n I\{Y_{i,j} \leq y\}}{n+1}. \quad (8.26)$$

With this modified divisor, the maximum value of $\widehat{F}_{Y_j}(Y_{i,j})$ is $n/(n+1)$ rather than 1. Avoiding a value of 1 is essential when, as is often the case, $c_Y(u_1, \dots, u_d | \boldsymbol{\theta}_C) = \infty$ if some of u_1, \dots, u_d are equal to 1.

When both steps are parametric, the estimation method is called *parametric pseudo-maximum likelihood*. The combination of a nonparametric first step and a parametric second step is called *semiparametric pseudo-maximum likelihood*.

In the second step of pseudo-maximum likelihood, the maximization can be difficult when $\boldsymbol{\theta}_C$ is high-dimensional. For example, if one uses a Gaussian or t -copula, then there are $d(d-1)/2$ correlation parameters. One way to solve this problem is to assume some structure among the correlations. An extreme case of this is the *equi-correlation model* where all non-diagonal elements of the correlation matrix have a common value, call it ρ . If one is reluctant to assume some type of structured correlation matrix, then it is essential to have good starting values for the correlation matrix when maximizing (8.25). For Gaussian and t -copulas, starting values can be obtained via rank correlations as discussed in the next section.

The values $\widehat{F}_{Y_j}(Y_{i,j})$, $i = 1, \dots, n$ and $j = 1, \dots, d$, will be called the *uniform-transformed variables*, since they should be distributed approximately Uniform(0,1). The multivariate empirical CDF [see Eq. (A.38)] of the uniform-transformed variables is called the *empirical copula* and is a nonparametric estimate of the copula function. The empirical copula is useful for checking the goodness of fit of parametric copula models; see Example 8.1.

8.7.3 Calibrating Meta-Gaussian and Meta- t -Distributions

Gaussian Copulas

Rank correlation can be useful for estimating the parameters of a copula. Suppose $\mathbf{Y}_{1:n} = \{(Y_{i,1}, \dots, Y_{i,d}) : i = 1, \dots, n\}$, is an i.i.d. sample from a meta-Gaussian distribution. Then its copula is $C_{\text{Gauss}}(\cdot | \boldsymbol{\Omega})$ for some correlation matrix $\boldsymbol{\Omega}$. To estimate the distribution of \mathbf{Y} , we need to estimate the

univariate marginal distributions and Ω . The marginal distribution can be estimated by the methods discussed in Chap. 5. Result (8.28) in the following theorem shows that Ω can be estimated by the sample Spearman's correlation matrix.

Result 8.1 Let $\mathbf{Y} = (Y_1, \dots, Y_d)$ have a meta-Gaussian distribution with continuous univariate marginal distributions and copula $C_{\text{Gauss}}(\cdot | \Omega)$, and let $\Omega_{ij} = [\Omega]_{ij}$. Then

$$\rho_\tau(Y_i, Y_j) = \frac{2}{\pi} \arcsin(\Omega_{ij}), \text{ and} \quad (8.27)$$

$$\rho_S(Y_i, Y_j) = \frac{6}{\pi} \arcsin(\Omega_{ij}/2) \approx \Omega_{ij}. \quad (8.28)$$

Suppose, instead, that \mathbf{Y} has a meta-t-distribution with continuous univariate marginal distributions and copula $C_t(\cdot | \Omega, \nu)$. Then (8.27) still holds, but (8.28) does not hold.

The approximation in (8.28) uses the result that

$$\frac{6}{\pi} \arcsin(x/2) \approx x \text{ for } |x| \leq 1. \quad (8.29)$$

The left- and right-hand sides of (8.29) are equal when $x = -1, 0, 1$, and their maximum difference over the range $-1 \leq x \leq 1$ is 0.018. However, the relative error $\left\{ \frac{6}{\pi} \arcsin(x/2) - x \right\} / \frac{6}{\pi} \arcsin(x/2)$ can be larger, as much as 0.047, and is largest near $x = 0$.

By (8.28), the sample Spearman's rank correlation matrix $\widehat{\Omega}(\mathbf{Y}_{1:n})$ can be used as an estimate of the correlation matrix Ω associated with $C_{\text{Gauss}}(\cdot | \Omega)$. This estimate could be the final one or could be used as a starting value for numeric maximum likelihood or pseudo-maximum likelihood estimation.

t-Copulas

If $\mathbf{Y}_{1:n} = \{(Y_{i,1}, \dots, Y_{i,d}) : i = 1, \dots, n\}$ is a sample from a distribution with a t-copula $C_t(\cdot | \Omega, \nu)$ then we can use (8.27) and the sample Kendall's tau correlations to estimate Ω . Let $\widehat{\Omega}_{\tau,jk}$ be the sample Kendall's tau correlation of $\{Y_{1,j}, \dots, Y_{n,j}\}$ and $\{Y_{1,k}, \dots, Y_{n,k}\}$, the j th and k th components, and let $\widetilde{\Omega}^{**}$ be defined such that $[\widetilde{\Omega}^{**}]_{jk} = \sin\{\frac{\pi}{2} \widehat{\Omega}_{\tau,jk}\}$. Then $\widetilde{\Omega}^{**}$ will have two of the three properties of a correlation matrix; it will be symmetric, with all diagonal entries equal to 1. However, it may not be positive definite, or even semidefinite, because some of its eigenvalues may be negative.

If all of the eigenvalues of $\widetilde{\Omega}^{**}$ are positive, then we will use $\widetilde{\Omega}^{**}$ to estimate Ω . Otherwise, we alter $\widetilde{\Omega}^{**}$ slightly to make it positive definite. By (A.50),

$$\tilde{\boldsymbol{\Omega}}^{**} = \mathbf{O} \operatorname{diag}(\lambda_i) \mathbf{O}^T,$$

where \mathbf{O} is an orthogonal matrix whose columns are the eigenvectors of $\tilde{\boldsymbol{\Omega}}^{**}$ and $\lambda_1, \dots, \lambda_d$ are the corresponding eigenvalues. We then define

$$\tilde{\boldsymbol{\Omega}}^* = \mathbf{O} \operatorname{diag}\{\max(\epsilon, \lambda_i)\} \mathbf{O}^T,$$

where ϵ is some small positive quantity, for example, $\epsilon = 0.001$. Now, $\tilde{\boldsymbol{\Omega}}^*$ is symmetric and positive definite, but its diagonal elements, $\tilde{\Omega}_{ii}^*$, $i = 1, \dots, d$, may not be equal to 1. This problem is easily fixed; multiply the i th row and the i th column of $\tilde{\boldsymbol{\Omega}}^*$ by $(\tilde{\Omega}_{ii}^*)^{-1/2}$, for $i = 1, \dots, d$. The final result, which we denote as $\tilde{\boldsymbol{\Omega}}$, is a bona fide correlation matrix; that is, it is symmetric, positive definite, and it has all diagonal entries equal to 1.

After $\boldsymbol{\Omega}$ has been estimated by $\tilde{\boldsymbol{\Omega}}$, an estimate of the tail index ν is still needed. One can be obtained by plugging $\tilde{\boldsymbol{\Omega}}$ into the log-likelihood (8.25) and then maximizing over ν .

Example 8.1. Flows in pipelines

In this example, we will continue the analysis of the pipeline flows data introduced in Example 4.2. Only the flows in the first two pipelines will be used.

In a fully parametric pseudo-likelihood analysis, the univariate skewed t -model will be used for flows 1 and 2. Let $\hat{U}_{1,j}, \dots, \hat{U}_{n,j}$ be the flows in pipeline j , $j = 1, 2$, transformed by their estimated skewed- t CDFs. We will call the $\hat{U}_{i,j}$ “uniform-transformed flows.” Define $\hat{Z}_{i,j} = \Phi^{-1}(\hat{U}_{i,j})$, where Φ^{-1} is the standard normal quantile function. The $\hat{Z}_{i,j}$ should each be approximately $N(0, 1)$ -distributed, and we will call them “normal-transformed flows.”

```

64 library(copula)
65 library(sn)
66 dat = read.csv("FlowData.csv")
67 dat = dat/10000
68 n = nrow(dat)
69 x1 = dat$Flow1
70 fit1 = st.mple(matrix(1,n,1), y=x1, dp=c(mean(x1), sd(x1), 0, 10))
71 est1 = fit1$dp
72 u1 = pst(x1, dp=est1)
73 x2 = dat$Flow2
74 fit2 = st.mple(matrix(1,n,1), y=x2, dp=c(mean(x2), sd(x2), 0, 10))
75 est2 = fit2$dp
76 u2 = pst(x2, dp=est2)
77 U.hat = cbind(u1, u2)
78 z1 = qnorm(u1)
79 z2 = qnorm(u2)
80 Z.hat = cbind(z1, z2)

```

Both sets of uniform-transformed flows should be approximately Uniform(0,1). Figure 8.7 shows density histograms of both samples of uniform-transformed flows as well as their scatterplot and two-dimensional KDE density contours. The histograms show some deviations from uniform distributions, which suggests that the skewed- t model may not provide adequate fits and that a semiparametric pseudo-maximum likelihood approach might be tried—this is considered below. However, the deviations may be due to random variation.

```

81 library(ks)
82 fhatU = kde(x=U.hat, H=Hscv(x=U.hat))
83 par(mfrow=c(2,2), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
84 hist(u1, main=(a), xlab=expression(hat(U)[1]), freq = FALSE)
85 hist(u2, main=(b), xlab=expression(hat(U)[2]), freq = FALSE)
86 plot(u1, u2, main=(c), xlab = expression(hat(U)[1]),
87       ylab = expression(hat(U)[2]), mgp = c(2.5, 1, 0))
88 plot(fhatU, drawpoints=FALSE, drawlabels=FALSE,
89       cont=seq(10, 80, 10), main=(d), xlab=expression(hat(U)[1]),
90       ylab=expression(hat(U)[2]), mgp = c(2.5, 1, 0))

```

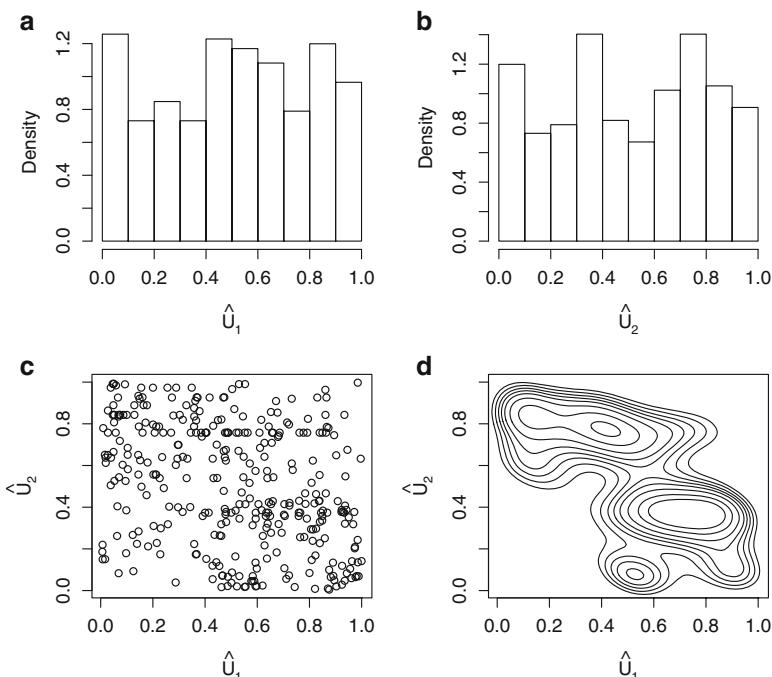


Fig. 8.7. Pipeline data. Density histograms (a) and (b) and a scatterplot (c) of the uniform-transformed flows. The empirical copula \hat{C} is the empirical CDF of the data in (c). Contours (d) from an estimated copula density \hat{c} via a two-dimensional KDE of (c).

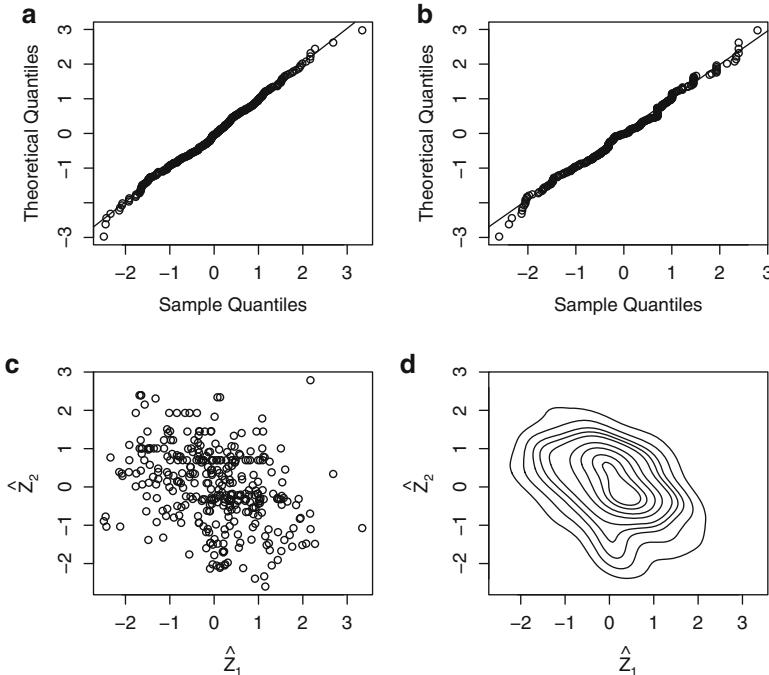


Fig. 8.8. Pipeline data. Normal quantile plots (a) and (b), a scatterplot (c) and KDE density contours for the normal-transformed flows.

The scatterplot in Fig. 8.7 shows some negative association as the data are somewhat concentrated along the diagonal from top left to bottom right. Thus, the Gumbel copula and Joe copula, which cannot have negative dependence, are not appropriate. Also, the Clayton copula may not fit well either, since the scatterplot shows data in the region where both \hat{U}_1 and \hat{U}_2 have small values, but this region is excluded by a Clayton copula with negative dependence. We will soon see that AIC agrees with these conclusions from a graphical analysis, since the Clayton model has higher (worse) AIC values compared to the Gaussian, *t*, and Frank copula models.

Figure 8.8 shows that the normal-transformed flows have approximately linear normal quantile plots, which would be expected if the estimated univariate marginal CDFs were adequate fits. Their scatterplot and KDE density contours again show negative association.

```

91 fhatZ = kde(x=Z.hat, H=Hscv(x=Z.hat))
92 par(mfrow=c(2,2), cex.axis=1.2, cex.lab=1.2, cex.main=1.2)
93 qqnorm(z1, data=T, main="(a)" ; qqline(z1)
94 qqnorm(z2, data=T, main="(b)" ; qqline(z2)
95 plot(z1, z2, main="(c)", xlab = expression(hat(Z)[1]),
96       ylab = expression(hat(Z)[2]), mgp = c(2.5, 1, 0))
97 plot(fhatZ, drawpoints=FALSE, drawlabels=FALSE,

```

```

98     cont=seq(10, 90, 10), main="(d)", xlab=expression(hat(Z)[1]),
99     ylab=expression(hat(Z)[2]), mgp = c(2.5, 1, 0))

```

We will assume for now that the two flows have a meta-Gaussian distribution. There are three ways to estimate the correlation in their Gaussian copula. The first, Spearman's rank correlation, is estimated -0.357 . The second, which uses (8.27) is $\sin(\hat{\rho}_\tau \pi/2)$, where $\hat{\rho}_\tau$ is the sample Kendall's tau rank correlation; its value is -0.371 . The third, Pearson correlation of the normal-transformed flows, is -0.335 . There is reasonably close agreement among the three values, especially relative to their uncertainties; for example, the approximate 95 % confidence interval for the Pearson correlation of the normal-transformed flows is $(-0.426, -0.238)$, and the other two estimate are well within this interval.

```

100 cor.test(u1, u2, method="spearman")
101 cor.test(u1, u2, method="kendall")
102 sin(-0.242*pi/2)
103 cor.test(u1, u2, method="pearson")
104 cor.test(z1, z2, method="pearson")

```

Pearson's product-moment correlation

```

data: z1 and z2
t = -6.56, df = 340, p-value = 2.003e-10
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.426 -0.238
sample estimates:
cor
-0.335

```

Four parametric copulas were fit to the uniform-transformed flows: t , Gaussian, Frank and Clayton. Estimation of the copula distributions discussed in this chapter may be performed using the `fitCopula()` function from R's `copula` package. The Gumbel and Joe copulas are not considered since they only allow positive dependence and these data show negative dependence; attempting to fit these models results in numerical failures. Since we used parametric estimates to transform the flows, we are fitting the copulas by parametric pseudo-maximum likelihood.

```

105 omega = -0.371
106 Ct = fitCopula(copula=tCopula(dim = 2), data=U.hat,
107                 method="ml", start=c(omega, 10))
108 Ct@estimate
109 loglikCopula(param=Ct@estimate, x=U.hat, copula=tCopula(dim = 2))
110 -2*.Last.value + 2*length(Ct@estimate)
111 #
112 Cgauss = fitCopula(copula=normalCopula(dim = 2), data=U.hat,
113                      method="ml", start=c(omega))
114 Cgauss@estimate

```

```

115 loglikCopula(param=Cgauss@estimate, x=U.hat,
116   copula=normalCopula(dim = 2))
117 -2*.Last.value + 2*length(Cgauss@estimate)
118 #
119 Cfr = fitCopula(copula=frankCopula(1, dim=2), data=U.hat,
120   method="ml")
121 Cfr@estimate
122 loglikCopula(param=Cfr@estimate, x=U.hat,
123   copula=frankCopula(dim = 2))
124 -2*.Last.value + 2*length(Cfr@estimate)
125 #
126 Ccl = fitCopula(copula=claytonCopula(1, dim=2), data=U.hat,
127   method="ml")
128 Ccl@estimate
129 loglikCopula(param=Ccl@estimate, x=U.hat,
130   copula=claytonCopula(dim = 2))
131 -2*.Last.value + 2*length(Ccl@estimate)

```

The results are summarized in Table 8.1. Looking at the maximized log-likelihood values, we see that the Frank copula fits best since it minimizes AIC, but the t and Gaussian fit reasonably well. Figure 8.9 shows the uniform-transformed flows scatterplot and contours of the distribution functions of five copulas: the independence copula and the four estimated parametric copulas; the empirical copula contours have been overlaid for comparison. The t -copula is similar to the Gaussian since $\hat{\nu} = 22.247$ is large. The Frank copula fits best in the sense that its contours are closest to those of the empirical copula. This is in agreement with the AIC values.

Table 8.1. Estimates of copula parameters, maximized log-likelihood, and AIC using the uniform-transformed pipeline flow data.

Copula family	Estimates	Maximized log-likelihood	AIC
t	$\hat{\rho} = -0.340$ $\hat{\nu} = 22.247$	20.98	-37.96
Gaussian	$\hat{\rho} = -0.331$	20.36	-38.71
Frank	$\hat{\theta} = -2.249$	23.07	-44.13
Clayton	$\hat{\theta} = -0.166$	9.86	-17.72

```

132 par(mfrow=c(2,3), mgp = c(2.5, 1, 0))
133 plot(u1, u2, main="Uniform-Transformed Data",
134   xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
135 Udex = (1:n)/(n+1)
136 Cn = C.n(u = cbind(rep(Udex, n), rep(Udex, each=n)) , U = U.hat,
137   offset=0, method="C")
138 EmpCop = expression(contour(Udex,Udex,matrix(Cn,n,n), col=2, add=T))
139 #

```

```

140 contour(normalCopula(param=0,dim=2), pCopula, main=expression(C[0]),
141           xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
142 eval(EmpCop)
143 #
144 contour(tCopula(param=Ct@estimate[1], dim=2,
145                   df=round(Ct@estimate[2])),
146           pCopula, main = expression(hat(C)[t]),
147           xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
148 eval(EmpCop)
149 #
150 contour(normalCopula(param=Cgauss@estimate[1], dim = 2),
151           pCopula, main = expression(hat(C)[Gauss]),
152           xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
153 eval(EmpCop)
154 #
155 contour(frankCopula(param=Cfr@estimate[1], dim = 2),
156           pCopula, main = expression(hat(C)[Fr]),
157           xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
158 eval(EmpCop)
159 #
160 contour(claytonCopula(param=Ccl@estimate[1], dim = 2),
161           pCopula, main = expression(hat(C)[Cl]),
162           xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
163 eval(EmpCop)

```

The analysis in the previous paragraph was repeated with the flows transformed by their empirical CDFs. Doing this yielded the semiparametric pseudo-maximum likelihood estimates. Since the results were very similar to those for parametric pseudo-maximum likelihood estimates, they are not presented here. \square

8.8 Bibliographic Notes

For discussion of Archimedean copula with non-strict generators, see McNeil, Frey, and Embrechts (2005). These authors discuss a number of other topics in more detail than is done here. They discuss methods defining nonexchangeable Archimedean copulas. The coefficients of tail dependence for Gaussian and t -copulas are derived in their Sect. 5.2. The theorem and calibration methods in Sect. 8.7.3 are discussed in their Sect. 5.5.

Cherubini et al. (2004) treat the application of copulas to finance. Joe (1997) and Nelsen (2007) are standard references on copulas. Chapter 4 of Mari and Kotz (2001) discusses additional copula families.

Li (2000) developed a well-known but controversial model for credit risk using exponentially distributed default times with a Gaussian copula. An article in *Wired* magazine states that Li's Gaussian copula model was “a quick—and

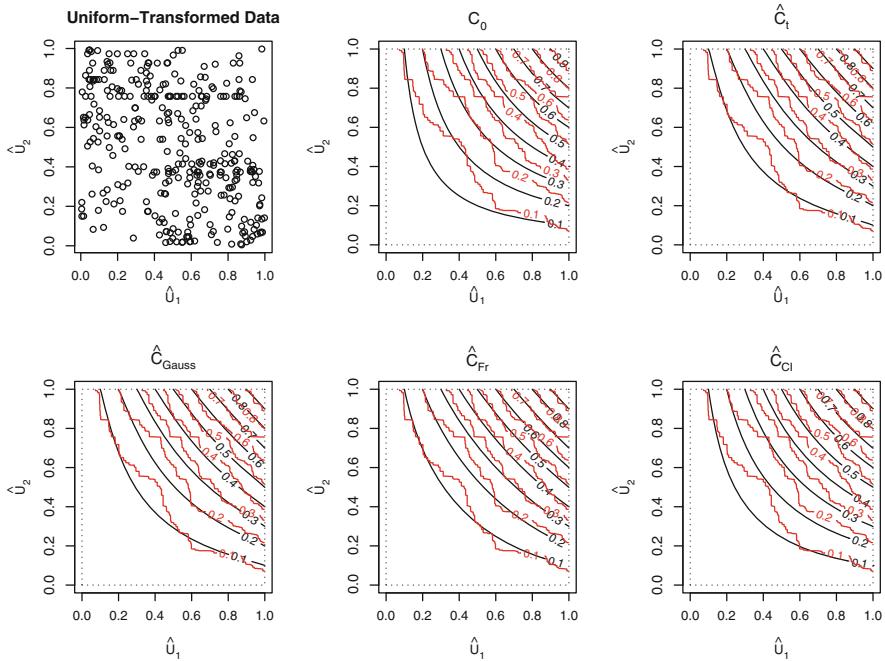


Fig. 8.9. Uniform-transformed flows for pipeline data. Scatterplot; independence copula contours and four fitted copula contours via parametric models, versus the empirical copula contours.

fatally flawed—way to assess risk” (Salmon 2009); in particular, the model does not include tail dependence. Duffie and Singleton’s (2003, Section 10.4) also discusses copula-based methods for modeling dependent default times.

8.9 R Lab

8.9.1 Simulating from Copula Models

Run the R code that appears below to generate data from a copula. Line 1 loads the `copula` package. Lines 2–3 defines a copula object. At this point, nothing is done with the copula object—it is simply defined. However, the copula object is used in line 5 to generate a random sample from the specified copula model. The remaining lines create a scatterplot matrix of the sample and print its sample Pearson correlation matrix.

```

1 library(copula)
2 cop_t_dim3 = tCopula(dim = 3, param = c(-0.6,0.75,0),
3                      dispstr = "un", df = 1)
4 set.seed(5640)
```

```

5 rand_t_cop = rCopula(n = 500, copula = cop_t_dim3)
6 pairs(rand_t_cop)
7 cor(rand_t_cop)

```

You can use R's help to learn more about the functions `tCopula()` and `rCopula()`.

Problem 1 Consider the R code above.

- (a) What type of copula model has been sampled? Give the copula family, the correlation matrix, and any other parameters that specify the copula.
- (b) What is the sample size?

Problem 2 Examine the scatterplot matrix (generated by line 6) and answer the questions below. Include the scatterplot matrix with your answer.

- (a) Components 2 and 3 are uncorrelated. Do they appear independent? Why or why not?
 - (b) Do you see signs of tail dependence? If so, where?
 - (c) What are the effects of dependence upon the plots?
 - (d) The nonzero correlations in the copula do not have the same values as the corresponding sample correlations. Do you think this is just due to random variation or is something else going on? If there is another cause besides random variation, what might that be? To help answer this question, you can get confidence intervals for the Pearson correlation: For example,
- ```

8 cor.test(rand_t_cop[,1],rand_t_cop[,3])

```
- will give a confidence interval (95 percent by default) for the correlation (Pearson by default) between components 1 and 3. Does this confidence interval include 0.75?

Lines 9–10 in the R code below defines a normal (Gaussian) copula. Lines 11–13 define a multivariate distribution by specifying its copula and its marginal distributions—the copula is the one just defined. Line 15 generates a random sample of size 1,000 from this distribution, which has three components. The remaining lines create a scatterplot matrix and kernel estimates of the marginal densities for each component.

```

9 cop_normal_dim3 = normalCopula(dim = 3, param = c(-0.6,0.75,0),
10 dispstr = "un")
11 mvdc_normal = mvdc(copula = cop_normal_dim3, margins = rep("exp",3),
12 paramMargins = list(list(rate=2), list(rate=3),
13 list(rate=4)))
14 set.seed(5640)
15 rand_mvdc = rMvdc(n = 1000, mvdc = mvdc_normal)
16 pairs(rand_mvdc)
17 par(mfrow = c(2,2))
18 for(i in 1:3) plot(density(rand_mvdc[,i]))

```

**Problem 3** Run the R code above to generate a random sample.

- (a) What are the marginal distributions of the three components in `rand_mvdc`? What are their expected values?
- (b) Are the second and third components independent? Why or why not?

### 8.9.2 Fitting Copula Models to Bivariate Return Data

In this section, you will fit copula models to a bivariate data set of daily returns on IBM stock and the S&P 500 index.

First, you will fit a model with univariate marginal  $t$ -distributions and a  $t$ -copula. The model has three degrees-of-freedom (tail index) parameters, one for each of the two univariate models and a third for the copula. This means that the univariate distributions can have different tail indices and that their tail indices are independent of the tail dependence from the copula.

Run the following R code to load the data and necessary libraries, fit univariate  $t$ -distributions to the two components, and convert estimated scale parameters to estimated standard deviations:

```

1 library(MASS) # for fitdistr() and kde2d() functions
2 library(copula) # for copula functions
3 library(fGarch) # for standardized t density
4 netRtns = read.csv("IBM_SP500_04_14_daily_netRtns.csv", header = T)
5 ibm = netRtns[,2]
6 sp500 = netRtns[,3]
7 est.ibm = as.numeric(fitdistr(ibm,"t")$estimate)
8 est.sp500 = as.numeric(fitdistr(sp500,"t")$estimate)
9 est.ibm[2] = est.ibm[2] * sqrt(est.ibm[3] / (est.ibm[3]-2))
10 est.sp500[2] = est.sp500[2] * sqrt(est.sp500[3] / (est.sp500[3]-2))

```

The univariate estimates will be used as starting values when the meta- $t$ -distribution is fit by maximum likelihood. You also need an estimate of the correlation coefficient in the  $t$ -copula. This can be obtained using Kendall's tau. Run the following code and complete line 12 so that `omega` is the estimate of the Pearson correlation based on Kendall's tau.

```

11 cor_tau = cor(ibm, sp500, method = "kendall")
12 omega =

```

**Problem 4** How did you complete line 12 of the code? What was the computed value of `omega`?

Next, define the  $t$ -copula using `omega` as the correlation parameter and 4 as the degrees-of-freedom (tail index) parameter.

```
13 cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)
```

Now fit copulas to the uniform-transformed data.

```
14 data1 = cbind(pstd(ibm, est.ibm[1], est.ibm[2], est.ibm[3]),
15 pstd(sp500, est.sp500[1], est.sp500[2], est.sp500[3]))
16 n = nrow(netRtns) ; n
17 data2 = cbind(rank(ibm)/(n+1), rank(sp500)/(n+1))
18 ft1 = fitCopula(cop_t_dim2, data1, method="ml", start=c(omega,4))
19 ft2 = fitCopula(cop_t_dim2, data2, method="ml", start=c(omega,4))
```

### Problem 5

- (a) Explain the difference between methods used to obtain the two estimates `ft1` and `ft2`.
- (b) Do the two estimates seem significantly different (in a practical sense)?

The next step defines a meta-*t*-distribution by specifying its *t*-copula and its univariate marginal distributions. Values for the parameters in the univariate margins are also specified. The values of the copula parameter were already defined in the previous step.

```
20 mvdc_t_t = mvdc(cop_t_dim2, c("std","std"), list(
21 list(mean=est.ibm[1],sd=est.ibm[2],nu=est.ibm[3]),
22 list(mean=est.sp500[1],sd=est.sp500[2],nu=est.sp500[3])))
```

Now fit the meta *t*-distribution. Be patient. This takes awhile; for instance, it took one minute on my laptop. The elapsed time in minutes will be printed.

```
23 start = c(est.ibm, est.sp500, ft1@estimate)
24 objFn = function(param) -loglikMvdc(param,cbind(ibm,sp500),mvdc_t_t)
25 tic = proc.time()
26 ft = optim(start, objFn, method="L-BFGS-B",
27 lower = c(-.1,0.001,2.2, -0.1,0.001,2.2, 0.2,2.5),
28 upper = c(.1, 10, 15, 0.1, 10, 15, 0.9, 15))
29 toc = proc.time()
30 total_time = toc - tic ; total_time[3]/60
```

Lower and upper bounds are used to constrain the algorithm to stay inside a region where the log-likelihood is defined and finite. The function `fitMvdc()` in the `copula` package does not allow setting lower and upper bounds and did not converge on this problem.

### Problem 6

- (a) What are the estimates of the copula parameters in `fit_cop`?
- (b) What are the estimates of the parameters in the univariate marginal distributions?

- (c) Was the estimation method maximum likelihood, semiparametric pseudo-maximum likelihood, or parametric pseudo-maximum likelihood?
- (d) Estimate the coefficient of lower tail dependence for this copula.

Now fit normal (Gaussian), Frank, Clayton, Gumbel and Joe copulas to the data.

```

31 fnorm = fitCopula(copula=normalCopula(dim=2),data=data1,method="ml")
32 ffrank = fitCopula(copula = frankCopula(3, dim = 2),
33 data = data1, method = "ml")
34 fclayton = fitCopula(copula = claytonCopula(1, dim=2),
35 data = data1, method = "ml")
36 fgumbel = fitCopula(copula = gumbelCopula(3, dim=2),
37 data = data1, method = "ml")
38 fjoe = fitCopula(copula=joeCopula(2,dim=2),data=data1,method="ml")

```

The estimated copulas (CDFs) will be compared with the empirical copula.

```

39 Udex = (1:n)/(n+1)
40 Cn = C.n(u=cbind(rep(Udex,n),rep(Udex,each=n)), U=data1, method="C")
41 EmpCop = expression(contour(Udex, Udex, matrix(Cn, n, n),
42 col = 2, add = TRUE))
43 par(mfrow=c(2,3), mgp = c(2.5,1,0))
44 contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
45 pCopula, main = expression(hat(C)[t]),
46 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
47 eval(EmpCop)
48 contour(normalCopula(param=fnorm@estimate[1], dim = 2),
49 pCopula, main = expression(hat(C)[Gauss]),
50 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
51 eval(EmpCop)
52 contour(frankCopula(param=ffrank@estimate[1], dim = 2),
53 pCopula, main = expression(hat(C)[Fr]),
54 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
55 eval(EmpCop)
56 contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
57 pCopula, main = expression(hat(C)[Cl]),
58 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
59 eval(EmpCop)
60 contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
61 pCopula, main = expression(hat(C)[Gu]),
62 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
63 eval(EmpCop)
64 contour(joeCopula(param=fjoe@estimate[1], dim = 2),
65 pCopula, main = expression(hat(C)[Joe]),
66 xlab = expression(hat(U)[1]), ylab = expression(hat(U)[2]))
67 eval(EmpCop)

```

**Problem 7** Do you see any difference between the parametric estimates of the copula? If so, which seem closest to the empirical copula? Include the plot with your work.

A two-dimensional KDE of the copula's density will be compared with the parametric density estimates (PDFs).

```

68 par(mfrow=c(2,3), mgp = c(2.5,1,0))
69 contour(tCopula(param=ft$par[7],dim=2,df=round(ft$par[8])),
70 dCopula, main = expression(hat(c)[t]),
71 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
72 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
73 contour(normalCopula(param=fnorm@estimate[1], dim = 2),
74 dCopula, main = expression(hat(c)[Gauss]),
75 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
76 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
77 contour(frankCopula(param=ffrank@estimate[1], dim = 2),
78 dCopula, main = expression(hat(c)[Fr]),
79 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
80 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
81 contour(claytonCopula(param=fclayton@estimate[1], dim = 2),
82 dCopula, main = expression(hat(c)[Cl]),
83 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
84 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
85 contour(gumbelCopula(param=fgumbel@estimate[1], dim = 2),
86 dCopula, main = expression(hat(c)[Gu]),
87 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
88 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)
89 contour(joeCopula(param=fjoe@estimate[1], dim = 2),
90 dCopula, main = expression(hat(c)[Joe]),
91 nlevels=25, xlab=expression(hat(U)[1]),ylab=expression(hat(U)[2]))
92 contour(kde2d(data1[,1],data1[,2]), col = 2, add = TRUE)

```

**Problem 8** Do you see any difference between the parametric estimates of the copula density? If so, which seem closest to the KDE? Include the plot with your work.

**Problem 9** Find AIC for the  $t$ , (Gaussian), Frank, Clayton, Gumbel and Joe copulas. Which copula model fits best by AIC? (Hint: The `fitCopula()` function returns the log-likelihood.)

## 8.10 Exercises

1. Kendall's tau rank correlation between  $X$  and  $Y$  is 0.55. Both  $X$  and  $Y$  are positive. What is Kendall's tau between  $X$  and  $1/Y$ ? What is Kendall's tau between  $1/X$  and  $1/Y$ ?

2. Suppose that  $X$  is Uniform(0,1) and  $Y = X^2$ . Then the Spearman rank correlation and the Kendall's tau between  $X$  and  $Y$  will both equal 1, but the Pearson correlation between  $X$  and  $Y$  will be less than 1. Explain why.
3. Show that an Archimedean copula with generator function  $\varphi(u) = -\log(u)$  is equal to the independence copula  $C_0$ . Does the same hold when the natural logarithm is replaced by the common logarithm, i.e.,  $\varphi(u) = -\log_{10}(u)$ ?
4. The co-monotonicity copula  $C_+$  is not an Archimedean copula; however, in the two-dimensional case, the counter-monotonicity copula  $C_-(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$  is. What is its generator function?
5. Show that the generator of a Frank copula

$$\varphi_{\text{Fr}}(u|\theta) = -\log \left\{ \frac{e^{-\theta u} - 1}{e^{-\theta} - 1} \right\}, \quad \theta \in \{(-\infty, 0) \cup (0, \infty)\},$$

satisfies assumptions 1–3 of a strict generator.

6. Show that as  $\theta \rightarrow \infty$ ,  $C_{\text{Fr}}(u_1, u_2|\theta) \rightarrow \min(u_1, u_2)$ , the co-monotonicity copula  $C_+$ .
7. Suppose that  $\varphi_1, \dots, \varphi_k$  are  $k$  strict generator functions and define a new generator  $\varphi$  as a convex combination of these  $k$  generators, that is

$$\varphi(u) = a_1\varphi_1(u) + \dots + a_k\varphi_k(u),$$

in which  $a_1, \dots, a_k$  are any non-negative constants which sum to 1. Show that  $\varphi(u)$  is a strict generator function. For the case in which  $k = 2$ , what is the corresponding copula function for  $\varphi(u)$ ?

8. Let  $\varphi(u|\theta) = (1 - u)^\theta$ , for some  $\theta \geq 1$ , and show that for the two-dimensional case this generates the copula

$$C(u_1, u_2|\theta) = \max[0, 1 - \{(1 - u_1)^\theta + (1 - u_2)^\theta\}^{1/\theta}].$$

Further, show that as  $\theta \rightarrow \infty$ ,  $C(u_1, u_2|\theta) \rightarrow \min(u_1, u_2)$ , the co-monotonicity copula  $C_+$ , and that as  $\theta \rightarrow 1$ ,  $C(u_1, u_2|\theta) \rightarrow \max(u_1 + u_2 - 1, 0)$ , the counter-monotonicity copula  $C_-$ .

9. A convex combination of  $k$  joint CDFs is itself a joint CDF (finite mixture), but is a convex combination of  $k$  copula functions a copula function itself?
10. Suppose  $\mathbf{Y} = (Y_1, \dots, Y_d)$  has a meta-Gaussian distribution with continuous marginal distributions and copula  $C^{\text{Gauss}}(\cdot|\Omega)$ . Show that if  $\rho_\tau(Y_i, Y_j) = 0$  then  $Y_i$  and  $Y_j$  are independent.

## References

Cherubini, U., Luciano, E., and Vecchiato, W. (2004) *Copula Methods in Finance*, John Wiley, New York.

- Duffie, D. and Singleton, K. J. (2003) *Credit Risk*, Princeton University Press, Princeton and Oxford.
- Joe, H. (1997) *Multivariate Models and Dependence Concepts*, Chapman & Hall, London.
- Li, D (2000) On default correlation: A copula function approach, *Journal of Fixed Income*, **9**, 43–54.
- Mari, D. D. and Kotz, S. (2001) *Correlation and Dependence*, World Scientific, London.
- McNeil, A., Frey, R., and Embrechts, P. (2005) *Quantitative Risk Management*, Princeton University Press, Princeton and Oxford.
- Nelsen, R. B. (2007) *An Introduction to Copulas*, 2nd ed., Springer, New York.
- Salmon, F. (2009) Recipe for Disaster: The Formula That Killed Wall Street, *Wired* [http://www.wired.com/techbiz/it/magazine/17-03/wp\\_quant?currentPage=all](http://www.wired.com/techbiz/it/magazine/17-03/wp_quant?currentPage=all)

---

## Regression: Basics

### 9.1 Introduction

Regression is one of the most widely used of all statistical methods. For univariate regression, the available data are one response variable and  $p$  predictor variables, all measured on each of  $n$  observations. We let  $Y$  denote the response variable and  $X_1, \dots, X_p$  be the predictor or explanatory variables. Also,  $Y_i$  and  $X_{i,1}, \dots, X_{i,p}$  are the values of these variables for the  $i$ th observation. The goals of regression modeling include the investigation of how  $Y$  is related to  $X_1, \dots, X_p$ , estimation of the conditional expectation of  $Y$  given  $X_1, \dots, X_p$ , and prediction of future  $Y$  values when the corresponding values of  $X_1, \dots, X_p$  are already available. These goals are closely connected.

The *multiple linear regression* model relating  $Y$  to the predictor or regressor variables is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i, \quad (9.1)$$

where  $\epsilon_i$  is called the noise, disturbances, or errors. The adjective “multiple” refers to the predictor variables. Multivariate regression, which has more than one response variable, is covered in Chap. 18. The  $\epsilon_i$  are often called “errors” because they are the prediction errors when  $Y_i$  is predicted by  $\beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}$ . It is assumed that

$$E(\epsilon_i | X_{i,1}, \dots, X_{i,p}) = 0, \quad (9.2)$$

which, with (9.1), implies that

$$E(Y_i | X_{i,1}, \dots, X_{i,p}) = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}.$$

The parameter  $\beta_0$  is the intercept. The regression coefficients  $\beta_1, \dots, \beta_p$  are the slopes. More precisely,  $\beta_j$  is the partial derivative of the expected response with respect to the  $j$ th predictor:

$$\beta_j = \frac{\partial E(Y_i|X_{i,1}, \dots, X_{i,p})}{\partial X_{i,j}}.$$

Therefore,  $\beta_j$  is the change in the expected value of  $Y_i$  when  $X_{i,j}$  changes one unit. It is assumed that the noise is i.i.d. white so that

$$\epsilon_1, \dots, \epsilon_n \text{ are i.i.d. with mean 0 and variance } \sigma_\epsilon^2. \quad (9.3)$$

Often the  $\epsilon_i$ s are assumed to be normally distributed, which with (9.3) implies Gaussian white noise.

For the reader's convenience, the assumptions of the linear regression model are summarized:

1. linearity of the conditional expectation:  $E(Y_i|X_{i,1}, \dots, X_{i,p}) = \beta_0 + \beta_1 X_{i,1} + \dots + \beta_p X_{i,p}$ ;
2. independent noise:  $\epsilon_1, \dots, \epsilon_n$  are independent;
3. constant variance:  $\text{Var}(\epsilon_i) = \sigma_\epsilon^2$  for all  $i$ ;
4. Gaussian noise:  $\epsilon_i$  is normally distributed for all  $i$ .

This chapter and, especially, the next two chapters discuss methods for checking these assumptions, the consequences of their violations, and possible remedies when they do not hold.

## 9.2 Straight-Line Regression

*Straight-line regression* is linear regression with only one predictor variable. The model is

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \quad (9.4)$$

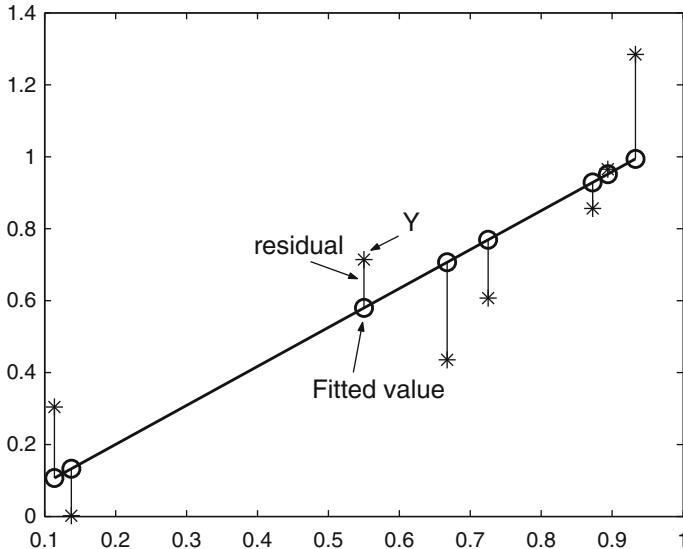
where  $\beta_0$  and  $\beta_1$  are the unknown intercept and slope of the line and  $\epsilon_i$  is called the noise or error.

### 9.2.1 Least-Squares Estimation

The regression coefficients can be estimated by the *method of least squares*. The least-squares estimates are the values of  $\hat{\beta}_0$  and  $\hat{\beta}_1$  that minimize

$$\sum_{i=1}^n \left\{ Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_i) \right\}^2. \quad (9.5)$$

Geometrically, we are minimizing the sum of the squared lengths of the vertical lines in Fig. 9.1. The data points are shown as asterisks. The vertical lines connect the data points and the predictions using the linear equation. The predictions themselves are called the *fitted values* or “ $y$ -hats” and shown as open circles. The differences between the  $Y$ -values and the fitted values are called the *residuals*. Using calculus to minimize (9.5), one can show that



**Fig. 9.1.** Least-squares estimation. The vertical lines connect the data (\*) and the fitted values (o) represent the residuals. The least-squares line is defined as the line making the sum of the squared residuals as small as possible.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^n (X_i - \bar{X})^2} = \frac{\sum_{i=1}^n Y_i(X_i - \bar{X})}{\sum_{i=1}^n (X_i - \bar{X})^2}. \quad (9.6)$$

and

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}. \quad (9.7)$$

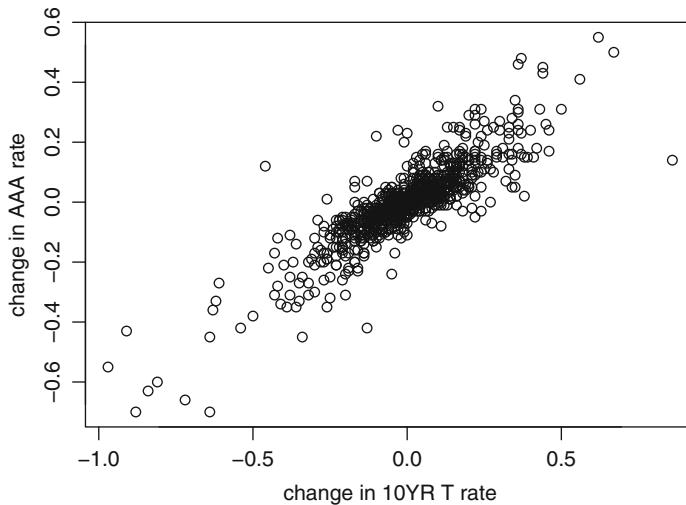
The *least-squares line* is

$$\begin{aligned} \hat{Y} &= \hat{\beta}_0 + \hat{\beta}_1 X = \bar{Y} + \hat{\beta}_1(X - \bar{X}) \\ &= \bar{Y} + \left\{ \frac{\sum_{i=1}^n (Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^n (X_i - \bar{X})^2} \right\} (X - \bar{X}) \\ &= \bar{Y} + \frac{s_{XY}}{s_X^2}(X - \bar{X}), \end{aligned}$$

where  $s_{XY} = (n-1)^{-1} \sum_{i=1}^n (Y_i - \bar{Y})(X_i - \bar{X})$  is the sample covariance between  $X$  and  $Y$  and  $s_X^2$  is the sample variance of  $X$ .

*Example 9.1. Weekly interest rates — least-squares estimates*

Weekly interest rates from February 16, 1977, to December 31, 1993, were obtained from the Federal Reserve Bank of Chicago. Figure 9.2 is a plot of



**Fig. 9.2.** Changes in Moody's seasoned corporate AAA bond yields plotted against changes in 10-year Treasury constant maturity rate. Data from Federal Reserve Statistical Release H.15 and were taken from the Chicago Federal Bank's website.

changes in the 10-year Treasury constant maturity rate and changes in the Moody's seasoned corporate AAA bond yield. The plot looks linear, so we try linear regression using R's `lm()` function. The code is:

```
options(digits = 3)
summary(lm(ddd_dif ~ cm10_dif))
```

The code `ddd_dif ~ cm10_dif` is an example of a formula in R with the outcome variable to the left of “`~`” and the explanatory variables to the right of “`~`”. In this example, there is only one explanatory variable. In cases where there are multiple explanatory variables, they are separated by “`+`”. Here is the output.

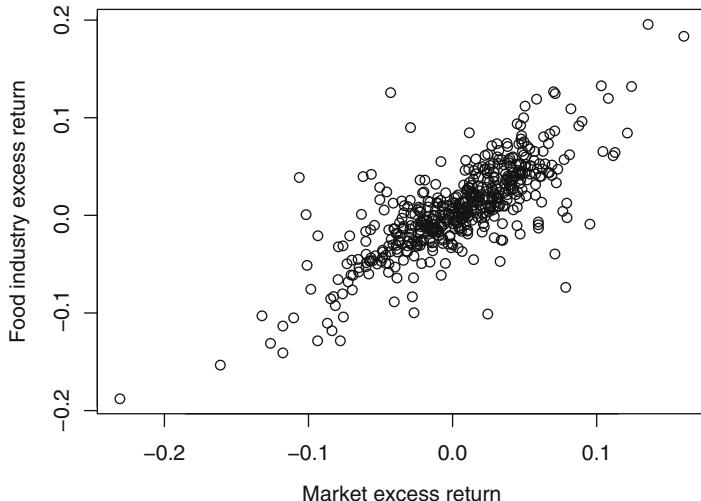
```
Call:
lm(formula = ddd_dif ~ cm10_dif)

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.000109 0.002221 -0.05 0.96
cm10_dif 0.615762 0.012117 50.82 <2e-16 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.066 on 878 degrees of freedom
Multiple R-Squared: 0.746, Adjusted R-squared: 0.746
F-statistic: 2.58e+03 on 1 and 878 DF, p-value: <2e-16
```

From the output we see that the least-squares estimates of the intercept and slope are  $-0.000109$  and  $0.616$ . The **Residual standard error** is  $0.066$ ; this is what we call  $\hat{\sigma}_\epsilon$  or  $s$ , the estimate of  $\sigma_\epsilon$ ; see Sect. 9.3. The remaining items of the output are explained shortly.  $\square$



**Fig. 9.3.** Plot of excess returns on the food industry versus excess returns on the market. Data from the data set `Capm` in R's `Ecdat` package.

### Example 9.2. Excess returns on the food sector and the market portfolio

The excess return on a security or market index is the return minus the risk-free interest rate. An important application of linear regression in finance is the regression of the excess return of an asset or market sector on the excess return of the entire market. This type of application will be discussed much more fully in Chap. 17. In this example, we will regress the excess monthly return of the food sector (`rfood`) on the excess monthly return of the market portfolio (`rmrf`). The data are in R's `Capm` data set in the `Ecdat` package and are plotted in Fig. 9.3. The returns are expressed as percentages in the data set but have been converted to fractions in this example. The output from `lm` is

```
Call:
lm(formula = rfood ~ rmrf)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 0.00339 0.00128 2.66 0.0081 **
rmrf 0.78342 0.02835 27.63 <2e-16 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 0.0289 on 514 degrees of freedom
Multiple R-Squared: 0.598, Adjusted R-squared: 0.597
F-statistic: 763 on 1 and 514 DF, p-value: <2e-16
```

Thus, the fitted regression equation is

$$rfood = 0.00339 + 0.78342 \text{rmrf} + \epsilon,$$

and  $\hat{\sigma}_\epsilon = 0.0289$ . □

### 9.2.2 Variance of $\hat{\beta}_1$

It is useful to have a formula for the variance of an estimator to show how the estimator's precision depends on various aspects of the data such as the sample size and the values of the predictor variables. Fortunately, it is easy to derive a formula for the variance of  $\hat{\beta}_1$ . By (9.6), we can write  $\hat{\beta}_1$  as a weighted average of the responses

$$\hat{\beta}_1 = \sum_{i=1}^n w_i Y_i,$$

where  $w_i$  is the weight given by

$$w_i = \frac{X_i - \bar{X}}{\sum_{i=1}^n (X_i - \bar{X})^2}.$$

We consider  $X_1, \dots, X_n$  as fixed, so if they are random we are conditioning upon their values. From the assumptions of the regression model, it follows that  $\text{Var}(Y_i|X_1, \dots, X_n) = \sigma_\epsilon^2$  and  $Y_1, \dots, Y_n$  are conditionally uncorrelated. Therefore,

$$\text{Var}(\hat{\beta}_1|X_1, \dots, X_n) = \sigma_\epsilon^2 \sum_{i=1}^n w_i^2 = \frac{\sigma_\epsilon^2}{\sum_{i=1}^n (X_i - \bar{X})^2} = \frac{\sigma_\epsilon^2}{(n-1)s_X^2}. \quad (9.8)$$

It is worth taking some time to examine this formula. First, the numerator  $\sigma_\epsilon^2$  is simply the variance of the  $\epsilon_i$ . This is not surprising. More variability in the noise means more variable estimators. The denominator shows us that the variance of  $\hat{\beta}_1$  is inversely proportional to  $(n-1)$  and to  $s_X^2$ . So the precision of  $\hat{\beta}_1$  increases as  $\sigma_\epsilon^2$  is reduced,  $n$  is increased, or  $s_X^2$  is increased. Why does increasing  $s_X^2$  decrease  $\text{Var}(\hat{\beta}_1|X_1, \dots, X_n)$ ? The reason is that increasing  $s_X^2$  means that the  $X_i$  are spread farther apart, which makes the slope of the line easier to estimate.

*Example 9.3. Optimal sampling frequencies for regression*

Here is an important application of (9.8). Suppose that we have two stationary time series,  $X_t$  and  $Y_t$ , and we wish to regress  $Y_t$  on  $X_t$ . We have just seen examples of this. A significant practical question is whether one should use daily or weekly data, or perhaps even monthly or quarterly data. Does it matter which sampling frequency we use? The answer is “yes” and the highest possible sampling frequency gives the most precise estimate of the slope. To understand why this is so, we compare daily and weekly data. Assume that the  $X_t$  and  $Y_t$  are white noise sequences. Since a weekly log return is simply the sum of the five daily log returns within a week,  $\sigma_\epsilon^2$  and  $s_X^2$  will each increase by a factor of five if we change from daily to weekly log returns, so the ratio  $\sigma_\epsilon^2/s_X^2$  will not change. However, by changing from daily to weekly log returns,  $(n - 1)$  is reduced by approximately a factor of five. The result is that  $\text{Var}(\hat{\beta}_1|X_1, \dots, X_n)$  is approximately five times smaller using daily rather than weekly log returns. Similarly,  $\text{Var}(\hat{\beta}_1|X_1, \dots, X_n)$  is about four times larger using monthly rather than weekly returns.

The obvious conclusion is that one should use the highest sampling frequency available, which is often daily returns. We have assumed that the  $X_t$  and  $Y_t$  are white noise in order to simplify the calculations, but this conclusion still holds if they are stationary but autocorrelated. (Autocorrelation is discussed in Chap. 12.) However, the noise series, that is  $\epsilon_i$ ,  $i = 1, \dots$ , in Eq. (9.4) needs to be uncorrelated. If the noise is autocorrelated and becomes more highly correlated as the sampling frequency increases, then this conclusion need not hold. There may be a point of diminishing returns where more frequent sampling does not improve estimation accuracy.  $\square$

### 9.3 Multiple Linear Regression

The multiple linear regression model is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i.$$

The least-squares estimates are the values  $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$  that minimize

$$\sum_{i=1}^n \left\{ Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{i,1} + \cdots + \hat{\beta}_p X_{i,p}) \right\}^2. \quad (9.9)$$

Calculation of the least-squares estimates is discussed in Sect. 11.1. For applications, the technical details are not important, since software for least-squares estimation is readily available.

The  $i$ th *fitted value* is

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i,1} + \cdots + \hat{\beta}_p X_{i,p} \quad (9.10)$$

and estimates  $E(Y_i|X_{i,1}, \dots, X_{i,p})$ . The  $i$ th residual is

$$\hat{\epsilon}_i = Y_i - \hat{Y}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{i,1} + \dots + \hat{\beta}_p X_{i,p}) \quad (9.11)$$

and estimates  $\epsilon_i$ . It is worth noting that (9.11) can be re-expressed as

$$Y_i = \hat{Y}_i + \hat{\epsilon}_i. \quad (9.12)$$

An unbiased estimate of  $\sigma^2_\epsilon$  is

$$\hat{\sigma}^2_\epsilon = \frac{\sum_{i=1}^n \hat{\epsilon}_i^2}{n - 1 - p}. \quad (9.13)$$

The denominator in (9.13) is the sample size minus the number of regression coefficients that are estimated.

#### *Example 9.4. Multiple linear regression with interest rates*

As an example, we continue the analysis of the weekly interest-rate data but now with changes in the 30-year Treasury rate (`cm30_dif`) and changes in the Federal funds rate (`ff_dif`) as additional predictors. Thus  $p = 3$ . Figure 9.4 is a scatterplot matrix of the four time series. There is a strong linear relationship between all pairs of `aaa_dif`, `cm10_dif`, and `cm30_dif`, but `ff_dif` is not strongly related to the other series. The code is

```
summary(lm(aaa_dif ~ cm10_dif + cm30_dif + ff_dif))
```

The `lm()` output for this regression is

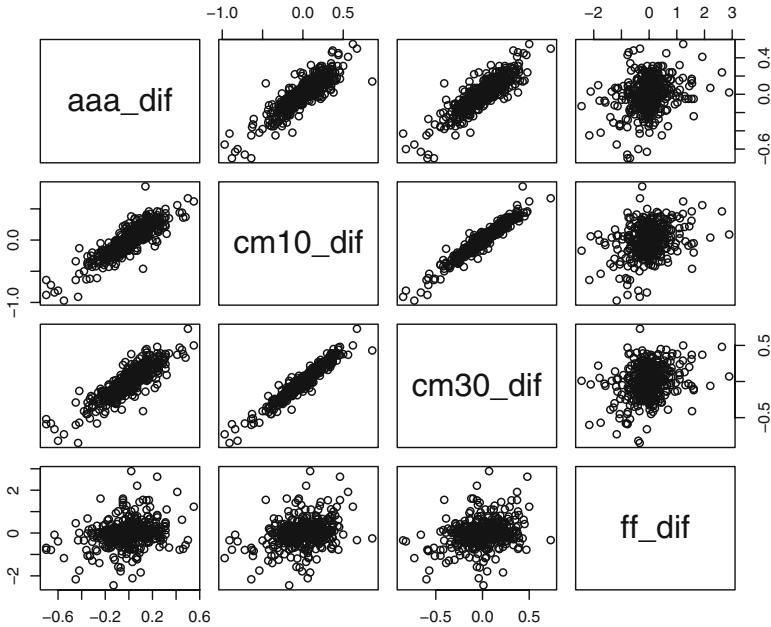
```
Call:
lm(formula = aaa_dif ~ cm10_dif + cm30_dif + ff_dif)

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.07e-05 2.18e-03 -0.04 0.97
cm10_dif 3.55e-01 4.51e-02 7.86 1.1e-14 ***
cm30_dif 3.00e-01 5.00e-02 6.00 2.9e-09 ***
ff_dif 4.12e-03 5.28e-03 0.78 0.44

Residual standard error: 0.0646 on 876 degrees of freedom
Multiple R-Squared: 0.756, Adjusted R-squared: 0.755
F-statistic: 906 on 3 and 876 DF, p-value: <2e-16
```

We see that  $\hat{\beta}_0 = -9.07 \times 10^{-5}$ ,  $\hat{\beta}_1 = 0.355$ ,  $\hat{\beta}_2 = 0.300$ , and  $\hat{\beta}_3 = 0.00412$ .  $\square$

A commonly used special case of multiple regression is the polynomial regression model which uses powers of the predictors as well as the predictors



**Fig. 9.4.** Scatterplot matrix of the changes in four weekly interest rates. The variable `aaa_dif` is the response in Example 9.4.

themselves. For example, when there is one  $X$ -variable, the  $p$ -degree polynomial regression model is

$$Y_i = \beta_0 + \beta_1 X_i + \cdots + \beta_p X_i^p + \epsilon_i.$$

As another example, the quadratic regression model with two predictors is

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,1}^2 + \beta_3 X_{i,1} X_{i,2} + \beta_4 X_{i,2} + \beta_5 X_{i,2}^2 + \epsilon_i.$$

### 9.3.1 Standard Errors, $t$ -Values, and $p$ -Values

In this section we explain the use of several statistics included in regression output. We use the output in Example 9.4 as an illustration.

As noted before, the estimated coefficients are  $\hat{\beta}_0 = -9.07 \times 10^{-5}$ ,  $\hat{\beta}_1 = 0.355$ ,  $\hat{\beta}_2 = 0.300$ , and  $\hat{\beta}_3 = 0.00412$ . Each of these coefficients has three other statistics associated with it.

- The standard error (SE), which is the estimated standard deviation of the least-squares estimator, tells us the precision of the estimator.
- The  $t$ -value, is the  $t$ -statistic for testing that the coefficient is 0. The  $t$ -value is the ratio of the estimate to its standard error. For example, for `cm10_dif`, the  $t$ -value is  $7.86 = 0.355/0.0451$ .

- The  $p$ -value ( $\text{Pr} > |\mathbf{t}|$  in the `lm()` output), associated with testing the null hypothesis that the coefficient is 0 versus the alternative that it is not 0. If a  $p$ -value for a slope parameter is small, as it is here for  $\beta_1$ , then this is evidence that the corresponding coefficient is *not* 0, which means that the predictor has a *linear* relationship with the response.

It is important to keep in mind that the  $p$ -value only tells us if there is a linear relationship. The existence of a linear relationship between  $Y_i$  and  $X_{i,j}$  means only that the linear predictor of  $Y_i$  has a nonzero slope on  $X_{i,j}$ , or, equivalently, that partial correlation between  $X_{i,j}$  and  $Y_i$  is not zero. (The partial correlation between two variables is their correlation when all other variables are held fixed.) When the  $p$ -value is small (so a linear relationship exists), there could also be a strong nonlinear deviation from the linear relationship as in Fig. A.4g. Moreover, when the  $p$ -value is large (so no linear relationship exists), there could still be a strong nonlinear relationship in Fig. A.4f. Because of the potential for nonlinear relationships to go undetected in a linear regression analysis, graphical analysis of the data (e.g., Fig. 9.4) and residual analysis (see Chap. 10) are essential.

The  $p$ -values for  $\beta_1$  and  $\beta_2$  are *very* small, so we can conclude that these slopes are *not* 0. The  $p$ -value is large (0.97) for  $\beta_0$ , so we would not reject the hypothesis that the intercept is 0.

Similarly, we would not reject the null hypothesis that  $\beta_3$  is zero. Stated differently, we can accept the null hypothesis that, conditional on `cm10_dif` and `cm30_dif`, `aaa_dif` and `ff_dif` are not linearly related. This result should *not* be interpreted as stating that `aaa_dif` and `ff_dif` are unrelated, but only that `ff_dif` is not useful for predicting `aaa_dif` when `cm10_dif` and `cm30_dif` are included in the regression model. (In fact, `aaa_dif` and `ff_dif` have a correlation of 0.25 (this is the full, not partial, correlation) and the linear regression of `aaa_dif` on `ff_dif` alone is highly significant; the  $p$ -value for testing that the slope is zero is  $5.158 \times 10^{-14}$ .)

Since the Federal Funds rate is a short-term (overnight) rate, it is not surprising that `ff_dif` is less useful than changes in the 10- and 30-year Treasury rates for predicting `aaa_dif`.

For regression with one predictor variable, by (9.8) the standard error of  $\hat{\beta}_1$  is  $\hat{\sigma}_\epsilon / \sqrt{\sum_{i=1}^n (X_i - \bar{X})^2}$ . When there are more than two predictor variables, formulas of standard errors are more complex and are facilitated by the use of matrix notation. Because standard errors can be computed with standard software such as `lm`, the formulas are not needed for applications and so are postponed to Sect. 11.1.

## 9.4 Analysis of Variance, Sums of Squares, and $R^2$

### 9.4.1 ANOVA Table

Certain results of a regression fit are often displayed in an *analysis of variance table*, also called the ANOVA or AOV table. The idea behind the ANOVA table is to describe how much of the variation in  $Y$  is predictable if one knows  $X_1, \dots, X_p$ . Here is the ANOVA table for the model in Example 9.4.

```
> anova(lm(ddd_dif ~ cm10_dif + cm30_dif + ff_dif))
Analysis of Variance Table

Response: ddd_dif
 Df Sum Sq Mean Sq F value Pr(>F)
cm10_dif 1 11.21 11.21 2682.61 < 2e-16 ***
cm30_dif 1 0.15 0.15 35.46 3.8e-09 ***
ff_dif 1 0.0025 0.0025 0.61 0.44
Residuals 876 3.66 0.0042

```

The total variation in  $Y$  can be partitioned into two parts: the variation that can be predicted by  $X_1, \dots, X_p$  and the variation that cannot be predicted. The variation that can be predicted is measured by the regression sum of squares, which is

$$\text{regression SS} = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2.$$

The regression sum of squares for the model that uses only `cm10_dif` is in the first row of the ANOVA table and is 11.21. The entry, 0.15, in the second row is the increase in the regression sum of squares when `cm30_dif` is added to the model. Similarly, 0.0025 is the increase in the regression sum of squares when `ff_dif` is added. Thus, rounding to two decimal places,  $11.36 = 11.21 + 0.15 + 0.00$  is the regression sum of squares with all three predictors in the model.

The amount of variation in  $Y$  that cannot be predicted by a linear function of  $X_1, \dots, X_p$  is measured by the residual error sum of squares, which is the sum of the squared residuals; i.e.,

$$\text{residual error SS} = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$

In the ANOVA table, the residual error sum of squares is in the last row and is 3.66. The total variation is measured by the total sum of squares (total SS), which is the sum of the squared deviations of  $Y$  from its mean; that is,

$$\text{total SS} = \sum_{i=1}^n (Y_i - \bar{Y})^2. \quad (9.14)$$

It can be shown algebraically that

$$\text{total SS} = \text{regression SS} + \text{residual error SS}. \quad (9.15)$$

Therefore, in Example 9.4, the total SS is  $11.36 + 3.66 = 15.02$ .

R-squared, denoted by  $R^2$ , is

$$R^2 = \frac{\text{regression SS}}{\text{total SS}} = 1 - \frac{\text{residual error SS}}{\text{total SS}}$$

and measures the proportion of the total variation in  $Y$  that can be linearly predicted by  $X$ . In the example,  $R^2$  is  $0.746 = 11.21/15.02$  if only `cm10_dif` is the model and is  $11.36/15.02 = 0.756$  if all three predictors are in the model. This value can be found in the output displayed in Example 9.4.

When there is only a single  $X$  variable, then  $R^2 = r_{XY}^2 = r_{\hat{Y}Y}^2$ , where  $r_{XY}$  and  $r_{\hat{Y}Y}$  are the sample correlations between  $Y$  and  $X$  and between  $Y$  and the predicted values, respectively. Put differently,  $R^2$  is the squared correlation between  $Y$  and  $X$  and also between  $Y$  and  $\hat{Y}$ . When there are multiple predictors, then we still have  $R^2 = r_{\hat{Y}Y}^2$ . Since  $\hat{Y}$  is a linear combination of the  $X$  variables,  $R$  can be viewed as the “multiple” correlation between  $Y$  and many  $X$ s. The residual error sum of squares is also called the error sum of squares or sum of squared errors and is denoted by SSE.

It is important to understand that sums of squares in an ANOVA table depend upon the order of the predictor variables in the regression, because the sum of squares for any variable is the increase in the regression sum of squares when that variable is added to the predictors already in the model.

The table below has the same variables as before, but the order of the predictor variables is reversed. Now that `ff_dif` is the first predictor, its sum of squares is much larger than before and its  $p$ -value is highly significant; before it was nonsignificant, only 0.44. The sum of squares for `cm30_dif` is now much larger than that of `cm10_dif`, the reverse of what we saw earlier, since `cm10_dif` and `cm30_dif` are highly correlated and the first of them in the list of predictors will have the larger sum of squares.

```
> anova(lm(aaa_dif ~ ff_dif + cm30_dif + cm10_dif))
Analysis of Variance Table

Response: aaa_dif
 Df Sum Sq Mean Sq F value Pr(>F)
ff_dif 1 0.94 0.94 224.8 < 2e-16 ***
cm30_dif 1 10.16 10.16 2432.1 < 2e-16 ***
cm10_dif 1 0.26 0.26 61.8 1.1e-14 ***
Residuals 876 3.66 0.0042
```

The lesson here is that an ANOVA table is most useful for assessing the effects of adding predictors in some natural order. Since AAA bonds have maturities closer to 10 than to 30 years, and since the Federal Funds rate is an overnight rate, it made sense to order the predictors as `cm10_dif`, `cm30_dif`, and `ff_dif` as done initially.

### 9.4.2 Degrees of Freedom (DF)

There are degrees of freedom (DF) associated with each of these sources of variation. The degrees of freedom for regression is  $p$ , which is the number of predictor variables. The total degrees of freedom is  $n - 1$ . The residual error degrees of freedom is  $n - p - 1$ . Here is a way to think of degrees of freedom. Initially, there are  $n$  degrees of freedom, one for each observation. Then one degree of freedom is allocated to estimation of the intercept. This leaves a total of  $n - 1$  degrees of freedom for estimating the effects of the  $X$  variables and  $\sigma_\epsilon^2$ . Each regression parameter uses one degree of freedom for estimation. Thus, there are  $(n - 1) - p$  degrees of freedom remaining for estimation of  $\sigma_\epsilon^2$  using the residuals. There is an elegant geometrical theory of regression where the responses are viewed as lying in an  $n$ -dimensional vector space and degrees of freedom are the dimensions of various subspaces. However, there is not sufficient space to pursue this subject here.

### 9.4.3 Mean Sums of Squares (MS) and $F$ -Tests

As just discussed, every sum of squares in an ANOVA table has an associated degrees of freedom. The ratio of the sum of squares to the degrees of freedom is the mean sum of squares:

$$\text{mean sum of squares} = \frac{\text{sum of squares}}{\text{degrees of freedom}}.$$

The residual mean sum of squares is the unbiased estimate  $\hat{\sigma}_\epsilon^2$  given by (9.13); that is,

$$\begin{aligned}\hat{\sigma}_\epsilon^2 &= \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n - 1 - p} \\ &= \text{residual mean sum of squares} \\ &= \frac{\text{residual error SS}}{\text{residual degrees of freedom}}.\end{aligned}\tag{9.16}$$

Other mean sums of squares are used in testing. Suppose we have two models, I and II, and the predictor variables in model I are a subset of those in model II, so that model I is a submodel of II. A common null hypothesis is that the data are generated by model I. Equivalently, in model II the slopes are zero for variables not also in model I. To test this hypothesis, we use the excess regression sum of squares of model II relative to model I:

$$\begin{aligned}\text{SS(II | I)} &= \text{regression SS for model II} - \text{regression SS for model I} \\ &= \text{residual SS for model I} - \text{residual SS for model II}.\end{aligned}\tag{9.17}$$

Equality (9.17) holds because (9.15) is true for all models and, in particular, for both model I and model II. The degrees of freedom for  $\text{SS(II | I)}$  is the number

of extra predictor variables in model II compared to model I. The mean square is denoted as  $\text{MS}(\text{II} | \text{I})$ . Stated differently, if  $p_{\text{I}}$  and  $p_{\text{II}}$  are the number of parameters in models I and II, respectively, then  $\text{df}_{\text{II}| \text{I}} = p_{\text{II}} - p_{\text{I}}$  and  $\text{MS}(\text{II} | \text{I}) = \text{SS}(\text{II} | \text{I})/\text{df}_{\text{II}| \text{I}}$ . The  $F$ -statistic for testing the null hypothesis is

$$F = \frac{\text{MS}(\text{II}| \text{I})}{\hat{\sigma}_\epsilon^2},$$

where  $\hat{\sigma}_\epsilon^2$  is the mean residual sum of squares for model II. Under the null hypothesis, the  $F$ -statistic has an  $F$ -distribution with  $\text{df}_{\text{II}| \text{I}}$  and  $n - p_{\text{II}} - 1$  degrees of freedom and the null hypothesis is rejected if the  $F$ -statistic exceeds the  $\alpha$ -upper quantile of this  $F$ -distribution.

*Example 9.5. Weekly interest rates—Testing the one-predictor versus three-predictor model*

In this example, the null hypothesis is that, in the three-predictor model, the slopes for `cm30_dif` and `ff_dif` are zero. The  $F$ -test can be computed using R's `anova` function. The output is

```
Analysis of Variance Table

Model 1: aaa_dif ~ cm10_dif
Model 2: aaa_dif ~ cm10_dif + cm30_dif + ff_dif
Res.Df RSS Df Sum of Sq F Pr(>F)
1 878 3.81
2 876 3.66 2 0.15 18.0 2.1e-08 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

In the last row, the entry 2 in the “Df” column is the difference between the two models in the number of parameters and 0.15 in the “Sum of Sq” column is the difference between the residual sum of squares (RSS) for the two models.

The very small  $p$ -value ( $2.1 \times 10^{-8}$ ) leads us to reject the null hypothesis and say that the result is “highly significant.” It is important to be aware that this phrase refers to statistical significance. When the sample size is as large as it is here, it is common to reject the null hypothesis. The reason for this is that the null hypothesis is rarely true exactly, and with a large sample size it is highly likely that even a small deviation from the null hypothesis will be detected. Statistically significance must be distinguished from practical significance. The adjusted  $R^2$  values for the two- and three-variable models are very similar, 0.746 and 0.755, respectively. Therefore, the rejection of the two-variable model may not be of practical importance.  $\square$

*Example 9.6. Weekly interest rates—Testing a two-predictor versus three-predictor model*

In this example, the null hypothesis is that, in the three predictor model, the slope `ff_dif` is zero. The  $F$ -test is again computed using R's `anova` function with output:

```
Analysis of Variance Table

Model 1: aaa_dif ~ cm10_dif + cm30_dif
Model 2: aaa_dif ~ cm10_dif + cm30_dif + ff_dif
Res.Df RSS Df Sum of Sq F Pr(>F)
1 877 3.66
2 876 3.66 1 0.0025 0.61 0.44
```

The large  $p$ -value (0.44) leads us to accept the null hypothesis. Notice that this is the same as the  $p$ -value for `ff_dif` in the ANOVA table in Sect. 9.4.1. This is not a coincidence. Both  $p$ -values are the same because they are testing the same hypothesis.  $\square$

#### 9.4.4 Adjusted $R^2$

$R^2$  is biased in favor of large models, because  $R^2$  is always increased by adding more predictors to the model, even if they are independent of the response. Recall that

$$R^2 = 1 - \frac{\text{residual error SS}}{\text{total SS}} = 1 - \frac{n^{-1}\text{residual error SS}}{n^{-1}\text{total SS}}.$$

The bias in  $R^2$  can be reduced by using the following “adjustment,” which replaces both occurrences of  $n$  by the appropriate degrees of freedom:

$$\text{adjusted } R^2 = 1 - \frac{(n - p - 1)^{-1}\text{residual error SS}}{(n - 1)^{-1}\text{total SS}} = 1 - \frac{\text{residual error MS}}{\text{total MS}}.$$

The presence of  $p$  in the adjusted  $R^2$  penalizes the criterion for the number of predictor variables, so adjusted  $R^2$  can either increase or decrease when predictor variables are added to the model. Adjusted  $R^2$  increases if the added variables decrease the residual sum of squares enough to compensate for the increase in  $p$ .

## 9.5 Model Selection

When there are many potential predictor variables, often we wish to find a subset of them that provide a parsimonious regression model.  $F$ -tests are not very suitable for model selection. One problem is that there are many possible  $F$ -tests and the joint statistical behavior of all of them is not known. For model selection, it is more appropriate to use a model selection criterion such as AIC or BIC. For linear regression models, AIC is

$$\text{AIC} = n \log(\hat{\sigma}^2) + 2(1 + p),$$

where  $1 + p$  is the number of parameters in a model with  $p$  predictor variables; the intercept gives us the final parameter. BIC replaces  $2(1 + p)$  in AIC by  $\log(n)(1 + p)$ . The first term,  $n \log(\hat{\sigma}^2)$ , is equal to, up to an additive constant that does not affect model comparisons,  $-2$  times the log-likelihood evaluated at the MLE, assuming that the noise is Gaussian.

In addition to AIC and BIC, there are two model selection criteria specialized for regression. One is adjusted  $R^2$ , which we have seen before. Another is  $C_p$ .  $C_p$  is related to AIC and usually  $C_p$  and AIC are minimized by the same model. The primary reason for using  $C_p$  instead of AIC is that some regression software computes only  $C_p$ , not AIC—this is true of the `regsubsets()` function in R's `leaps` package which will be used in the following example.

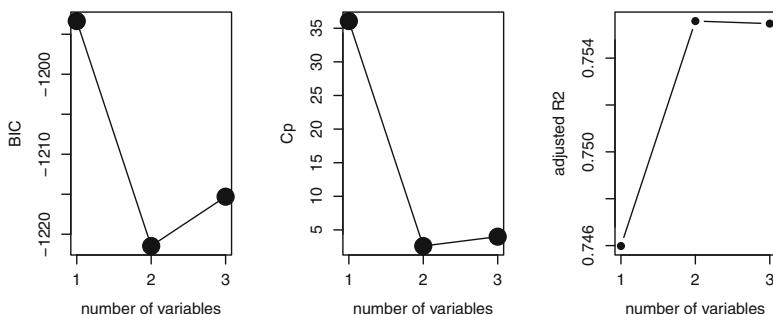
To define  $C_p$ , suppose there are  $M$  predictor variables. Let  $\hat{\sigma}_{\epsilon,M}^2$  be the estimate of  $\sigma_\epsilon^2$  using all of them, and let  $\text{SSE}(p)$  be the sum of squares for residual error for a model with some subset of only  $p \leq M$  of the predictors. As usual,  $n$  is the sample size. Then  $C_p$  is

$$C_p = \frac{\text{SSE}(p)}{\hat{\sigma}_{\epsilon,M}^2} - n + 2(p + 1). \quad (9.18)$$

Of course,  $C_p$  will depend on which particular model is used among all of those with  $p$  predictors, so the notation “ $C_p$ ” may not be ideal.

With  $C_p$ , AIC, and BIC, smaller values are better, but for adjusted  $R^2$ , larger values are better.

One should not use model selection criteria blindly. Model choice should be guided by economic theory and practical considerations, as well as by model selection criteria. It is important that the final model makes sense to the user. Subject-matter expertise might lead to adoption of a model not optimal according to the criterion being used but, instead, to a model slightly below optimal but more parsimonious or with a better economic rationale.



**Fig. 9.5.** Changes in weekly interest rates. Plots for model selection.

*Example 9.7. Weekly interest rates—Model selection by AIC and BIC*

Figure 9.5 contains plots of the number of predictors in the model versus the optimized value of a selection criterion. By “optimized value,” we mean the best value among all models with the given number of predictor variables. “Best” means smallest for BIC and  $C_p$  and largest for adjusted  $R^2$ . There are three plots, one for each of BIC,  $C_p$ , and adjusted  $R^2$ . All three criteria are optimized by two predictor variables.

There are three models with two of the three predictors. The one that optimized the criteria<sup>1</sup> is the model with `cm10_dif` and `cm30_dif`, as can be seen in the following output from `regsubsets`. Here "\*" indicates a variable in the model and " " indicates a variable not in the model, so the three rows of the table indicate that the best one-variable model is `cm10_dif` and the best two-variable model is `cm10_dif` and `cm30_dif`—the third row does not contain any real information since, with only three variables, there is only one possible three-variable model.

```
Selection Algorithm: exhaustive
 cm10_dif cm30_dif ff_dif
1 (1) "*" " " " "
2 (1) "*" "*" " "
3 (1) "*" "*" "*"
```

□

## 9.6 Collinearity and Variance Inflation

If two or more predictor variables are highly correlated with one another, then it is difficult to estimate their separate effects on the response. For example, `cm10_dif` and `cm30_dif` have a correlation of 0.96 and the scatterplot in Fig. 9.4 shows that they are highly related to each other. If we regress `aaa_dif` on `cm10_dif`, then the adjusted  $R^2$  is 0.7460, but adjusted  $R^2$  only increases to 0.7556 if we add `cm30_dif` as a second predictor. This suggests that `cm30_dif` might not be related to `aaa_dif`, but this is not the case. In fact, the adjusted  $R^2$  is 0.7376 when `cm30_dif` is the only predictor, which indicates that `cm30_dif` is a good predictor of `aaa_dif`, nearly as good as `cm10_dif`.

Another effect of the high correlation between the predictor variables is that the regression coefficient for each variable is very sensitive to whether the other variable is in the model. For example, the coefficient of `cm10_dif` is 0.616 when `cm10_dif` is the sole predictor variable but only 0.360 if `cm30_dif` is also included.

---

<sup>1</sup> When comparing models with the same number of parameters, all three criteria are optimized by the same model.

The problem here is that `cm10_dif` and `cm30_dif` provide redundant information because of their high correlation. This problem is called *collinearity* or, in the case of more than two predictors, *multicollinearity*. Collinearity increases standard errors. The standard error of the  $\beta$  of `cm10_dif` is 0.01212 when only `cm10_dif` is in the model, but increases to 0.0451, a 372 % increase, if `cm30_dif` is added to the model.

The *variance inflation factor (VIF)* of a variable tells us how much the squared standard error, i.e., the variance of  $\hat{\beta}$ , of that variable is increased by having the other predictor variables in the model. For example, if a variable has a VIF of 4, then the variance of its  $\hat{\beta}$  is four times larger than it would be if the other predictors were either deleted or were not correlated with it. The standard error is increased by a factor of 2.

Suppose we have predictor variables  $X_1, \dots, X_p$ . Then the VIF of  $X_j$  is found by regressing  $X_j$  on the  $p - 1$  other predictors. Let  $R_j^2$  be the  $R^2$ -value of this regression, so that  $R_j^2$  measures how well  $X_j$  can be predicted from the other  $X$ s. Then the VIF of  $X_j$  is

$$\text{VIF}_j = \frac{1}{1 - R_j^2}.$$

A value of  $R_j^2$  close to 1 implies a large VIF. In other words, the more accurately that  $X_j$  can be predicted from the other  $X$ s, the more redundant it is and the higher its VIF. The minimum value of  $\text{VIF}_j$  is 1 and occurs when  $R_j^2$  is 0. There is, unfortunately, no upper bound to  $\text{VIF}_j$ . Variance inflation becomes infinite as  $R_j^2$  approaches 1.

When interpreting VIFs, it is important to keep in mind that  $\text{VIF}_j$  tells us nothing about the relationship between the response and  $j$ th predictor. Rather, it tells us only how correlated the  $j$ th predictor is with the other predictors. In fact, the VIFs can be computed without knowing the values of the response variable.

The usual remedy to collinearity is to reduce the number of predictor variables by using one of the model selection criteria discussed in Sect. 9.5.

*Example 9.8. Variance inflation factors for the weekly interest-rate example.*

The function `vif()` in R's `faraway` library returned the following VIF values for the changes in weekly interest rates:

```
> library(faraway)
> options(digits = 2)
> vif(lm(aaa_dif ~ cm10_dif + cm30_dif + ff_dif))
 cm10_dif cm30_dif ff_dif
 14.4 14.1 1.1
```

`cm10_dif` and `cm30_dif` have large VIFs due to their high correlation with each other. The predictor `ff_dif` is not highly correlated with `cm10_dif` and `cm30_dif` and has a lower VIF.

VIF values give us information about linear relationships between the predictor variables, but not about their relationships with the response. In this example, `ff_dif` has a small VIF value but is not an important predictor because of its low correlation with the response. Despite their high VIF values, `cm10_dif` and `cm30_dif` are important predictors. The high VIF values tell us only that the regression coefficients for `cm10_dif` and `cm30_dif` are impossible to estimate with high precision.

The question is whether VIF values of 14.4 and 14.1 are so large that the number of predictor variables should be reduced to 1, that is, whether we should use only `cm10_dif`. The answer is “perhaps not” because the model with both `cm10_dif` and `cm30_dif` minimizes BIC. BIC generally selects a parsimonious model because of the high penalty BIC places on the number of predictor variables. Therefore, a model that minimizes BIC is unlikely to need further deletion of predictor variables simply to reduce VIF values. However, we saw earlier that adding `cm30_dif` to the model with `cm10_dif` offers only a minor increase in adjusted  $R^2$ , so the issue of whether or not to include `cm30_dif` is not clear.  $\square$

### *Example 9.9. Nelson–Plosser macroeconomic variables*

To illustrate model selection, we now turn to an example with more predictors. We will start with six predictors but will find that a model with only two predictors fits rather well.

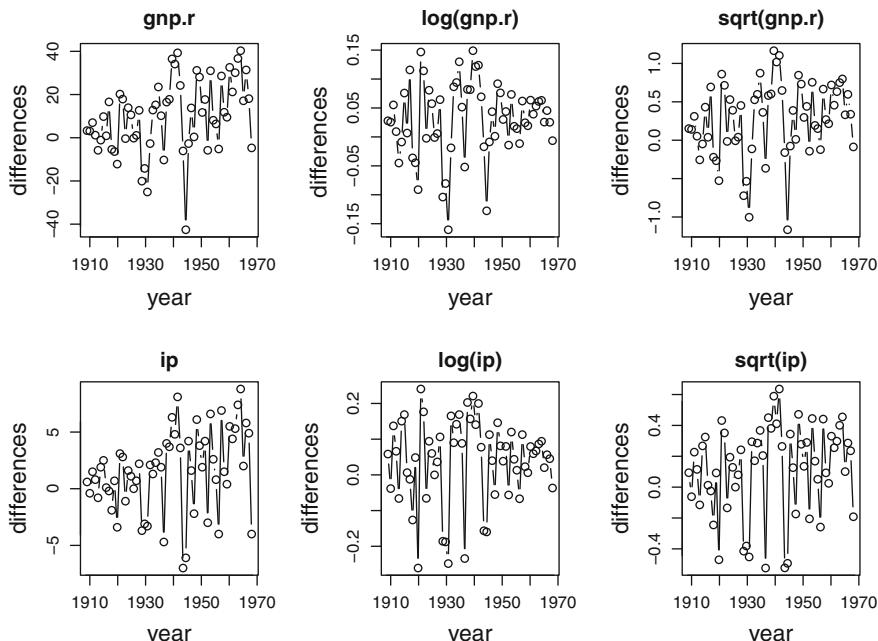
This example uses a subset of the well-known Nelson–Plosser data set of U.S. yearly macroeconomic time series. These data are available in the file `nelsonplosser.csv`. The variables we will use are:

1. sp-Stock Prices, [Index; 1941-43 = 100], [1871–1970].
2. gnp.r-Real GNP, [Billions of 1958 Dollars], [1909–1970],
3. gnp.pc-Real Per Capita GNP, [1958 Dollars], [1909–1970],
4. ip-Industrial Production Index, [1967 = 100], [1860–1970],
5. cpi-Consumer Price Index, [1967 = 100], [1860–1970],
6. emp-Total Employment, [Thousands], [1890–1970],
7. bnd-Basic Yields 30-year Corporate Bonds, [% pa], [1900–1970].

Since two of the time series start in 1909, we use only the data from 1909 until the end of the series in 1970, a total of 62 years. The response will be the differences of `log(sp)`, the log returns on the stock prices. The regressors will be the differences of variables 2 through 7, with variables 4 and 5 log-transformed before differencing. A differenced log-series contains the approximate relative changes in the original variable, in the same way that a log return approximates a return that is the relative change in price.

How does one decide whether to difference the original series, the log-transformed series, or some other function of the series? Usually the aim is to stabilize the fluctuations in the differenced series. The top row of Fig. 9.6 has time series plots of changes in `gnp.r`, `log(gnp.r)`, and `sqrt(gnp.r)` and the bottom row has similar plots for `ip`. For `ip` the fluctuations in the differenced series increase steadily over time, but this is less true if one uses the square roots or logs of the series. This is the reason why `diff(log(ip))` is used here as a regressor. For `gnp.r`, the fluctuations in changes are more stable and we used `diff(gnp.r)` rather than `diff(log(gnp.r))` as a regressor. In this analysis, we did not consider using square-root transformations, since changes in the square roots are less interpretable than changes in the original variable or its logarithm. However, the changes in the square roots of both series are reasonably stable, so square-root transformations might be considered. Another possibility would be to use the transformation that gives the best-fitting model. One could, for example, put all three variables, `diff(ip)`, `diff(log(ip))`, and `diff(sqrt(ip))`, into the model and use model selection to decide which gives the best fit. The same could be done with `gnp.r` and the other regressors.

Notice that the variables are transformed first and then differenced. Differencing first and then taking logarithms or square roots would result in complex-valued variables, which would be difficult to interpret, to say the least.



**Fig. 9.6.** Differences in `gnp.r` and `ip` with and without transformations.

There are additional variables in this data set that could be tried in the model. The analysis presented here is only an illustration and much more exploration is certainly possible with this rich data set.

Time series and normal plots of all eight differenced series did not reveal any outliers. The normal plots were only used to check for outliers, not to check for normal distributions. There is no assumption in a regression analysis that the regressors are normally distributed or that the response has a marginal normal distribution. It is only the conditional distribution of the response given the regressors that is assumed to be normal, and even that assumption can be weakened.

A linear regression with all of the regressors shows that only two, `diff(log(ip))` and `diff(bnd)`, are statistically significant at the 0.05 level and some have very large *p*-values:

```
Call:
lm(formula = diff(log(sp)) ~ diff(gnp.r) + diff(gnp.pc)
+ diff(log(ip)) + diff(log(cpi))
+ diff(emp) + diff(bnd), data = new_np)

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.766e-02 3.135e-02 -0.882 0.3815
diff(gnp.r) 8.384e-03 4.605e-03 1.821 0.0742
diff(gnp.pc) -9.752e-04 9.490e-04 -1.028 0.3087
diff(log(ip)) 6.245e-01 2.996e-01 2.085 0.0418
diff(log(cpi)) 4.935e-01 4.017e-01 1.229 0.2246
diff(emp) -9.591e-06 3.347e-05 -0.287 0.7756
diff(bnd) -2.030e-01 7.394e-02 -2.745 0.0082
```

A likely problem here is multicollinearity, so variance inflation factors were computed:

|                          |                           |                            |                             |
|--------------------------|---------------------------|----------------------------|-----------------------------|
| <code>diff(gnp.r)</code> | <code>diff(gnp.pc)</code> | <code>diff(log(ip))</code> | <code>diff(log(cpi))</code> |
| 16.0                     | 31.8                      | 3.3                        | 1.3                         |
| <code>diff(emp)</code>   | <code>diff(bnd)</code>    |                            |                             |
| 10.9                     | 1.5                       |                            |                             |

We see that `diff(gnp.r)` and `diff(gnp.pc)` have high VIF values, which is not surprising since they are expected to be highly correlated. In fact, their correlation is 0.96.

Next, we search for a more parsimonious model using `stepAIC()`, a variable selection procedure in R that starts with a user-specified model and adds or deletes variables sequentially. At each step it either makes the addition or deletion that most improves AIC. In this example, `stepAIC()` will start with all six predictors.

Here is the first step:

```
Start: AIC=-224.92
diff(log(sp)) ~ diff(gnp.r) + diff(gnp.pc) + diff(log(ip)) +
 diff(log(cpi)) + diff(emp) + diff(bnd)
```

|                  | Df | Sum of Sq | RSS   | AIC      |
|------------------|----|-----------|-------|----------|
| - diff(emp)      | 1  | 0.002     | 1.216 | -226.826 |
| - diff(gnp.pc)   | 1  | 0.024     | 1.238 | -225.737 |
| - diff(log(cpi)) | 1  | 0.034     | 1.248 | -225.237 |
| <none>           |    |           | 1.214 | -224.918 |
| - diff(gnp.r)    | 1  | 0.075     | 1.289 | -223.284 |
| - diff(log(ip))  | 1  | 0.098     | 1.312 | -222.196 |
| - diff(bnd)      | 1  | 0.169     | 1.384 | -218.949 |

The listed models have either zero or one variable removed from the starting model with all regressors. The models are listed in order of their AIC values. The first model, which has `diff(emp)` removed (the minus sign indicates a variable that has been removed), has the best (smallest) AIC. Therefore, in the first step, `diff(emp)` is removed. Notice that the fourth-best model has no variables removed.

The second step starts with the model without `diff(emp)` and examines the effect on AIC of removing additional variables. The removal of `diff(log(cpi))` leads to the largest improvement in AIC, so in the second step this variable is removed:

```
Step: AIC=-226.83
diff(log(sp)) ~ diff(gnp.r) + diff(gnp.pc) + diff(log(ip)) +
 diff(log(cpi)) + diff(bnd)
```

|                  | Df | Sum of Sq | RSS   | AIC      |
|------------------|----|-----------|-------|----------|
| - diff(log(cpi)) | 1  | 0.032     | 1.248 | -227.236 |
| <none>           |    |           | 1.216 | -226.826 |
| - diff(gnp.pc)   | 1  | 0.057     | 1.273 | -226.025 |
| - diff(gnp.r)    | 1  | 0.084     | 1.301 | -224.730 |
| - diff(log(ip))  | 1  | 0.096     | 1.312 | -224.179 |
| - diff(bnd)      | 1  | 0.189     | 1.405 | -220.032 |

On the third step no variables are removed and the process stops:

```
Step: AIC=-227.24
diff(log(sp)) ~ diff(gnp.r) + diff(gnp.pc) + diff(log(ip)) +
 diff(bnd)
```

|                 | Df | Sum of Sq | RSS   | AIC      |
|-----------------|----|-----------|-------|----------|
| <none>          |    |           | 1.248 | -227.236 |
| - diff(gnp.pc)  | 1  | 0.047     | 1.295 | -227.001 |
| - diff(gnp.r)   | 1  | 0.069     | 1.318 | -225.942 |
| - diff(log(ip)) | 1  | 0.122     | 1.371 | -223.534 |
| - diff(bnd)     | 1  | 0.157     | 1.405 | -222.001 |

Notice that the removal of `diff(gnp.pc)` would cause only a very small increase in AIC. We should investigate whether this variable might be removed. The new model was refit to the data.

**Coefficients:**

|               | Estimate  | Std. Error | t value | Pr(> t ) |
|---------------|-----------|------------|---------|----------|
| (Intercept)   | -0.018664 | 0.028723   | -0.65   | 0.518    |
| diff(gnp.r)   | 0.007743  | 0.004393   | 1.76    | 0.083    |
| diff(gnp.pc)  | -0.001029 | 0.000712   | -1.45   | 0.154    |
| diff(log(ip)) | 0.672924  | 0.287276   | 2.34    | 0.023    |
| diff(bnd)     | -0.177490 | 0.066840   | -2.66   | 0.010    |

Residual standard error: 0.15 on 56 degrees of freedom  
 Multiple R-squared: 0.347, Adjusted R-squared: 0.3  
 F-statistic: 7.44 on 4 and 56 DF, p-value: 7.06e-05

Now three of the four variables are statistically significant at 0.1, though `diff(gnp.pc)` has a rather large *p*-value, and it seems to be worth exploring other possible models.

The R function `leaps()` in the `leaps` package will compute  $C_p$  for all possible models. To reduce the amount of output, only the `nbest` models with  $k$  regressors [for each  $k = 1, \dots, \dim(\beta)$ ] are printed. The value of `nbest` is selected by the user and in this analysis `nbest` was set at 1, so only the best model is given for each value of  $k$ . The following table gives the value of  $C_p$  (last column) for the best  $k$ -variable models, for  $k = 1, \dots, 6$  ( $k$  is in the first column). The remaining columns indicate with a “1” which variables are in the models. All predictors have been differenced, but to save space “diff” has been omitted from the variable names heading the columns.

|   | gnp.r | gnp.pc | log(ip) | log(cpi) | emp | bnd | Cp  |
|---|-------|--------|---------|----------|-----|-----|-----|
| 1 | 0     | 0      | 1       | 0        | 0   | 0   | 6.3 |
| 2 | 0     | 0      | 1       | 0        | 0   | 1   | 3.8 |
| 3 | 1     | 0      | 1       | 0        | 0   | 1   | 4.6 |
| 4 | 1     | 1      | 1       | 0        | 0   | 1   | 4.5 |
| 5 | 1     | 1      | 1       | 1        | 0   | 1   | 5.1 |
| 6 | 1     | 1      | 1       | 1        | 1   | 1   | 7.0 |

We see that `stepAIC` stopping at the four-variable model was perhaps premature. The model selection process was stopped at the four-variable model because the three-variable model had a slightly larger  $C_p$ -value. However, if one continues to the best two-variable model, the minimum of  $C_p$  is obtained. Here is the fit to the best two-variable model:

```
Call:
lm(formula = diff(log(sp)) ~ +diff(log(ip)) + diff(bnd),
 data = new_np)
```

**Residuals:**

| Min      | 1Q       | Median  | 3Q      | Max     |
|----------|----------|---------|---------|---------|
| -0.44254 | -0.09786 | 0.00377 | 0.10525 | 0.28136 |

Coefficients:

|               | Estimate | Std. Error | t value | Pr(> t ) |
|---------------|----------|------------|---------|----------|
| (Intercept)   | 0.0166   | 0.0210     | 0.79    | 0.43332  |
| diff(log(ip)) | 0.6975   | 0.1683     | 4.14    | 0.00011  |
| diff(bnd)     | -0.1322  | 0.0623     | -2.12   | 0.03792  |

Residual standard error: 0.15 on 58 degrees of freedom  
 Multiple R-squared: 0.309, Adjusted R-squared: 0.285  
 F-statistic: 12.9 on 2 and 58 DF, p-value: 2.24e-05

Both variables are significant at 0.05. However, it is not crucial that all regressors be significant at 0.05 or at any other predetermined level. Other models could be used, especially if there were good economic reasons for doing so. One cannot say that the two-variable model is best, except in the narrow sense of minimizing  $C_p$ , and choosing instead the best three- or four-predictor model would not increase  $C_p$  by much. Also, which model is best depends on the criterion used. The best four-predictor model has a better adjusted  $R^2$  than the best two-predictor model.  $\square$

## 9.7 Partial Residual Plots

A partial residual plot is used to visualize the effect of a predictor on the response while removing the effects of the other predictors. The partial residual for the  $j$ th predictor variable is

$$Y_i - \left( \hat{\beta}_0 + \sum_{j' \neq j} X_{i,j'} \hat{\beta}_{j'} \right) = \hat{Y}_i + \hat{\epsilon}_i - \left( \hat{\beta}_0 + \sum_{j' \neq j} X_{i,j'} \hat{\beta}_{j'} \right) = X_{i,j} \hat{\beta}_j + \hat{\epsilon}_i, \quad (9.19)$$

where the first equality uses (9.12) and the second uses (9.10). Notice that the left-hand side of (9.19) shows that the partial residual is the response with the effects of all predictors but the  $j$ th subtracted off. The right-hand side of (9.19) shows that the partial residual is also equal to the residual with the effect of the  $j$ th variable added back. The partial residual plot is simply the plot of the responses against these partial residuals.

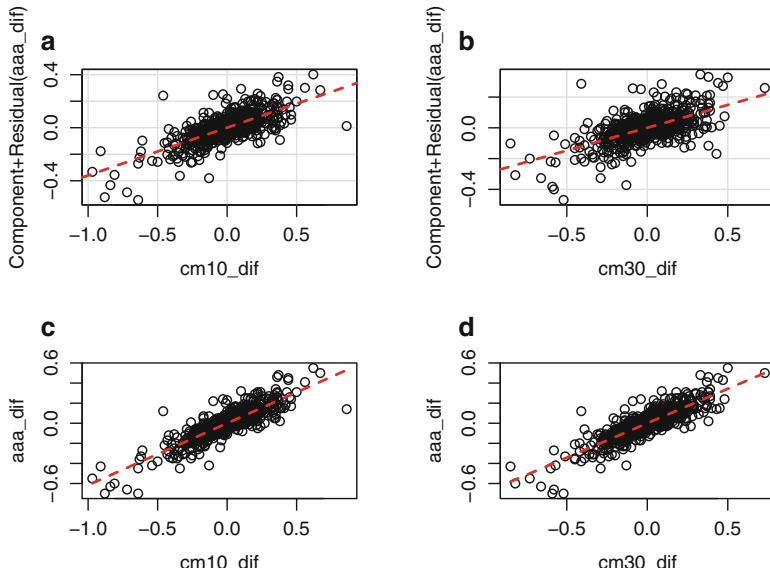
*Example 9.10. Partial residual plots for the weekly interest-rate example*

Partial residual plots for the weekly interest-rate example are shown in Fig. 9.7a, b. For comparison, scatterplots of cm10\_dif and cm30\_dif versus aaa\_dif with the corresponding one-variable fitted lines are shown in panels (c) and (d). The main conclusion from examining the plots is that the slopes in (a) and (b) are shallower than the slopes in (c) and (d). What does this tell

us? It says that, due to collinearity, the effect of `cm10_dif` on `aaa_dif` when `cm30_dif` is in the model [panel (a)] is less than when `cm30_dif` is not in the model [panel (c)], and similarly when the roles of `cm10_dif` and `cm30_dif` are reversed.

The same conclusion can be reached by looking at the estimated regression coefficients. From Examples 9.1 and 9.4, we can see that the coefficient of `cm10_dif` is 0.615 when `cm10_dif` is the only variable in the model, but the coefficient drops to 0.355 when `cm30_dif` is also in the model. There is a similar decrease in the coefficient for `cm30_dif` when `cm10_dif` is added to the model.  $\square$

*Example 9.11. Nelson–Plosser macroeconomic variables—Partial residual Plots*



**Fig. 9.7.** Partial residual plots for the weekly interest rates [panels (a) and (b)] and scatterplots of the predictors and the response [panels (c) and (d)].

This example continues the analysis of the Nelson–Plosser macroeconomic variables. Partial residual plots for the four-variable model selected by `stepAIC` in Example 9.9 are shown in Fig. 9.8. One can see that all four variables have explanatory power, since the partial residuals have linear trends in the variables.

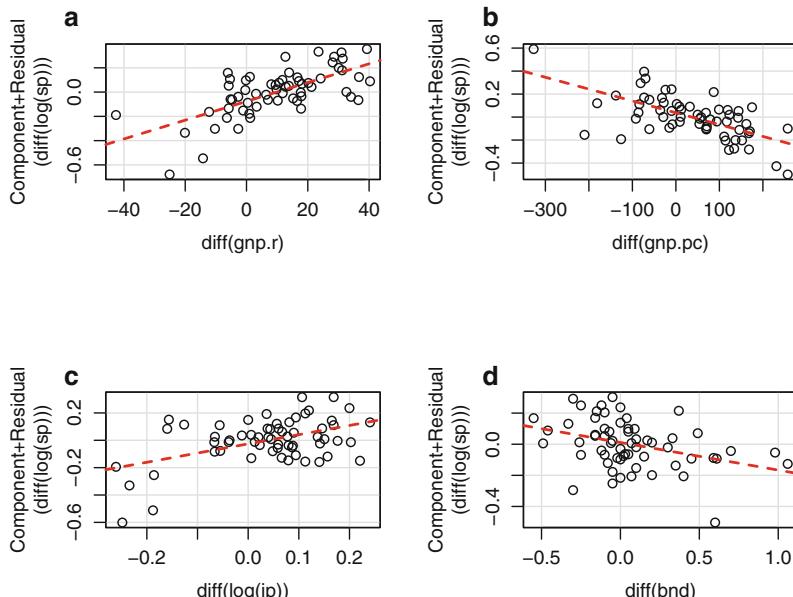
One puzzling aspect of this model is that the slope for `gnp.pc` is negative. However, the  $p$ -value for this regressor is large and the minimum  $C_p$  model

does not contain either `gnp.r` or `gnp.pc`. Often, a regressor that is highly correlated with other regressors has an estimated slope that is counterintuitive. If used alone, both `gnp.r` and `gnp.pc` have positive slopes. The slope of `gnp.pc` is negative only when `gnp.r` is in the model.  $\square$

## 9.8 Centering the Predictors

*Centering* or, more precisely, *mean-centering* a variable means expressing it as a deviation from its mean. Thus, if  $X_{1,k}, \dots, X_{n,k}$  are the values of the  $k$ th predictor and  $\bar{X}_k$  is their mean, then  $(X_{1,k} - \bar{X}_k), \dots, (X_{n,k} - \bar{X}_k)$  are values of the centered predictor.

Centering is useful for two reasons:



**Fig. 9.8.** Partial residual plots for the Nelson–Plosser U.S. economic time series. (a) Change in `gnp.r`. (b) Change in `gnp.pc`. (c) Change in `log(ip)`. (d) Change in `bnd`.

- centering can reduce collinearity in polynomial regression;
- if all predictors are centered, then  $\beta_0$  is the expected value of  $Y$  when each of the predictors is equal to its mean. This gives  $\beta_0$  an interpretable meaning. In contrast, if the variables are not centered, then  $\beta_0$  is the expected value of  $Y$  when all of the predictors are equal to 0. Frequently, 0 is outside the range of some predictors, making the interpretation of  $\beta_0$  of little real interest unless the variables are centered.

## 9.9 Orthogonal Polynomials

As just mentioned, centering can reduce collinearity in polynomial regression because, for example, if  $X$  is positive, then  $X$  and  $X^2$  will be highly correlated but  $X - \bar{X}$  and  $(X - \bar{X})^2$  will be less correlated.

Orthogonal polynomials can eliminate correlation entirely, since they are defined in a way so that they are uncorrelated. This is done using the Gram–Schmidt orthogonalization procedure discussed in textbooks on linear algebra. Orthogonal polynomials can be created easily in most software packages, for instance, by using the `poly()` function in R. Orthogonal polynomials are particularly useful for polynomial regression of degree higher than 2 where centering is less successful at reducing collinearity. However, the use of polynomial models of degree 4 and higher is discouraged and nonparametric regression (see Chap. 21) is recommended instead. Even cubic regression can be problematic because cubic polynomials have only a limited range of shapes.

## 9.10 Bibliographic Notes

Harrell (2001), Ryan (1997), Neter et al. (1996) and Draper and Smith (1998) are four of the many good introductions to regression. Faraway (2005) is an excellent modern treatment of linear regression with R. See Nelson and Plosser (1982) for information about their data set.

## 9.11 R Lab

### 9.11.1 U.S. Macroeconomic Variables

This section uses the data set `USMacroG` in R’s `AER` package. This data set contains quarterly times series on 12 U.S. macroeconomic variables for the period 1950–2000. We will use the variables `consumption` = real consumption expenditures, `dpi` = real disposable personal income, `government` = real government expenditures, and `unemp` = unemployment rate. Our goal is to predict changes in `consumption` from changes in the other variables.

Run the following R code to load the data, difference the data (since we wish to work with changes in these variables), and create a scatterplot matrix.

```
library(AER)
data("USMacroG")
MacroDiff = as.data.frame(apply(USMacroG, 2, diff))
attach(MacroDiff)
pairs(cbind(consumption, dpi, cpi, government, unemp))
```

**Problem 1** *Describe any interesting features, such as outliers, seen in the scatterplot matrix. Keep in mind that the goal is to predict changes in consumption. Which variables seem best suited for that purpose? Do you think there will be collinearity problems?*

Next, run the code below to fit a multiple linear regression model to `consumption` using the other four variables as predictors.

```
fitLm1 = lm(consumption ~ dpi + cpi + government + unemp)
summary(fitLm1)
confint(fitLm1)
```

**Problem 2** *From the summary, which variables seem useful for predicting changes in consumption?*

Next, print an ANOVA table.

```
anova(fitLm1)
```

**Problem 3** *For the purpose of variable selection, does the ANOVA table provide any useful information not already in the summary?*

Upon examination of the  $p$ -values, we might be tempted to drop several variables from the regression model, but we will not do that since variables should be removed from a model one at a time. The reason is that, due to correlation between the predictors, when one is removed the significance of the others changes. To remove variables sequentially, we will use the function `stepAIC()` in the `MASS` package.

```
library(MASS)
fitLm2 = stepAIC(fitLm1)
summary(fitLm2)
```

**Problem 4** *Which variables are removed from the model, and in what order?*

Now compare the initial and final models by AIC.

```
AIC(fitLm1)
AIC(fitLm2)
AIC(fitLm1) - AIC(fitLm2)
```

**Problem 5** *How much of an improvement in AIC was achieved by removing variables? Was the improvement large? Is so, can you suggest why? If not, why not?*

The function `vif()` in the `car` package will compute variance inflation factors. A similar function with the same name is in the `faraway` package. Run

```
library(car)
vif(fitLm1)
vif(fitLm2)
```

**Problem 6** *Was there much collinearity in the original four-variable model? Was the collinearity reduced much by dropping two variables?*

Partial residual plots, which are also called *component plus residual* or *cr* plots, can be constructed using the function `crPlot()` in the `car` package. Run

```
par(mfrow = c(2, 2))
sp = 0.8
crPlot(fitLm1, dpi, span = sp, col = "black")
crPlot(fitLm1, cpi, span = sp, col = "black")
crPlot(fitLm1, government, span = sp, col = "black")
crPlot(fitLm1, unemp, span = sp, col = "black")
```

Besides dashed least-squares lines, the partial residual plots have solid lowess smooths through them unless this feature is turned off by specifying `smooth=F`, as was done in Fig. 9.8. Lowess is an earlier version of loess. The smoothness of the lowess curves is determined by the parameter `span`, with larger values of `span` giving smoother plots. The default is `span = 0.5`. In the code above, `span` is `0.8` but can be changed for all four plots by changing the variable `sp`. Lowess, loess, and `span` are described in Sect. 21.2.1. A substantial deviation of the lowess curve from the least-squares line is an indication that the effect of the predictor is nonlinear. The default color of the `crPlot` figure is red, but this can be changed as in the code above.

**Problem 7** *What conclusions can you draw from the partial residual plots?*

## 9.12 Exercises

- Suppose that  $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ , where  $\epsilon_i$  is  $N(0, 0.3)$ ,  $\beta_0 = 1.4$ , and  $\beta_1 = 1.7$ .
  - What are the conditional mean and standard deviation of  $Y_i$  given that  $X_i = 1$ ? What is  $P(Y_i \leq 3 | X_i = 1)$ ?
  - A regression model is a model for the conditional distribution of  $Y_i$  given  $X_i$ . However, if we also have a model for the marginal distribution of  $X_i$ , then we can find the marginal distribution of  $Y_i$ . Assume that  $X_i$  is  $N(1, 0.7)$ . What is the marginal distribution of  $Y_i$ ? What is  $P(Y_i \leq 3)$ ?

2. Show that if  $\epsilon_1, \dots, \epsilon_n$  are i.i.d.  $N(0, \sigma_\epsilon^2)$ , then in straight-line regression the least-squares estimates of  $\beta_0$  and  $\beta_1$  are also the maximum likelihood estimates.

*Hint:* This problem is similar to the example in Sect. 5.9. The only difference is that in that section,  $Y_1, \dots, Y_n$  are independent  $N(\mu, \sigma^2)$ , while in this exercise  $Y_1, \dots, Y_n$  are independent  $N(\beta_0 + \beta_1 X_i, \sigma_\epsilon^2)$ .

3. Use (7.11), (9.3), and (9.2) to show that (9.8) holds.
4. It was stated in Sect. 9.8 that centering reduces collinearity. As an illustration, consider the example of quadratic polynomial regression where  $X$  takes 30 equally spaced values between 1 and 15.
- What is the correlation between  $X$  and  $X^2$ ? What are the VIFs of  $X$  and  $X^2$ ?
  - Now suppose that we center  $X$  before squaring. What is the correlation between  $(X - \bar{X})$  and  $(X - \bar{X})^2$ ? What are the VIFs of  $(X - \bar{X})$  and  $(X - \bar{X})^2$ ?
5. A linear regression model with three predictor variables was fit to a data set with 40 observations. The correlation between  $Y$  and  $\hat{Y}$  was 0.65. The total sum of squares was 100.
- What is the value of  $R^2$ ?
  - What is the value of the residual error SS?
  - What is the value of the regression SS?
  - What is the value of  $s^2$ ?
6. A data set has 66 observations and five predictor variables. Three models are being considered. One has all five predictors and the others are smaller. Below is residual error SS for all three models. The total SS was 48. Compute  $C_p$  and  $R^2$  for all three models. Which model should be used based on this information?

| Number<br>of predictors | Residual<br>error SS |
|-------------------------|----------------------|
| 3                       | 12.2                 |
| 4                       | 10.1                 |
| 5                       | 10.0                 |

7. The quadratic polynomial regression model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i$$

was fit to data. The  $p$ -value for  $\beta_1$  was 0.67 and for  $\beta_2$  was 0.84. Can we accept the hypothesis that  $\beta_1$  and  $\beta_2$  are both 0? Discuss.

8. Sometimes it is believed that  $\beta_0$  is 0 because we think that  $E(Y|X = 0) = 0$ . Then the appropriate model is

$$y_i = \beta_1 X_i + \epsilon_i.$$

This model is usually called “regression through the origin” since the regression line is forced through the origin. The least-squares estimator of  $\beta_1$  minimizes

$$\sum_{i=1}^n \{Y_i - \beta_1 X_i\}^2.$$

- Find a formula that gives  $\hat{\beta}_1$  as a function of the  $Y_i$ s and the  $X_i$ s.
9. Complete the following ANOVA table for the model  $Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i$ :

| Source     | df | SS   | MS | F | P    |
|------------|----|------|----|---|------|
| Regression | ?  | ?    | ?  | ? | 0.04 |
| Error      | ?  | 5.66 | ?  |   |      |
| Total      | 15 | ?    |    |   |      |

R-sq = ?

10. Pairs of random variables  $(X_i, Y_i)$  were observed. They were assumed to follow a linear regression with  $E(Y_i|X_i) = \theta_1 + \theta_2 X_i$  but with  $t$ -distributed noise, rather than the usual normally distributed noise. More specifically, the assumed model was that conditionally, given  $X_i$ ,  $Y_i$  is  $t$ -distributed with mean  $\theta_1 + \theta_2 X_i$ , standard deviation  $\theta_3$ , and degrees of freedom  $\theta_4$ . Also, the pairs  $(X_1, Y_1), \dots, (X_n, Y_n)$  are mutually independent. The model could also be expressed as

$$Y_i = \theta_1 + \theta_2 X_i + \epsilon_i$$

where  $\epsilon_1, \dots, \epsilon_n$  are i.i.d.  $t$  with mean 0 and standard deviation  $\theta_3$  and degrees of freedom  $\theta_4$ . The model was fit by maximum likelihood. The R code and output are

```
#(Code to input x and y not shown)
library(fGarch)
start = c(lmfit$coef, sd(lmfit$resid), 4)
loglik = function(theta)
{
 -sum(log(dstd(y, mean = theta[1] + theta[2] * x, sd = theta[3],
 nu = theta[4])))
}
mle = optim(start, loglik, hessian = TRUE)
InvFishInfo = solve(mle$hessian)
mle$par
mle$value
mle$convergence
sqrt(diag(InvFishInfo))
qnorm(0.975)

> mle$par
[1] 0.511 1.042 0.152 4.133
> mle$value
[1] -188
```

```
> mle$convergence
[1] 0
> sqrt(diag(InvFishInfo))
[1] 0.00697 0.11522 0.01209 0.93492
>
> qnorm(.975)
[1] 1.96
>
```

- (a) What is the MLE of the slope of  $Y_i$  on  $X_i$ ?
- (b) What is the standard error of the MLE of the degrees-of-freedom parameter?
- (c) Find a 95 % confidence interval for the standard deviation of the noise.
- (d) Did `optim` converge? Why or why not?

## References

- Draper, N. R. and Smith, H. (1998) *Applied Regression Analysis*, 3rd ed., Wiley, New York.
- Faraway, J. J. (2005) *Linear Models with R*, Chapman & Hall, Boca Raton, FL.
- Harrell, F. E., Jr. (2001) *Regression Modeling Strategies*, Springer-Verlag, New York.
- Nelson C.R., and Plosser C.I. (1982) Trends and random walks in macroeconomic time series. *Journal of Monetary Economics*, **10**, 139–162.
- Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996) *Applied Linear Statistical Models*, 4th ed., Irwin, Chicago.
- Ryan, T. P. (1997) *Modern Regression Methods*, Wiley, New York.

## Regression: Troubleshooting

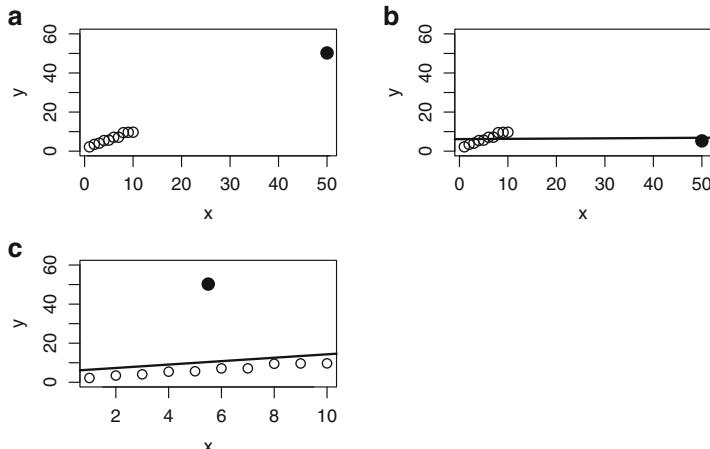
### 10.1 Regression Diagnostics

Many things can, and often do, go wrong when data are analyzed. There may be data that were entered incorrectly, one might not be analyzing the data set one thinks, the variables may have been mislabeled, and so forth. In Example 10.5, presented shortly, one of the weekly time series of interest rates began with 371 weeks of zeros, indicating missing data. However, I was unaware of this problem when I first analyzed the data. The lesson here is that I should have plotted each of the data series first before starting to analyze them, but I hadn't. Fortunately, the diagnostics presented in this section showed quickly that there was some type of serious problem, and then after plotting each of the time series I easily discovered the nature of the problem.

Besides problems with the data, the assumed model may not be a good approximation to reality. The usual estimation methods, such as least squares in regression, are highly nonrobust, which means that they are particularly sensitive to problems with the data or the model.

Experienced data analysts know that they should always look at the raw data. Graphical analysis often reveals any problems that exist, especially the types of gross errors that can seriously degrade the analysis. However, some problems are only revealed by fitting a regression model and examining residuals.

*Example 10.1. High-leverage points and residual outliers—Simulated data example*



**Fig. 10.1.** (a) Linear regression with a high-leverage point that is not a residual outlier (solid circle). (b) Linear regression with a high-leverage point that is a residual outlier (solid circle). (c) Linear regression with a low-leverage point that is a residual outlier (solid circle). Least-squares fits are shown as solid lines.

Figure 10.1 uses data simulated to illustrate some of the problems that can arise in regression. There are 11 observations. The predictor variable takes on values  $1, \dots, 10$  and  $50$ , and  $Y = 1 + X + \epsilon$ , where  $\epsilon \sim N(0, 1)$ . The last observation is clearly an extreme value in  $X$ . Such a point is said to have *high leverage*. However, a high-leverage point is not necessarily a problem, only a potential problem. In panel (a), the data have been recorded correctly so that  $Y$  is linearly related to  $X$  and the extreme  $X$ -value is, in fact, helpful as it increases the precision of the estimated slope. In panel (b), the value of  $Y$  for the high-leverage point has been misrecorded as  $5.254$  rather than  $50.254$ . This data point is called a *residual outlier*. As can be seen by comparing the least-squares lines in (a) and (b), the high-leverage point has an extreme influence on the estimated slope. In panel (c),  $X$  has been misrecorded for the high-leverage point as  $5.5$  instead of  $50$ . Thus, this point is no longer high-leverage, but now it is a residual outlier. Its effect now is to bias the estimated intercept.

One should also look at the residuals after the model has been fit, because the residuals may indicate problems not visible in plots of the raw data. However, there are several types of residuals and, as explained soon, one type, called the *externally studentized residual* or *rstudent*, is best for diagnosing problems. Ordinary (or raw) residuals are not necessarily useful for diagnosing problems. For example, in Fig. 10.1b, none of the raw residuals is large, not even the one associated with the residual outlier. The problem is that the raw residuals are too sensitive to the outliers, particularly at high-leverage points, and problems can remain hidden when raw residuals are plotted.  $\square$

Three important tools will be discussed for diagnosing problems with the model or the data:

- leverages;
- externally studentized residuals; and
- Cook's Distance (Cook's D), which quantifies the overall influence of each observation on the fitted values.

### 10.1.1 Leverages

The *leverage* of the  $i$ th observation, denoted by  $H_{ii}$ , measures how much influence  $Y_i$  has on its own fitted value  $\hat{Y}_i$ . We will not go into the algebraic details until Sect. 11.1. An important result in that section is that there are weights  $H_{ij}$  depending on the values of the predictor variables but *not* on  $Y_1, \dots, Y_n$  such that

$$\hat{Y}_i = \sum_{j=1}^n H_{ij} Y_j. \quad (10.1)$$

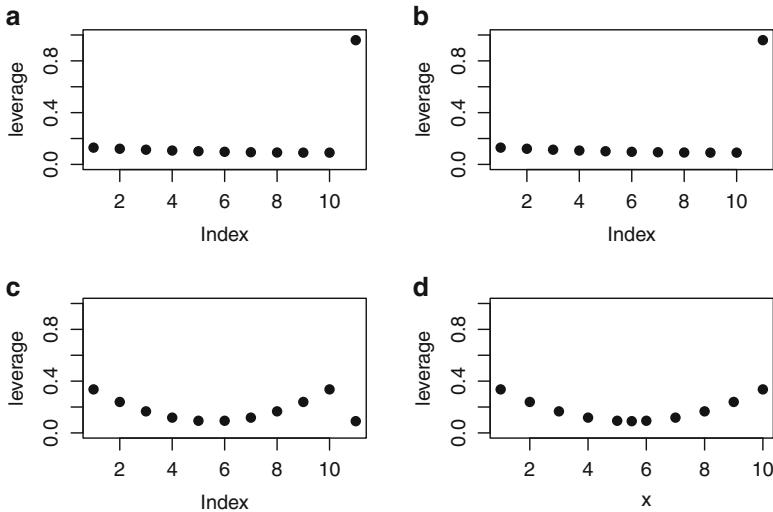
In particular,  $H_{ii}$  is the weight of  $Y_i$  in the determination of  $\hat{Y}_i$ . It is a potential problem if  $H_{ii}$  is large since then  $\hat{Y}_i$  is determined too much by  $Y_i$  itself and not enough by the other data. The result is that the residual  $\hat{\epsilon}_i = Y_i - \hat{Y}_i$  will be small and not a good estimate of  $\epsilon_i$ . Also, the standard error of  $\hat{Y}_i$  is  $\sigma_\epsilon \sqrt{H_{ii}}$ , so a high value of  $H_{ii}$  means a fitted value with low accuracy.

The leverage value  $H_{ii}$  is large when the predictor variables for the  $i$ th case are atypical of those values in the data, for example, because one of the predictor variables for that case is extremely outlying. It can be shown by some elegant algebra that the average of  $H_{11}, \dots, H_{nn}$  is  $(p + 1)/n$ , where  $p + 1$  is the number of parameters (one intercept and  $p$  slopes) and that therefore  $0 < H_{ii} < 1$ . A value of  $H_{ii}$  exceeding  $2(p + 1)/n$ , that is, over twice the average value, is generally considered to be too large and therefore a cause for concern Belsley et al. (1980).

The square matrix with  $i, j$ th element equal to  $H_{ij}$  is called the hat matrix since by (10.1) it converts  $Y_j$ ,  $j = 1, \dots, n$ , to  $\hat{Y}_i$ . The  $H_{ii}$  are sometimes called the *hat diagonals*.

#### *Example 10.2. Leverages in Example 10.1*

Figure 10.2 plots the leverages for the three cases in Fig. 10.1. Because the leverages depend only on the  $X$ -values, the leverages are the same in panels (a) and (b). In both panels, the high-leverage point has a leverage equal to 0.960. In these examples, the rule-of-thumb cutoff point for high leverage is only  $2(p + 1)/n = 2 * 2/11 = 0.364$ , so 0.960 is a huge leverage and close to the maximum possible value of 1. In panel (c), none of the leverages is greater than 0.364.



**Fig. 10.2.** (a)–(c) Leverages plotted again case number (index) for the data sets in Fig. 10.1. Panels (a) and (b) are identical because leverages do not depend on the response values. Panel (d) plots the leverages in (c) against  $X_i$ .

In the special case  $p = 1$ , there is a simple formula for the leverages:

$$H_{ii} = \frac{1}{n} + \frac{(X_i - \bar{X})^2}{\sum_{i=1}^n (X_i - \bar{X})^2}, \quad (10.2)$$

It is easy to check that in this case,  $H_{11} + \cdots + H_{nn} = p + 1 = 2$ , so the average of the hat diagonals is, indeed,  $(p + 1)/n$ . Formula (10.2) shows that  $H_{ii} \geq 1/n$ ,  $H_{ii}$  is equal  $1/n$  if and only if  $X_i = \bar{X}$ , and  $H_{ii}$  increases quadratically with the distance between  $X_i$  and  $\bar{X}$ . This behavior can be seen in Fig. 10.2d.  $\square$

### 10.1.2 Residuals

The *raw residual* is  $\hat{\epsilon}_i = Y_i - \hat{Y}_i$ . Under ideal circumstances such as a reasonably large sample and no outliers or high-leverage points, the raw residuals are approximately  $N(0, \sigma_\epsilon^2)$ , so absolute values greater than  $2\hat{\sigma}_\epsilon^2$  are outlying and greater than  $3\hat{\sigma}_\epsilon^2$  are extremely outlying. However, circumstances are often not ideal. When residual outliers occur at high-leverage points, they can so distort the least-squares fit that they are not seen to be outlying. The problem in these cases is that  $\hat{\epsilon}_i$  is not close to  $\epsilon_i$  because of the bias in the least-squares fit. The bias is due to residual outliers themselves. This problem can be seen in Fig. 10.1b.

The standard error of  $\hat{\epsilon}_i$  is  $\hat{\sigma}_\epsilon \sqrt{1 - H_{ii}}$ , so the raw residuals do not have a constant variance, and those raw residuals with large leverages close to 1

are much less variable than the others. To fix the problem of nonconstant variance, one can use the *standardized residual*, sometimes called the *internally studentized residual*,<sup>1</sup> which is  $\hat{\epsilon}_i$  divided by its standard error, that is,  $\hat{\epsilon}_i / (\hat{\sigma}_\epsilon \sqrt{1 - H_{ii}})$ .

There is still another problem with standardized residuals. An extreme residual outlier can inflate  $\hat{\sigma}_\epsilon$ , causing the standardized residual for the outlying point to appear too small. The solution is to redefine the  $i$ th studentized residual with an estimate of  $\sigma_\epsilon$  that does not use the  $i$ th data point. Thus, the *externally studentized residual*, often called *rstudent*, is defined to be  $\hat{\epsilon}_i / \{\hat{\sigma}_{\epsilon,(-i)} \sqrt{1 - H_{ii}}\}$ , where  $\hat{\sigma}_{\epsilon,(-i)}$  is the estimate of  $\sigma_\epsilon$  computed by fitting the model to the data with the  $i$ th observation deleted.<sup>2</sup> For diagnostics, *rstudent* is considered the best type of residual to plot and is the type of residual used in this book.

**Warning:** The terms “standardized residual” and “studentized residual” do not have the same definitions in all textbooks and software packages. The definitions used here agree with R’s `influence.measures()` function. Other software, such as, SAS uses different definitions.

*Example 10.3. Externally studentized and raw residuals in Example 10.1*

The top row of Fig. 10.3 shows the externally studentized residuals in each of the three cases of simulated data in Fig. 10.1. Case #11 is correctly identified as a residual outlier in data sets (b) and (c) and also correctly identified in data set (a) as not being a residual outlier. The bottom row of Fig. 10.3 shows the raw residuals, rather than the externally studentized residuals. It is not apparent from the raw residuals that in data set (b), case #11 is a residual outlier. This shows the inappropriateness of raw residuals for the detection of outliers, especially when there are high-leverage points.  $\square$

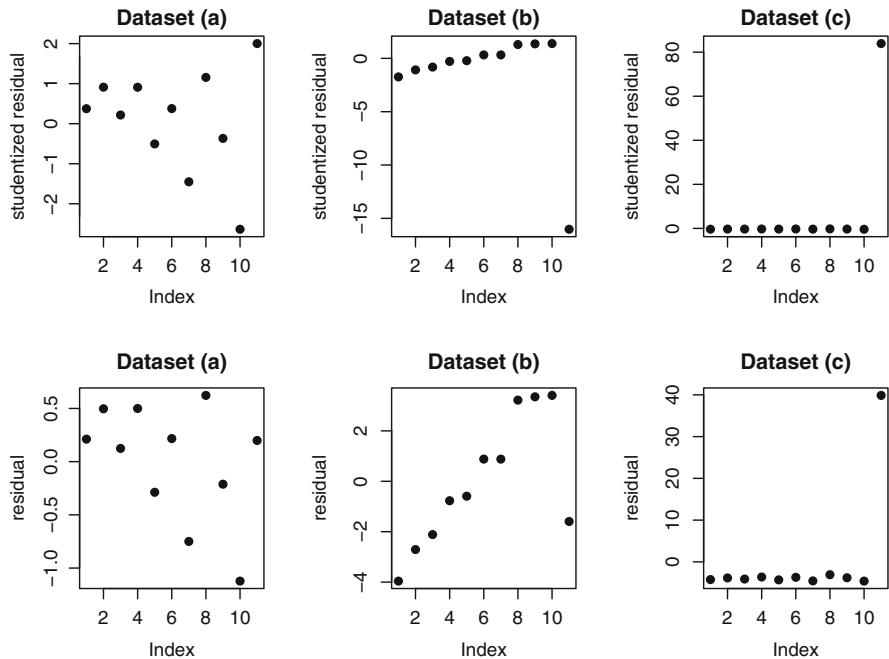
### 10.1.3 Cook’s Distance

A high-leverage value or a large absolute externally studentized residual indicates only a *potential* problem with a data point. Neither tells how much influence the data point actually has on the estimates. For that information, we can use *Cook’s distance*, often called *Cook’s D*, which measures how much the fitted values change if the  $i$ th observation is deleted. We say that Cook’s D measures influence, and any case with a large Cook’s D is called a high-influence case. Leverage and *rstudent* alone do not measure influence.

Let  $\hat{Y}_j(-i)$  be the  $j$ th fitted value using estimates of the  $\hat{\beta}$ s obtained with the  $i$ th observation deleted. Then Cook’s D for the  $i$ th observation is

<sup>1</sup> *Studentization* means dividing a statistic by its standard error.

<sup>2</sup> The notation  $(-i)$  signifies the deletion of the  $i$ th observation.



**Fig. 10.3.** Top row: Externally studentized residuals for the data sets in Fig. 10.1; data set (a) is the data set in panel (a) of Fig. 10.1, and so forth. Case #11 is an outlier in data sets (b) and (c) but not in data set (a). Bottom row: Raw residuals for the same three data sets as in the top row. For data set (b), the raw residual does not reveal that case #11 is outlying.

$$\frac{\sum_{j=1}^n \{\hat{Y}_j - \hat{Y}_j(-i)\}^2}{(p+1)s^2}. \quad (10.3)$$

The numerator in (10.3) is the sum of squared changes in the fitted values when the  $i$ th observation is deleted. The denominator standardizes this sum by dividing by the number of estimated parameters and an estimate of  $\sigma_\epsilon^2$ .

One way to use Cook's D is to plot the values of Cook's D against case number and look for unusually large values. However, it can be difficult to decide which, if any, values of Cook's D are outlying. Of course, some Cook's D values will be larger than others, but are any so large as to be worrisome? To answer this question, a half-normal plot of values of Cook's D, or perhaps of their square roots, can be useful. Neither Cook's D nor its square root is normally distributed, so one does not check for linearity. Instead, one looks for values that are "detached" from the rest.

*Example 10.4.* Cook's D for simulated data in Example 10.1

The three columns of Fig. 10.4 show the values of square roots of Cook's D for the three simulated data examples in Fig. 10.1. In the top row, the square roots of Cook's D values are plotted versus case number (index). The bottom row contains half-normal plots of the square roots of the Cook's D values. In all panels, case #11 has the largest Cook's D, indicating that one should examine this case to see if there is a problem. In data set (a), case #11 is a high-leverage point and has high influence despite not being a residual outlier. In data set (b), where case #11 is both a high-leverage point and a residual outlier, the value of Cook's D for this case is very large, larger than in data set (a). In data set(c), where case #11 has low leverage, all 11 Cook's D values are reasonably small, at least in comparison with data sets (a) and (b), but case #11 is still somewhat outlying.  $\square$

*Example 10.5. Weekly interest data with missing values recorded as zeros*

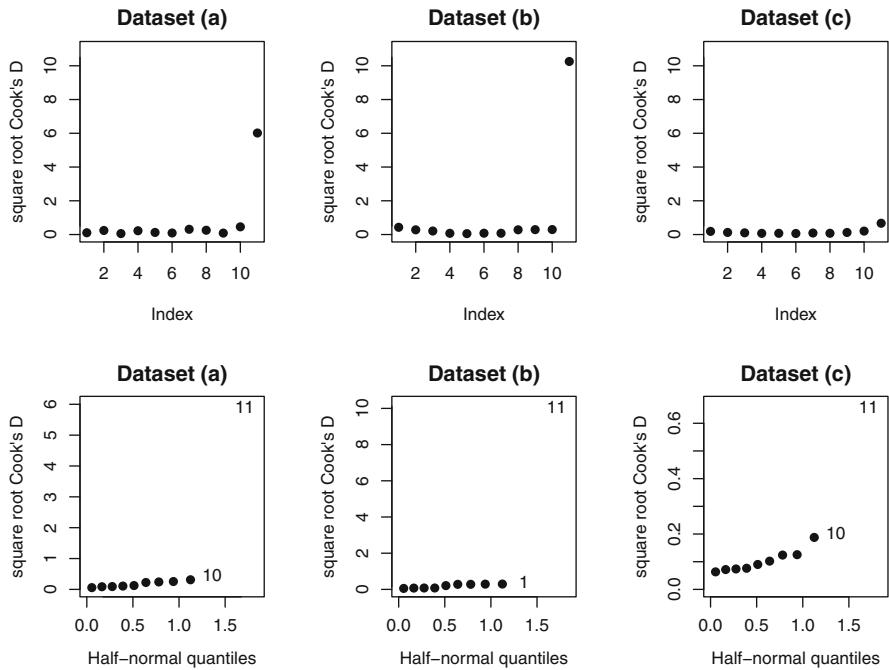
It was mentioned earlier that there were missing values of `cm30` at the beginning of the data set that were coded as zeros. In fact, there were 371 weeks of missing data for `cm30`. I started to analyze the data without realizing this problem. This created a huge outlying value of `cm30_dif` (the first differences) at observation number 372 when `cm30` jumps from 0 to the first nonmissing value. Fortunately, plots of `rstudent`, leverages, and Cook's D all reveal a serious problem somewhere between the 300th and 400th observations, and by zooming into this range of case numbers the problem was located in case #372; see Fig. 10.5. The nature of the problem is not evident from these plots, only its existence, so I plotted each of the series `aaa`, `cm10`, and `cm30`. After seeing the initial zero values of the latter series, the problem was obvious. Please remember this lesson: *ALWAYS look at the data*. Another lesson is that it is best to use nonnumeric values for missing values. For example, R uses “NA” for “not available.”  $\square$

## 10.2 Checking Model Assumptions

Because the  $i$ th residual  $\hat{\epsilon}_i$  estimates the “noise”  $\epsilon_i$ , the residuals can be used to check the assumptions behind regression. Residual analysis generally consists of various plots of the residuals, each plot being designed to check one or more of the regression assumptions. Regression software will output the several types of residuals discussed in Sect. 10.1.2. Externally studentized residuals (`rstudent`) are recommended, for reasons given in that section.

Problems to look for include

1. nonnormality of the errors,
2. nonconstant variance of the errors,



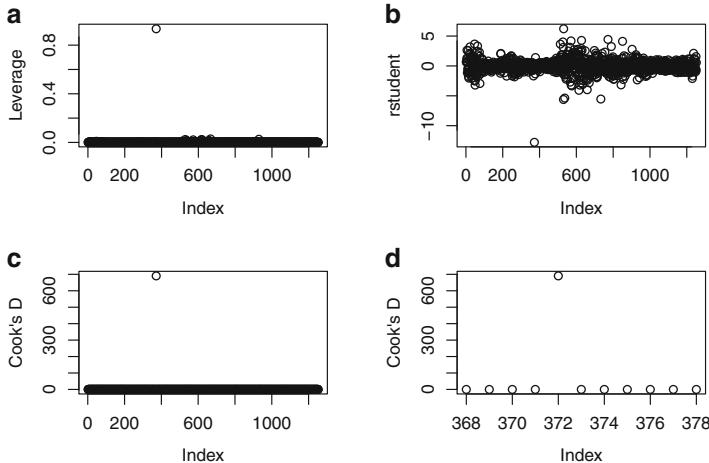
**Fig. 10.4.** Top row: Square roots of Cook's  $D$  for the simulated data plotted against case number. Bottom row: Half-normal plots of square roots of Cook's  $D$ . **Data set (a)** Case #11 has high leverage. It is not a residual outlier but has high influence nonetheless. **Data set (b)** Case #11 has high leverage and is a residual outlier. It has higher influence (as measured by Cook's  $D$ ) than in data set (a). **Data set (c)** Case #11 has low leverage but is a residual outlier. It has much lower influence than in data sets (a) and (b). **Note:** In the top row, the vertical scale is kept constant to emphasize differences among the three cases.

3. nonlinearity of the effects of the predictor variables on the response, and
4. correlation of the errors.

The first three problems are discussed below; correlation of the errors is discussed later in Sect. 13.3.

### 10.2.1 Nonnormality

Nonnormality of the errors (noise) can be detected by a normal probability plot, boxplot, and histogram of the residuals. Not all three are needed, but looking at a normal plot is highly recommended. Moreover, inexperienced data analysts have trouble with the interpretation of normal plots. Looking at side-by-side normal plots and histograms (or KDEs) is helpful when learning to interpret normal probability plots.



**Fig. 10.5.** Weekly interest data. Regression of aaa\_dif on cm10\_dif and cm30\_dif. Full data set including the first 371 weeks of data where cm30 was missing and assigned a value of 0. This caused severe problems at case number 372, which are detected by the leverages in (a),  $r_{student}$  in (b), and Cook's D in (c). Panel (d) zooms in on the outlier case to identify the case number as 372.

The residuals often appear nonnormal because there is an excess of outliers relative to the normal distribution. We have defined a value of  $r_{student}$  to be outlying if its absolute value exceeds 2 and extremely outlying if it exceeds 3. Of course, these cutoffs of 2 and 3 are arbitrary and only intended to give rough guidelines.

It is the presence of outliers, particularly extreme outliers, that is a concern when we have nonnormality. A deficiency of outliers relative to the normal distribution is less of a problem, if it is a problem at all. Sometimes outliers are due to errors, such as mistakes in the entry of the data or, as in Example 10.5, misinterpreting a zero as a true data value rather than the indicator of a missing value. If possible, outliers due to mistakes should be corrected, of course. However, in financial time series, outliers are often “good observations” due, *inter alia*, to excess volatility in the markets on certain days.

Another possible reason for an excess of both positive and negative outlying residuals is nonconstant residual variance, a problem that is explained shortly. Normal probability plots assume that all observations come from the same distribution, in particular, that they have the same variance. The purpose of that plot is to determine if the common distribution is normal or not. If there is no common distribution, for example, because of nonconstant variance, then the normal plot is not readily interpretable. Therefore, one should check for a constant variance before making an extended effort to interpret a normal plot.

Outliers can be a problem because they have an unduly large influence on the estimation results. As discussed in Sect. 4.6, a common solution to the problem of outliers is transformation of the response. Data transformation can be very effective at handling outliers, but it does not work in all situations. Moreover, transformations can induce outliers. For example, if a log transformation is applied to positive data, values very close to 0 could be transformed to outlying negative values since  $\log(x) \rightarrow -\infty$  as  $x \downarrow 0$ .

It is always wise to check whether outliers are due to erroneous data, for example, typing errors or other mistakes in data collection and entry. Of course, erroneous data should be corrected if possible and otherwise removed. Removal of outliers that are not known to be erroneous is dangerous and not recommended as routine statistical practice. However, reanalyzing the data with outliers removed is a sound practice. If the analysis changes drastically when the outliers are deleted, then one knows there is something about which to worry. On the other hand, if deletion of the outliers does not change the conclusions of the analysis, then there is less reason to be concerned with whether the outliers were erroneous data.

A certain amount of nonnormality of the errors is not necessarily a problem. Least-squares estimators are unbiased even without normality. Standard errors for regression coefficients are also correct and confidence intervals are nearly correct because the least-squares estimators obey a central limit theorem—they are nearly normally distributed even if the errors are not normally distributed. Nonetheless, outliers caused by highly skewed or heavy-tailed error distributions can cause the least-squares estimator to be highly variable and therefore inaccurate. Transformations of  $Y$  are commonly used when the errors have skewed distributions, especially when they also have a nonconstant variance. A common solution to heavy-tailed error distributions is robust regression; see Sect. 11.8.

### 10.2.2 Nonconstant Variance

Nonconstant residual variance means that the conditional variance of the response given the predictor variables is not constant as assumed by standard regression models. Nonconstant variance is also called *heteroskedasticity*. Nonconstant variance can be detected by an absolute residual plot, that is, by plotting the absolute residuals against the predicted values ( $\hat{Y}_i$ ) and, perhaps, also against the predictor variables. If the absolute residuals show a systematic trend, then this is an indication of nonconstant variance. Economic data often have the property that larger responses are more variable. A more technical way of stating this is that the conditional variance of the response (given the predictor variables) is an increasing function of the conditional mean of the response. This type of behavior can be detected by plotting the absolute residuals versus the predicted values and looking for an increasing trend.

Often, trends are difficult to detect just by looking at the plotted points and adding a so-called scatterplot smoother is very helpful. A *scatterplot*

*smoother* fits a smooth curve to a scatterplot. Nonparametric regression estimators such as loess and smoothing splines are commonly used scatterplot smoothers available in statistical software packages. These are discussed more fully in Chap. 21.

A potentially serious problem caused by nonconstant variance is inefficiency, that is, too-variable estimates, if ordinary (that is, unweighted) least squares is used. Weighted least squares estimates  $\beta$  efficiently by minimizing

$$\sum_{i=1}^n w_i \{Y_i - f(\mathbf{X}_i; \hat{\beta})\}^2. \quad (10.4)$$

Here  $w_i$  an estimate of the inverse (that is, reciprocal) conditional variance of  $Y_i$  given  $\mathbf{X}_i$ , so that the more variable observations are given less weight. Estimation of the conditional variance function to determine the  $w_i$ s is discussed in the more advanced textbooks mentioned in Sect. 10.3. Weighted least-squares for regression with GARCH errors is discussed in Sect. 14.12.

Another serious problem caused by heteroskedasticity is that standard errors and confidence intervals assume a constant variance and can be seriously wrong if there is substantial nonconstant variance.

Transformation of the response is a common solution to the problem of nonconstant variance; see Sect. 11.4. If the response can be transformed to constant variance, then unweighted least-squares will be efficient and standard errors and confidence intervals will be valid.

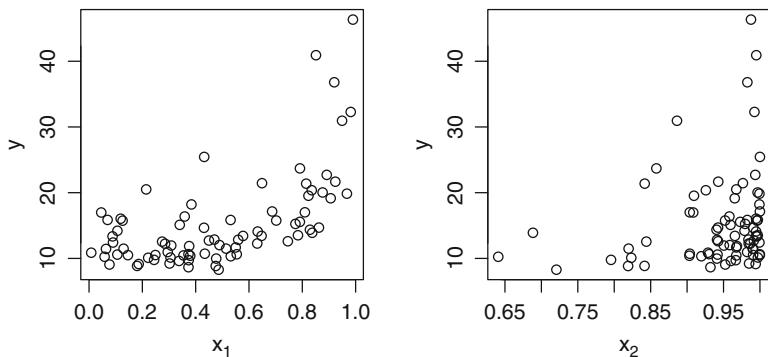
### 10.2.3 Nonlinearity

If a plot of the residuals versus a predictor variable shows a systematic nonlinear trend, then this is an indication that the effect of that predictor on the response is nonlinear. Nonlinearity causes biased estimates and a model that may predict poorly. Confidence intervals, which assume unbiasedness, can be seriously in error if there is nonlinearity. The value  $100(1 - \alpha)\%$  is called the *nominal value* of the coverage probability of a confidence interval and is guaranteed to be the actual coverage probability only if all modeling assumptions are met.

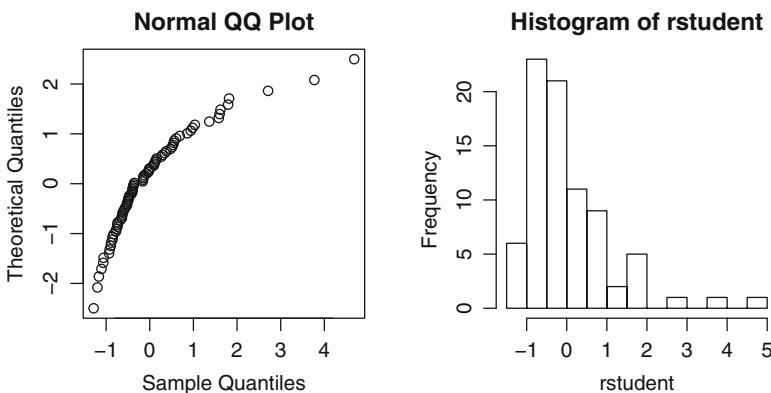
Response transformation, polynomial regression, and nonparametric regression (e.g., splines and loess—see Chap. 21) are common solutions to the problem of nonlinearity.

*Example 10.6. Detecting nonlinearity: A simulated data example*

Data were simulated to illustrate some of the techniques for diagnosing problems. In the example there are two predictor variables,  $X_1$  and  $X_2$ . The assumed model is multiple linear regression,  $Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \epsilon_i$ .



**Fig. 10.6.** Simulated data. Responses plotted against the two predictor variables.



**Fig. 10.7.** Simulated data. Normal plot and histogram of the studentized residuals. Right skewness is evident and perhaps a square root or log transformation of  $Y$  would be helpful.

Figure 10.6, which shows the responses plotted against each of the predictors, suggests that the errors are heteroskedastic because there is more vertical scatter on the right sides of the plots. Otherwise, it is not clear whether there are other problems with the data or the model. The point here is that plots of the raw data often fail to reveal all problems. Rather, it is plots of the residuals that can more reliably detect heteroskedasticity, nonnormality, and other difficulties.

Figure 10.7 contains a normal plot and a histogram of the residuals—the externally standardized residuals ( $r_{student}$ ) are used in all examples of this chapter. Notice the right skewness which suggests that a response transformation to remove right skewness, such as, a square-root or log transformation, should be investigated.

Figure 10.8a is a plot of the residuals versus  $X_1$ . The residuals appear to have a nonlinear trend. This is better revealed by adding a loess curve to the residuals. The curvature of the loess fit is evident and indicates that  $Y$  is not linear in  $X_1$ . A possible remedy is to add  $X_1^2$  as a third predictor. Figure 10.8a, a plot of the residuals against  $X_2$ , shows somewhat random scatter, indicating that  $Y$  appears to be linear in  $X_2$ . The concentration of the  $X_2$ -values near the right side is not a problem. This pattern only shows that the distribution of  $X_2$  is left-skewed, but the regression model makes no assumptions about the distributions of the predictors.

Before doing any more plotting, the model was augmented by adding  $X_1^2$  as a predictor, so the model is now

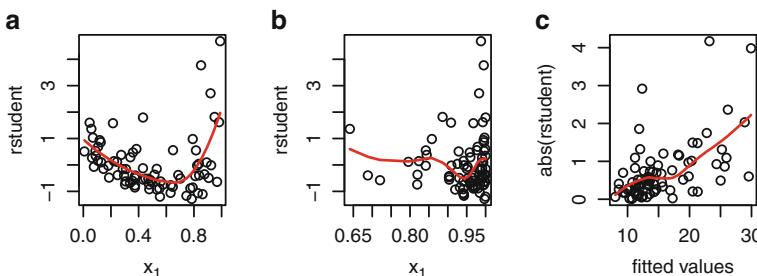
$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2}^2 + \beta_3 X_{i,2} + \epsilon_i. \quad (10.5)$$

Figure 10.8c is a plot of the absolute residuals versus the predicted values for model (10.5). Note that the absolute residuals are largest where the fitted values are also largest, which is a clear sign of heteroskedasticity. A loess smooth has been added to make the heteroskedasticity clearer.

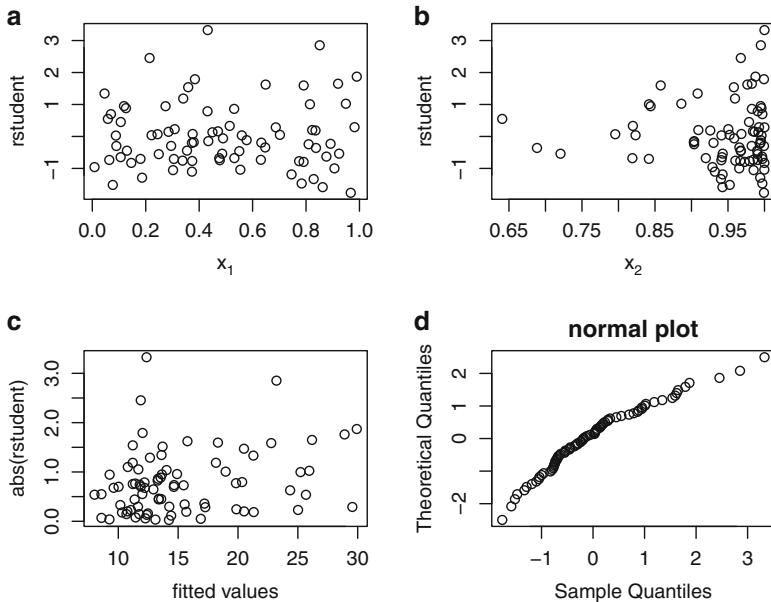
To remedy the problem of heteroskedasticity,  $Y_i$  was transformed to  $\log(Y_i)$ , so the model is now

$$\log(Y_i) = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2}^2 + \beta_3 X_{i,2} + \epsilon_i. \quad (10.6)$$

Figure 10.9 shows residual plots for model (10.6). The plots in panels (a) and (b) of residuals versus  $X_1$  and  $X_2$  show no patterns, indicating that the



**Fig. 10.8.** Simulated data. (a) Plot of externally studentized residuals versus  $X_1$ . This plot suggests that  $Y$  is not linearly related to  $X_1$  and perhaps a model quadratic in  $X_1$  is needed. (b) Plot of the residuals versus  $X_2$  with a loess smooth. This plot suggests that  $Y$  is linearly related to  $X_2$  so that the component of the model relating  $Y$  to  $X_2$  is satisfactory. (c) Plot of the absolute residuals versus the predicted values using a model that is quadratic in  $X_1$ . This plot reveals heteroskedasticity. A loess smooth has been added to each plot.



**Fig. 10.9.** Simulated data. Residual plots for fit of  $\log(Y)$  to  $X_1$ ,  $X_1^2$ , and  $X_2$ . (a) Residuals versus  $X_1$ . (b) Residuals versus  $X_2$ . (c) Residuals versus  $\hat{Y}$ .

model that is quadratic in  $X_1$  fits well. The plot in panel (c) of absolute residuals versus fitted values shows less heteroskedasticity than before, which shows the benefit of the log transformation. The normal plot of the residuals shown in panel (d) shows much less skewness than earlier, which is another benefit of the log transformation.  $\square$

### 10.3 Bibliographic Notes

Graphical methods for detecting nonconstant variance, transform-both-sides regression, and weighting are discussed in Carroll and Ruppert (1988). The idea of using half-normal plots to detect usual values of Cook's D was borrowed from Faraway (2005).

Comprehensive treatments of regression diagnostics can be found in Belsley et al. (1980) and in Cook and Weisberg (1982). Although variance inflation factors detect collinearity, they do not indicate what correlations are causing the problem. For this purpose, one should use collinearity diagnostics. These are also discussed in Belsley et al. (1980).

## 10.4 R Lab

### 10.4.1 Current Population Survey Data

This section uses the `CPS1988` data set from the March 1988 Current Population Survey by the U.S. Census Bureau and available in the `AER` package. These are cross-sectional data, meaning that the U.S. population was surveyed at a single time point. Cross-sectional data should be distinguished from longitudinal data where individuals are followed over time. Data collected and analyzed along two dimensions, that is, cross-sectionally and longitudinally, are called *panel data* by econometricians.

In this section, we will investigate how the variable `wage` (in dollars/week) depends on `education` (in years), `experience` (years of potential work experience), and `ethnicity` (Caucasian = “caus” or African-American = “afam”). Potential experience was (`age - education - 6`), the number of years of potential work experience assuming that education begins at age 6. Potential experience was used as a proxy for actual work experience, which was not available. The variable `ethnicity` is coded 0–1 for “cauc” and “afam,” so its regression coefficient is the difference in the expected values of `wage` between an African-American and a Caucasian with the same values of `education` and `experience`. Run the code below to load the data and run a multiple linear regression.

```
library(AER)
data(CPS1988)
attach(CPS1988)
fitLm1 = lm(wage ~ education + experience + ethnicity)
```

Next, create residual plots with the following code. In some of these plots, the  $y$ -axis limits are set so as to eliminate outliers. This was done to focus attention on the bulk of the data. This is a very large data set with 28,155 observations, so scatterplots are very dense with data and almost solid black in places. Therefore, lowess smooths were added as thick, red lines so that they can be seen clearly. Also, thick blue reference lines were added as appropriate.

```
par(mfrow = c(3, 2))
resid1 = rstudent(fitLm1)
plot(fitLm1$fit, resid1,
 ylim = c(-1500, 1500), main = "(a)")
lines(lowess(fitLm1$fit, resid1, f = 0.2), lwd = 5, col = "red")
abline(h = 0, col = "blue", lwd = 5)

plot(fitLm1$fit, abs(resid1),
 ylim = c(0, 1500), main = "(b)")
lines(lowess(fitLm1$fit, abs(resid1), f = 0.2),
 lwd = 5, col = "red")
abline(h = mean(abs(resid1)), col = "blue", lwd = 5)
```

```

qqnorm(resid1, datax = FALSE, main = "(c)")
qqline(resid1, datax = FALSE, lwd = 5, col = "blue")

plot(education, resid1, ylim = c(-1000, 1500), main = "(d)")
lines(lowess(education, resid1), lwd = 5, col = "red")
abline(h = 0, col = "blue", lwd = 5)

plot(experience, resid1, ylim = c(-1000, 1500), main = "(e)")
lines(lowess(experience, resid1), lwd = 5, col = "red")
abline(h = 0, col = "blue", lwd = 5)

```

**Problem 1** For each of the panels (a)–(e) in the figure you have just created, describe what is being plotted and any conclusions that should be drawn from the plot. Describe any problems and discuss how they might be remedied.

**Problem 2** Now fit a new model where the log of wage is regressed on education and experience. Create residual plots as done above for the first model. Describe differences between the residual plots for the two models. What do you suggest should be tried next?

**Problem 3** Implement whatever you suggested to try next in Problem 2. Describe how well it worked. Are you satisfied with your model? If not, try further enhancements of the model until arriving at a model that you feel is satisfactory. What is your final model?

**Problem 4** Use your final model to describe the effects of education, experience, and ethnicity on the wage. Use graphs where appropriate.

Check the data and your final model for possible problems or unusual features by examining the hat diagonals and Cook's D with the following code. Replace `fitLm4` by the name of the `lm` object for your final model.

```

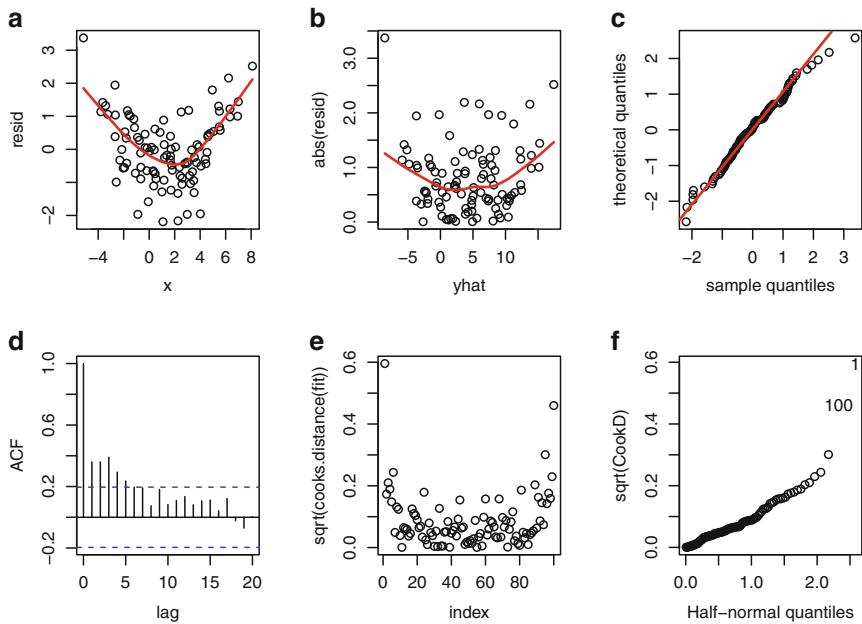
library(faraway) # required for halfnorm
par(mfrow=c(1, 3))
plot(hatvalues(fitLm4))
plot(sqrt(cooks.distance(fitLm4)))
halfnorm(sqrt(cooks.distance(fitLm4)))

```

**Problem 5** Do you see any high-leverage points or points with very high values of Cook's D? If you do, what is unusual about them?

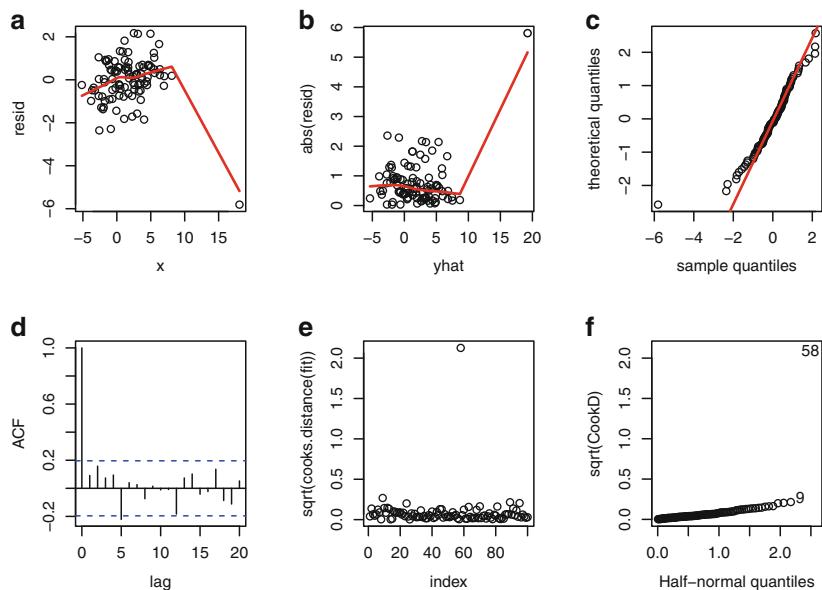
## 10.5 Exercises

1. Residual plots and other diagnostics are shown in Fig. 10.10 for a regression of  $Y$  on  $X$ . Describe any problems that you see and possible remedies.



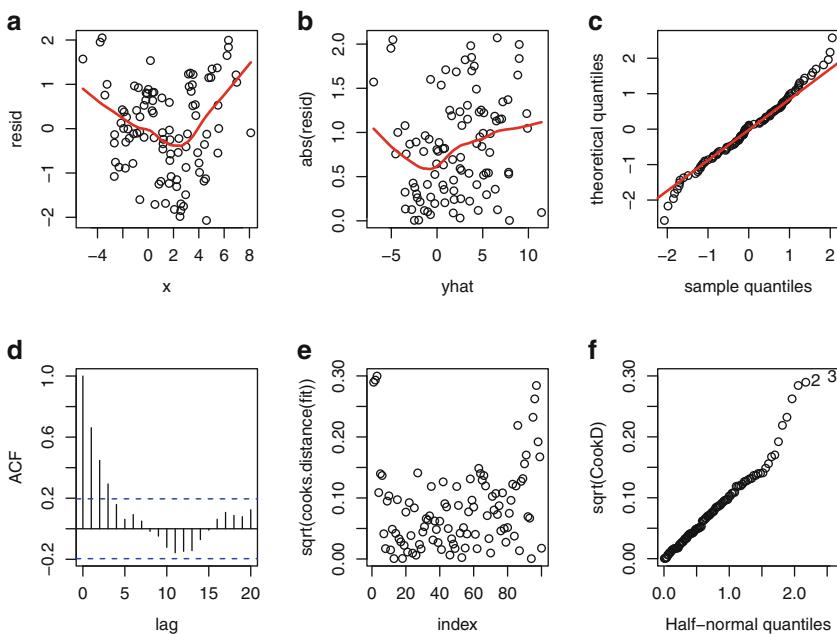
**Fig. 10.10.** Residual plots and diagnostics for regression of  $Y$  on  $X$  in Problem 1. The residuals are `rstudent` values. (a) Plot of residuals versus  $x$ . (b) Plot of absolute residuals versus fitted values. (c) Normal QQ plot of residuals. (d) ACF plot of residuals. (e) Plot of the square root of Cook's  $D$  versus index (= observation number). (f) Half-normal plot of square root of Cook's  $D$ .

2. Residual plots and other diagnostics are shown in Fig. 10.11 for a regression of  $Y$  on  $X$ . Describe any problems that you see and possible remedies.



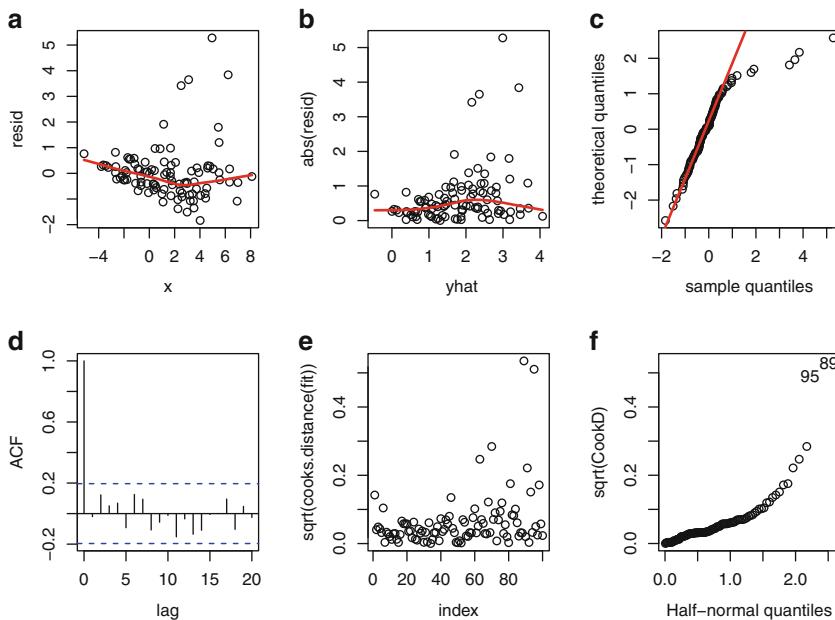
**Fig. 10.11.** Residual plots and diagnostics for regression of  $Y$  on  $X$  in Problem 2. The residuals are `rstudent` values. (a) Plot of residual versus  $x$ . (b) Plot of absolute residuals versus fitted values. (c) Normal Q-Q plot of residuals. (d) ACF plot of residuals. (e) Plot of the square root of Cook's  $D$  versus index (= observation number). (f) Half-normal plot of square root of Cook's  $D$ .

3. Residual plots and other diagnostics are shown in Fig. 10.12 for a regression of  $Y$  on  $X$ . Describe any problems that you see and possible remedies.



**Fig. 10.12.** Residual plots and diagnostics for regression of  $Y$  on  $X$  in Problem 3. The residuals are  $rstudent$  values. (a) Plot of residual versus  $x$ . (b) Plot of absolute residuals versus fitted values. (c) Normal QQ plot of residuals. (d) ACF plot of residuals. (e) Plot of the square root of Cook's  $D$  versus index (= observation number). (f) Half-normal plot of square root of Cook's  $D$ .

4. Residual plots and other diagnostics are shown in Fig. 10.13 for a regression of  $Y$  on  $X$ . Describe any problems that you see and possible remedies.



**Fig. 10.13.** Residual plots and diagnostics for regression of  $Y$  on  $X$  in Problem 4. The residuals are `rstudent` values. (a) Plot of residual versus  $x$ . (b) Plot of absolute residuals versus fitted values. (c) Normal QQ plot of residuals. (d) ACF plot of residuals. (e) Plot of the square root of Cook's  $D$  versus index (= observation number). (f) Half-normal plot of square root of Cook's  $D$ .

5. It was noticed that a certain observation had a large leverage (hat diagonal) but a small Cook's  $D$ . How could this happen?

## References

- Belsley, D. A., Kuh, E., and Welsch, R. E. (1980) *Regression Diagnostics*, Wiley, New York.  
 Carroll, R. J., and Ruppert, D. (1988) *Transformation and Weighting in Regression*, Chapman & Hall, New York.  
 Cook, R. D., and Weisberg, S. (1982) *Residuals and Influence in Regression*, Chapman & Hall, New York.  
 Faraway, J. J. (2005) *Linear Models with R*, Chapman & Hall, Boca Raton, FL.

---

## Regression: Advanced Topics

### 11.1 The Theory Behind Linear Regression

This section provides some theoretical results about linear least-squares estimation. The study of linear regression is facilitated by the use of matrices. Equation (9.1) can be written more succinctly as

$$Y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, n \quad (11.1)$$

where  $\mathbf{x}_i = (1 \ X_{i,1} \ \dots \ X_{i,p})^\top$  and  $\boldsymbol{\beta} = (\beta_0 \ \beta_1 \ \dots \ \beta_p)^\top$ . Let

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}, \text{ and } \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

Then, the  $n$  equations in (11.1) can be expressed as

$$\underbrace{\mathbf{Y}}_{n \times 1} = \underbrace{\mathbf{X}}_{n \times (p+1)} \underbrace{\boldsymbol{\beta}}_{(p+1) \times 1} + \underbrace{\boldsymbol{\epsilon}}_{n \times 1}, \quad (11.2)$$

with the matrix dimensions indicated by underbraces.

The least-squares estimate of  $\boldsymbol{\beta}$  minimizes

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{Y}^\top \mathbf{Y} - 2\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{Y} + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}. \quad (11.3)$$

By setting the derivatives of (11.3) with respect to  $\beta_0, \dots, \beta_p$  equal to 0 and simplifying the resulting equations, one finds that the least-squares estimator is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}. \quad (11.4)$$

Using (7.9), one can find the covariance matrix of  $\hat{\beta}$ :

$$\begin{aligned}\text{COV}(\hat{\beta} | \mathbf{x}_1, \dots, \mathbf{x}_n) &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \text{COV}(\mathbf{Y} | \mathbf{x}_1, \dots, \mathbf{x}_n) \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \\ &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\sigma_\epsilon^2 \mathbf{I}) \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \\ &= \sigma_\epsilon^2 (\mathbf{X}^\top \mathbf{X})^{-1},\end{aligned}$$

since  $\text{COV}(\mathbf{Y} | \mathbf{x}_1, \dots, \mathbf{x}_n) = \text{COV}(\boldsymbol{\epsilon}) = \sigma_\epsilon^2 \mathbf{I}$ , where  $\mathbf{I}$  is the  $n \times n$  identity matrix. Therefore, the standard error of  $\hat{\beta}_j$  is the square root of the  $j$ th diagonal element of  $\sigma_\epsilon^2 (\mathbf{X}^\top \mathbf{X})^{-1}$ .

The vector of fitted values is

$$\hat{\mathbf{Y}} = \mathbf{X} \hat{\beta} = \{\mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top\} \mathbf{Y} = \mathbf{H} \mathbf{Y},$$

where  $\mathbf{H} = \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$  is the *hat matrix*. The leverage of the  $i$ th observation is  $H_{ii}$ , the  $i$ th diagonal element of  $\mathbf{H}$ .

### 11.1.1 Maximum Likelihood Estimation for Regression

In this section, we assume a linear regression model with noise that may not be normally distributed and independent.

For example, consider the special case of i.i.d. errors. It is useful to put the scale parameter explicitly into the regression model, so we assume that

$$Y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \sigma \epsilon_i,$$

where  $\{\epsilon_i\}$  are i.i.d. with a known density  $f$  that has variance equal to 1 and  $\sigma$  is the unknown noise standard deviation. For example,  $f$  could be a standardized  $t$ -density. Then the likelihood of  $Y_1, \dots, Y_n$  is

$$\prod_{i=1}^n \frac{1}{\sigma} f\left\{\frac{Y_i - \mathbf{x}_i^\top \boldsymbol{\beta}}{\sigma}\right\}.$$

The maximum likelihood estimator maximizes the log-likelihood

$$L(\boldsymbol{\beta}, \sigma) = -n \log(\sigma) + \sum_{i=1}^n \log \left[ f\left\{\frac{Y_i - \mathbf{x}_i^\top \boldsymbol{\beta}}{\sigma}\right\} \right].$$

For normally distributed errors,  $\log\{f(x)\} = -\frac{1}{2}x^2 - \frac{1}{2}\log(2\pi)$ , and for the purpose of maximization, the constant  $-\frac{1}{2}\log(2\pi)$  can be ignored. Therefore, the log-likelihood is

$$L^{\text{GAUSS}}(\boldsymbol{\beta}, \sigma) = -n \log(\sigma) - \frac{1}{2} \sum_{i=1}^n \left( \frac{Y_i - \mathbf{x}_i^\top \boldsymbol{\beta}}{\sigma} \right)^2.$$

It should be obvious that the least-squares estimator is the MLE of  $\boldsymbol{\beta}$ . Also, maximizing  $L^{\text{GAUSS}}(\hat{\boldsymbol{\beta}}, \sigma)$  in  $\sigma$ , where  $\boldsymbol{\beta}$  has been replaced by the least-squares estimate, is a standard calculus exercise and the result is

$$\hat{\sigma}_{\text{MLE}}^2 = n^{-1} \sum_{i=1}^n (Y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}})^2.$$

It can be shown that  $\hat{\sigma}_{\text{MLE}}^2$  is biased but that the bias is eliminated if  $n^{-1}$  is replaced by  $\{n - (p + 1)\}^{-1}$  where  $p + 1$  is the dimension of  $\boldsymbol{\beta}$ . This give us the estimator (9.16).

Now assume that  $\boldsymbol{\epsilon}$  has a covariance matrix  $\boldsymbol{\Sigma}$  and, for some function  $f$ , density

$$|\boldsymbol{\Sigma}|^{-1/2} f\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\}.$$

Then the log-likelihood is

$$-\frac{1}{2} \log |\boldsymbol{\Sigma}| + \log [f\{(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\}] .$$

In the important special case where  $\boldsymbol{\epsilon}$  has a mean-zero multivariate normal distribution, the density of  $\boldsymbol{\epsilon}$  is

$$\left[ \frac{1}{|\boldsymbol{\Sigma}|^{1/2} (2\pi)^{p/2}} \right] \exp \left\{ -\frac{1}{2} \boldsymbol{\epsilon}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} \right\}, \quad (11.5)$$

If  $\boldsymbol{\Sigma}$  is known, then the MLE of  $\boldsymbol{\beta}$  minimizes

$$(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

and is called the *generalized least-squares estimator* (GLS estimator). If  $\epsilon_1, \dots, \epsilon_n$  are uncorrelated but with possibly different variances, then  $\boldsymbol{\Sigma}$  is the diagonal matrix of these variances and the generalized least-squares estimator is the weighted least-squares estimator (10.4).

The GLS estimator is

$$\hat{\boldsymbol{\beta}}_{\text{GLS}} = (\mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \mathbf{Y}. \quad (11.6)$$

Typically,  $\boldsymbol{\Sigma}$  is unknown and must be replaced by an estimate, for example, from an ARMA model for the errors.

## 11.2 Nonlinear Regression

Often we can derive a theoretical model relating predictor variables and a response, but the model we derive is not linear. In particular, models derived from economic theory are commonly used in finance and many are not linear.

The nonlinear regression model is

$$Y_i = f(\mathbf{X}_i; \boldsymbol{\beta}) + \epsilon_i, \quad (11.7)$$

where  $Y_i$  is the response measured on the  $i$ th observation,  $\mathbf{X}_i$  is a vector of observed predictor variables for the  $i$ th observation,  $f(\cdot; \cdot)$  is a *known*

function,  $\beta$  is an unknown parameter vector, and  $\epsilon_1, \dots, \epsilon_n$  are i.i.d. with mean 0 and variance  $\sigma_\epsilon^2$ . The least-squares estimate  $\hat{\beta}$  minimizes

$$\sum_{i=1}^n \{Y_i - f(\mathbf{X}_i; \beta)\}^2.$$

The predicted values are  $\hat{Y}_i = f(\mathbf{X}_i; \hat{\beta})$  and the residuals are  $\hat{\epsilon}_i = Y_i - \hat{Y}_i$ .

Since the model is nonlinear, finding the least-squares estimate requires nonlinear optimization. Because of the importance of nonlinear regression, almost every statistical software package will have routines for nonlinear least-squares estimation. This means that most of the difficult programming has already been done for us. However, we do need to write an equation that specifies the model we are using.<sup>1</sup> In contrast, when using linear regression only the predictor variables need to be specified.

### *Example 11.1. Simulated bond prices*

Consider prices of par \$1000 zero-coupon bonds issued by a particular borrower, perhaps the Federal government or a corporation. Suppose that there are several times to maturity, the  $i$ th being denoted by  $T_i$ . Suppose also that the yield to maturity is a constant, say  $r$ . The assumption that  $Y_T = r$  for all  $T$  is not realistic and is used only to keep this example simple. In Sect. 11.3 more realistic models will be used.

The rate  $r$  is determined by the market and can be estimated from prices. Under the assumption of a constant value of  $r$ , the present price of a bond with maturity  $T_i$  is

$$P_i = 1000 \exp(-rT_i). \quad (11.8)$$

There is some random variation in the observed prices. One reason is that the price of a bond can only be determined by the sale of the bond, so the observed prices have not been determined simultaneously. Prices that may no longer reflect current market values are called *stale*. Each bond's price was determined at the time of the last trade of a bond of that maturity, and  $r$  may have had a somewhat different value then. It is only as a function of time to maturity that  $r$  is assumed constant, so  $r$  may vary with calendar time. Thus, we augment model (11.8) by including a noise term to obtain the regression model

$$P_i = 1000 \exp(-rT_i) + \epsilon_i. \quad (11.9)$$

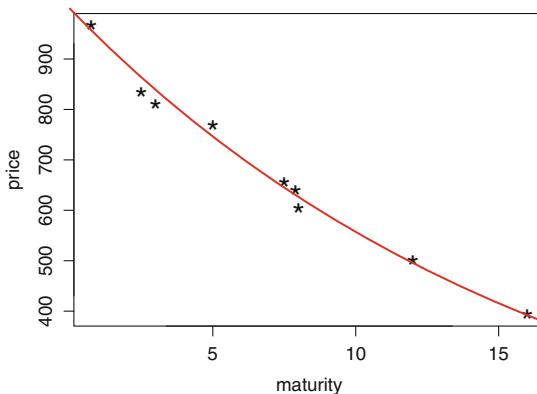
An estimate of  $r$  can be determined by least squares, that is, by minimizing over  $r$  the sum of squares:

$$\sum_{i=1}^n \left\{ P_i - 1,000 \exp(-rT_i) \right\}^2.$$

---

<sup>1</sup> Even this work can sometimes be avoided, since some nonlinear regression software has many standard models already programmed.

The least-squares estimator is denoted by  $\hat{r}$ .



**Fig. 11.1.** Plot of bond prices against maturities with the predicted price from the nonlinear least-squares fit.

Since it is unlikely that market data will have a constant  $r$ , this example uses simulated data. The data were generated with  $r$  fixed at 0.06 and plotted in Fig. 11.1. The nonlinear least-squares estimate of  $r$  was found using R's `nls()` function. Nonlinear optimization requires starting values for the parameters, and a starting value of 0.04 was used for  $r$ .

```
bondprices = read.table("bondprices.txt", header = TRUE)
attach(bondprices)
fit = nls(price ~ 1000 * exp(-r * maturity), start = list(r = 0.04))
summary(fit)
detach(bondprices)
```

The output is:

```
Formula: price ~ 1000 * exp(-r * maturity)

Parameters:
Estimate Std. Error t value Pr(>|t|)
r 0.05850 0.00149 39.3 1.9e-10 ***

Residual standard error: 20 on 8 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 5.53e-08
```

Notice that  $\hat{r} = 0.0585$  and the standard error of this estimate is 0.00149. The predicted price curve using nonlinear regression is shown in Fig. 11.1.  $\square$

As mentioned, in *nonlinear regression* the form of the regression function is nonlinear but *known* up to a few unknown parameters. For example, the regression function has an exponential form in model (11.9). For this reason, nonlinear regression would best be called *nonlinear parametric regression* to distinguish it from nonparametric regression, where the regression function is also nonlinear but not of a known parametric form. Nonparametric regression is discussed in Chap. 21.

Polynomial regression may appear to be nonlinear since polynomials are nonlinear functions. For example, the quadratic regression model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i \quad (11.10)$$

is nonlinear in  $X_i$ . However, by defining  $X_i^2$  as a second predictor variable, this model is linear in  $(X_i, X_i^2)$  and therefore is an example of multiple *linear* regression. What makes model (11.10) linear is that the right-hand side is a linear function of the parameters  $\beta_0$ ,  $\beta_1$ , and  $\beta_2$ , and therefore can be interpreted as a linear regression with the appropriate definition of the variables. In contrast, the exponential model

$$Y_i = \beta_0 e^{\beta_1 X_i} + \epsilon_i$$

is nonlinear in the parameter  $\beta_1$ , so it cannot be made into a linear model by redefining the predictor variable.

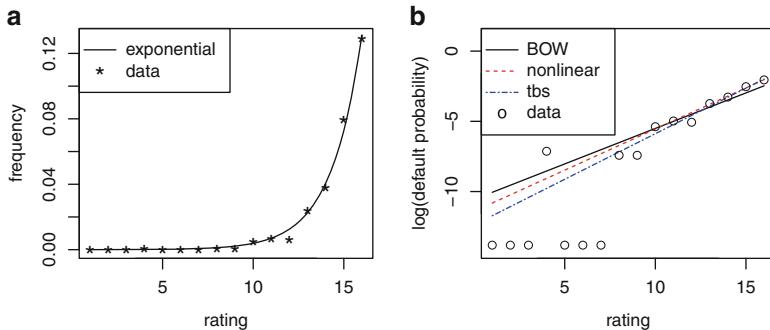
### *Example 11.2. Estimating default probabilities*

This example illustrates both nonlinear regression and the detection of heteroskedasticity by residual plotting.

Credit risk is the risk to a lender that a borrower will default on contractual obligations, for example, that a loan will not be repaid in full. A key parameter in the determination of credit risk is the probability of default. Bluhm, Overbeck, and Wagner (2003) illustrate how one can calibrate Moody's credit rating to estimate default probabilities. These authors use observed default frequencies for bonds in each of 16 Moody's ratings from Aaa (best credit rating) to B3 (worse rating). They convert the credit ratings to a 1 to 16 scale (Aaa = 1, ..., B3 = 16). Figure 11.2a shows default frequencies (as fractions, not percentages) plotted against the ratings. The data are from Bluhm, Overbeck, and Wagner (2003). The relationship is clearly nonlinear. Not surprisingly, Bluhm, Overbeck, and Wagner used a nonlinear model, specifically

$$\Pr\{\text{default} | \text{rating}\} = \exp\{\beta_0 + \beta_1 \text{rating}\}. \quad (11.11)$$

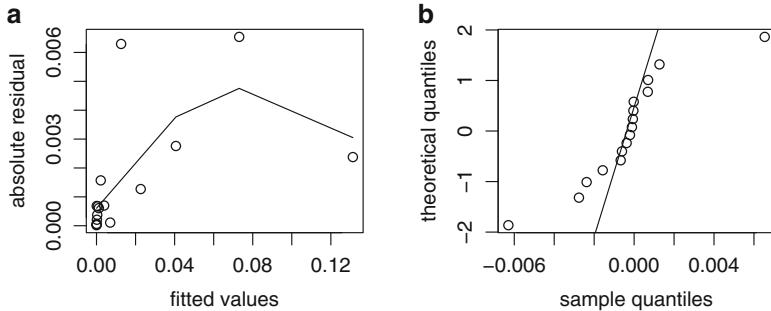
To use this model they fit a linear function to the logarithms of the default frequencies. One difficulty with doing this is that six of the default frequencies are zero giving a log transformation of  $-\infty$ .



**Fig. 11.2.** (a) Default frequencies with an exponential fit. “Rating” is a conversion of the Moody’s rating to a 1 to 16-point scale as follows: 1 = Aaa, 2 = Aa1, 3 = Aa3, 4 = A1, . . . , 16 = B3. (b) Estimation of default probabilities by Bluhm, Overbeck, and Wagner’s (2003) linear regression with ratings removed that have no observed defaults (BOW) and by nonlinear regression with all data (nonlinear). Because some default frequencies are zero, when plotting the data on a semilog plot,  $10^{-6}$  was added to the default frequencies. This constant was not added when estimating default frequencies, only for plotting the raw data. The six observations along the bottom of the plot are the ones removed by Bluhm, Overbeck, and Wagner. “TBS” is the transform-both-sides estimate, which will be discussed soon.

Bluhm, Overbeck, and Wagner (2003) address this issue by labeling default frequencies equal to zero as “unobserved” and not using them in the estimation process. The problem with their technique is that they have deleted the data with the lowest observed default frequencies. This biases their estimates of default probabilities in an upward direction. Bluhm, Overbeck, and Wagner argue that an observed default frequency of zero does not imply that the true default probability is zero. This is certainly true. However, the default frequencies, even when they are zero, are unbiased estimates of the true default probabilities. There is no intent here to be critical of their book, which is well-written and useful. However, one can avoid the bias of their method by using nonlinear regression with model (11.11). The advantage of fitting (11.11) by nonlinear regression is that it avoids the use of a logarithm transformation thus allowing the use of all the data, even data with a default frequency of zero. The fits by the Bluhm, Overbeck, and Wagner method and by nonlinear regression using model (11.11) are shown in Fig. 11.2b with a log scale on the vertical axis so that the fitted functions are linear. Notice that at good credit ratings the estimated default probabilities are lower using nonlinear regression compared to Bluhm, Overbeck, and Wagner’s biased method. The differences between the two sets of estimated default probabilities can be substantial. Bluhm,

Overbeck, and Wagner estimate the default probability of an Aaa bond as 0.005 %. In contrast, the unbiased estimate by nonlinear regression is only 40 % of that figure, specifically, 0.0020 %. Thus, the bias in the Bluhm, Overbeck, and Wagner estimate leads to a substantial overestimate of the credit risk of Aaa bonds and similar overestimation at other good credit ratings.



**Fig. 11.3.** (a) Residuals for estimation of default probabilities by nonlinear regression. Absolute studentized residuals plotted against fitted values with a loess smooth. Substantial heteroskedasticity is indicated because the data on the left side are less scattered than elsewhere. (b) Normal probability plot of the residuals. Notice the outliers caused by the nonconstant variance.

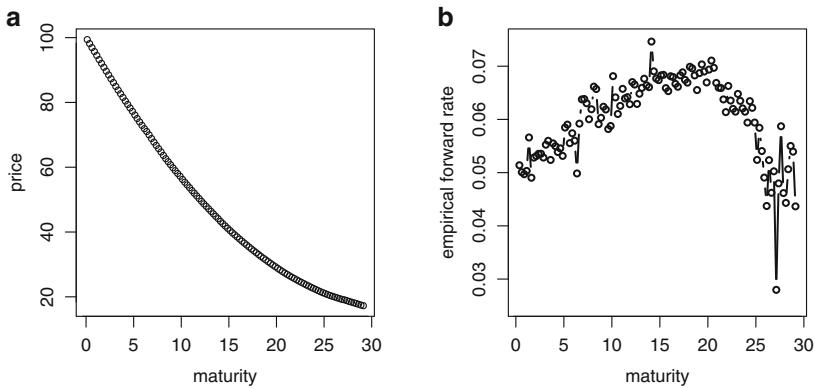
A plot of the absolute residuals versus the fitted values in Fig. 11.3a gives a clear indication of heteroskedasticity. Heteroskedasticity does not cause bias but it does cause inefficient estimates. In Sect. 11.4, this problem is fixed by a variance-stabilizing transformation. Figure 11.3b is a normal probability plot of the residuals. Outliers with both negative and positive values can be seen. These are due to the nonconstant variance and are not necessarily a sign of nonnormality. This plot illustrates the danger of attempting to interpret a normal plot when the data have a nonconstant variance. One should apply a variance-stabilizing transformation first before checking for normality.  $\square$

### 11.3 Estimating Forward Rates from Zero-Coupon Bond Prices

In practice, the forward-rate function  $r(t)$  is unknown. Only bond prices are known. If the prices  $P(T_i)$  of zero-coupon bonds are available on a relatively fine grid of values of  $T_1 < T_2 < \dots < T_n$ , then using (3.24) we can estimate the forward-rate curve at  $T_i$  with

$$-\frac{\Delta \log\{P(T_i)\}}{\Delta T_i} = -\frac{\log\{P(T_i)\} - \log\{P(T_{i-1})\}}{T_i - T_{i-1}}. \quad (11.12)$$

We will call these the *empirical forward-rate estimates*. Figure 11.4 shows prices and empirical forward-rate estimates from data to be described soon in Example 11.3. As can be seen in the plot, the empirical forward-rate estimates can be rather noisy when the denominators in (11.12) are small because the maturities are spaced closely together. If the maturities were more widely spaced, then bias rather than variance would be the major problem. Despite these difficulties, the empirical forward-rate estimates give a general impression of the forward-rate curve and are useful for comparing with estimates from parametric models, which are discussed next.



**Fig. 11.4.** (a) U.S. STRIPS prices. (b) Empirical forward-rate estimates from the prices.

We can estimate  $r(t)$  from the bond prices using nonlinear regression. An example of estimating  $r(t)$  was given in Sect. 11.2 assuming that  $r(t)$  was constant and using as data the prices of zero-coupon bonds of different maturities. In this section, we estimate  $r(t)$  without assuming it is constant.

Parametric estimation of the forward-rate curves starts with a parametric family  $r(t; \boldsymbol{\theta})$  of forward rates and the corresponding yield curves

$$y_T(\boldsymbol{\theta}) = T^{-1} \int_0^T r(t; \boldsymbol{\theta}) dt$$

and model for the price of a par-\$1 bond:

$$P_T(\boldsymbol{\theta}) = \exp\{-Ty_T(\boldsymbol{\theta})\} = \exp\left(-\int_0^T r(t; \boldsymbol{\theta}) dt\right).$$

For example, suppose that  $r(t; \boldsymbol{\theta})$  is a  $p$ th-degree polynomial, so that

$$r(t; \boldsymbol{\theta}) = \theta_0 + \theta_1 t + \cdots + \theta_p t^p$$

for some unknown parameters  $\theta_0, \dots, \theta_p$ . Then

$$\int_0^T r(t; \boldsymbol{\theta}) dt = \theta_0 T + \theta_1 \frac{T^2}{2} + \cdots + \theta_p \frac{T^{p+1}}{p},$$

and the yield curve is

$$y_T = T^{-1} \int_0^T r(t; \boldsymbol{\theta}) dt = \theta_0 + \theta_1 \frac{T}{2} + \cdots + \theta_p \frac{T^p}{p}.$$

A popular model is the Nelson–Siegel family with forward-rate and yield curves

$$\begin{aligned} r(t; \boldsymbol{\theta}) &= \theta_0 + (\theta_1 + \theta_2 t) \exp(-\theta_3 t), \\ y_t(\boldsymbol{\theta}) &= \theta_0 + \left( \theta_1 + \frac{\theta_2}{\theta_3} \right) \frac{1 - \exp(-\theta_3 t)}{\theta_3 t} - \frac{\theta_2}{\theta_3} \exp(-\theta_3 t). \end{aligned}$$

The six-parameter Svensson model extends the Nelson–Siegel model by adding the term  $\theta_4 t \exp(-\theta_5 t)$  to the forward rate.

The nonlinear regression model for estimating the forward-rate curve states that the price of the  $i$ th bond in the sample with maturity  $T_i$  expressed as a fraction of par value is

$$P_i = D(T_i) + \epsilon_i = \exp \left( - \int_0^{T_i} r(t; \boldsymbol{\theta}) dt \right) + \epsilon_i, \quad (11.13)$$

where  $D$  is the discount function and  $\epsilon_i$  is an “error” due to problems such as prices being somewhat stale and the bid–ask spread.<sup>2</sup>

### *Example 11.3. Estimating forward rates from STRIPS prices*

We now look at an example using data on U.S. STRIPS, a type of zero-coupon bond. STRIPS is an acronym for “Separate Trading of Registered Interest and Principal of Securities.” The interest and principal on Treasury bills, notes, and bonds are traded separately through the Federal Reserve’s book-entry system, in effect creating zero-coupon bonds by repackaging coupon bonds.<sup>3</sup>

The data are from December 31, 1995. The prices are given as a percentage of par value. Price is plotted against maturity in years in Fig. 11.4a. There are 117 prices and the maturities are nearly equally spaced from 0 to 30 years. We can see that the price drops smoothly with maturity and that there is not much noise in the price data. The empirical forward-rate estimates in Fig. 11.4b are much noisier than the prices.

<sup>2</sup> A bond dealer buys bonds at the bid price and sells them at the ask price, which is slightly higher than the bid price. The difference is called the bid–ask spread and covers the trader’s administrative costs and profit.

<sup>3</sup> Jarrow (2002, p. 15).

Three models for the forward curve were fit: quadratic polynomial, cubic polynomial, and quadratic polynomial spline with a knot at  $T = 15$ . The latter splices two quadratic functions together at  $T = 15$  so that the resulting curve is continuous and with a continuous first derivative. The spline's second derivative jumps at  $T = 15$ . One way to write the spline is

$$r(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 (t - 15)_+^2, \quad (11.14)$$

where the positive-part function is  $x_+ = x$  if  $x \geq 0$  and  $x_+ = 0$  if  $x < 0$ . Also,  $x_+^2$  means  $(x_+)^2$ , that is, take the positive part first. See Chap. 21 for further information about splines. From (11.14), one obtains

$$\int_0^T r(t) dt = \beta_0 T + \beta_1 \frac{T^2}{2} + \beta_2 \frac{T^3}{3} + \beta_3 \frac{(T - 15)_+^3}{3}, \quad (11.15)$$

and therefore the yield curve is

$$y_T = \beta_0 + \beta_1 \frac{T}{2} + \beta_2 \frac{T^2}{3} + \beta_3 \frac{(T - 15)_+^3}{3T}. \quad (11.16)$$

From (11.15), the model for a bond price (as a percentage of par) is

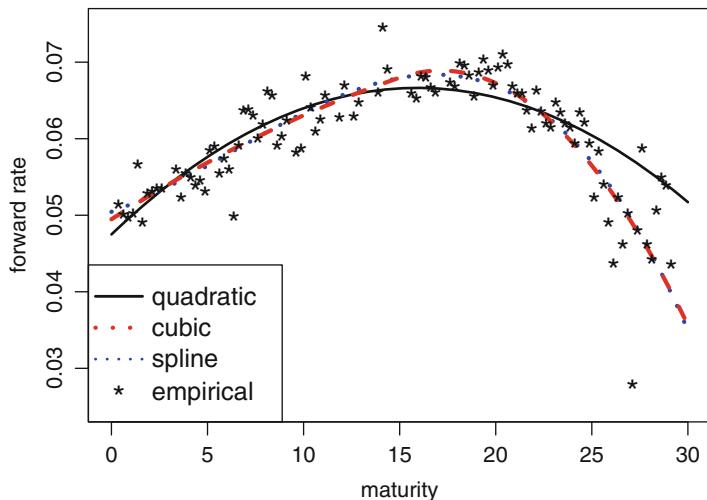
$$100 \exp \left\{ - \left( \beta_0 T + \beta_1 \frac{T^2}{2} + \beta_2 \frac{T^3}{3} + \beta_3 \frac{(T - 15)_+^3}{3} \right) \right\}. \quad (11.17)$$

R code to fit the quadratic spline and plot its forward-rate estimate is

```
fitSpline = nls(price ~ 100 * exp(-beta0 * T
 - (beta1 * T^2)/2 - (beta2 * T^3) / 3
 - (T > 15) * (beta3 * (T - 15)^3) / 3), data = dat,
 start = list(beta0 = 0.03, beta1 = 0, beta2 = 0, beta3 = 0))
coefSpline = summary(fitSpline)$coef[, 1]
forwardSpline = coefSpline[1] + (coefSpline[2] * t) +
 (coefSpline[3] * t^2) + (t > 15) * (coefSpline[4] * (t - 15)^2)
plot(t, forwardSpline, lty = 2, lwd = 2)
```

Only slight changes in the code are needed to fit the quadratic or cubic polynomial models.

Figure 11.5 contains all three estimates of the forward rate and the empirical forward rates. The cubic polynomial and quadratic spline models follow the empirical forward rates much more closely than the quadratic polynomial model. The cubic polynomial and quadratic spline fits both use four parameters and are similar to each other, though the spline has a slightly smaller residual sum of squares. The summary of the spline model's fit is



**Fig. 11.5.** Polynomial and spline estimates of forward rates of U.S. Treasury bonds. The empirical forward rates are also shown.

```
> summary(fitSpline)

Formula: price ~ 100 * exp(-beta0 * T - (beta1 * T^2)/2
- (beta2 * T^3)/3 - (T > 15) * (beta3 * (T - 15)^3)/3)

Parameters:
Estimate Std. Error t value Pr(>|t|)
beta0 4.947e-02 9.221e-05 536.52 <2e-16 ***
beta1 1.605e-03 3.116e-05 51.51 <2e-16 ***
beta2 -2.478e-05 1.820e-06 -13.62 <2e-16 ***
beta3 -1.763e-04 5.755e-06 -30.64 <2e-16 ***

Residual standard error: 0.0667 on 113 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 1.181e-07
```

Notice that all coefficients have very small  $p$ -values. The small  $p$ -value of **beta3** is further evidence that the spline model fits better than the quadratic polynomial model, since the two models differ only in that **beta3** is 0 for the quadratic model.

R's **nls** function could not find the least-squares estimator for the Nelson–Siegel model, but the least-squares estimator was found using the **optim** non-linear optimization function with the sum of squares as the objective function. The fit of the Nelson–Siegel model was noticeably inferior to that of the cubic

polynomial and quadratic spline models. In fact, the Nelson–Siegel model did not fit even as well as the quadratic polynomial model.

The Svensson model is likely to fit better than the Nelson–Siegel model, but the four-parameter cubic polynomial and quadratic spline models fit sufficiently well that it did not seem worthwhile to try the six-parameter Svensson model.  $\square$

## 11.4 Transform-Both-Sides Regression

Suppose we have a theoretical model that states that in the absence of any noise,

$$Y_i = f(\mathbf{X}_i; \boldsymbol{\beta}). \quad (11.18)$$

Model (11.18) is identical to the model

$$h\{Y_i\} = h\{f(\mathbf{X}_i; \boldsymbol{\beta})\}, \quad (11.19)$$

where  $h$  is *any* one-to-one function, such as, a strictly increasing function. In the absence of noise, one choice of  $h$  is as good as any other and one might as well stick with model (11.18), but when noise exists, this is no longer true.

When we have noisy data, Eq. (11.19) can be converted to the nonlinear regression model

$$h\{Y_i\} = h\{f(\mathbf{X}_i; \boldsymbol{\beta})\} + \epsilon_i. \quad (11.20)$$

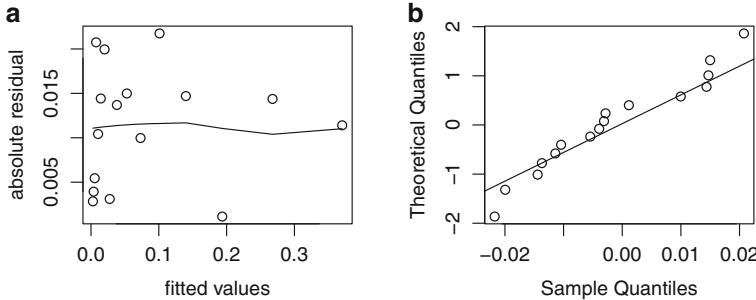
Model (11.20) is called *the transform-both-sides (TBS) regression model* because both sides of Eq. (11.19) have been transformed by the same function  $h$ . Typically,  $h$  will be one of the Box–Cox transformations and  $h$  is chosen to stabilize the variation and to induce nearly normally distributed errors. To estimate  $\boldsymbol{\beta}$  for a fixed  $h$ , one minimizes

$$\sum_{i=1}^n \left[ h\{Y_i\} - h\{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\} \right]^2. \quad (11.21)$$

Various choices of  $h$  can be compared by residual plots. The  $h$  that gives approximately normally distributed residuals with a constant variance is used for the final analysis.

### *Example 11.4. TBS regression for the default frequency data*

TBS regression was applied to the default frequency data. The Box–Cox transformation  $h(y) = y^{(\alpha)}$  was tried with various positive values of  $\alpha$ . It was found that  $\alpha = 1/2$  gave residuals that appeared normally distributed with a constant variance, so the square-root transformation was used for estimation; see Fig. 11.6. With this transformation,  $\boldsymbol{\beta}$  is estimated by minimizing



**Fig. 11.6.** (a) Transform-both-sides regression (TBS) with  $h(y) = \sqrt{y}$ . Absolute studentized residuals plotted against fitted values with a loess smooth. (b) Normal plot of the studentized residuals.

$$\sum_{i=1}^n \left[ \sqrt{Y_i} - \exp\{\beta_0/2 + (\beta_1/2)X_i\} \right]^2, \quad (11.22)$$

where  $Y_i$  is the  $i$ th default frequency and  $X_i$  is the  $i$ th rating. The square-root transformation of the model is accomplished by dividing  $\beta_0$  and  $\beta_1$  by 2.

The R code to fit the TBS model and create Fig. 11.6 is below. The fitted values `fit_tbs` are computed by subtracting the residuals from the responses; this is done because the function `summary()` does not return the fitted values.

```
DefaultData = read.table("DefaultData.txt", header = TRUE)
attach(DefaultData)
freq2 = freq / 100
fit_tbs = nls(sqrt(freq2) ~ exp(b1 / 2 + b2 * rating / 2),
 start = list(b1 = -6, b2 = 0.5))
sum_tbs = summary(fit_tbs)
par(mfrow = c(1, 2))
fitted_tbs = sqrt(freq2) - sum_tbs$resid
plot(fitted_tbs, abs(sum_tbs$resid), xlab = "fitted values",
 ylab = "absolute residual")
fit_loess_tbs = loess(abs(sum_tbs$resid) ~ fitted_tbs,
 span = 1, deg = 1)
ord_tbs = order(fitted_tbs)
lines(fitted_tbs[ord_tbs], fit_loess_tbs$fit[ord_tbs])
qqnorm(sum_tbs$resid, datax = TRUE, main = "")
qqline(sum_tbs$resid, datax = TRUE)
detach(DefaultData)
```

Using TBS regression, the estimated default probability of Aaa bonds is 0.0008 %, only 16 % of the estimate given by Bluhm, Overbeck, and Wagner (2003) and only 40 % of the estimate given by nonlinear regression without a transformation. Of course, a reduction in estimated risk by 84 % is a huge change. This shows how proper statistical modeling—e.g., using all the data and an appropriate transformation—can have a major impact on financial risk

analysis. TBS allows one to use all the data (for unbiasedness) and, as described next, to effectively weight the data by the reciprocals of their variances for high efficiency.

□

### 11.4.1 How TBS Works

TBS in effect weights the data. To appreciate this, we use a Taylor series linearization<sup>4</sup> to obtain

$$\sum_{i=1}^n \left[ h(Y_i) - h\left\{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\right\} \right]^2 = \sum_{i=1}^n \left[ h^{(1)}\left\{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\right\} \right]^2 \left\{ Y_i - f(\mathbf{X}_i; \hat{\boldsymbol{\beta}}) \right\}^2.$$

The weight of the  $i$ th observation is  $\left[ h^{(1)}\{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\} \right]^2$ . Since the best weights are inverse variances, the most appropriate transformation  $h$  solves

$$\text{Var}(Y_i | \mathbf{X}_i) \propto \left[ h^{(1)}\{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\} \right]^{-2}. \quad (11.23)$$

For example, if  $h(y) = \log(y)$ , then  $h^{(1)}(y) = 1/y$  and (11.23) becomes

$$\text{Var}(Y_i | \mathbf{X}_i) \propto \{f(\mathbf{X}_i; \hat{\boldsymbol{\beta}})\}^2, \quad (11.24)$$

so that the conditional standard deviation of the response is proportional to its conditional mean. This occurs frequently. For example, if the response is exponentially distributed then (11.24) must hold. Equation (11.24) holds also if the response is lognormally distributed and the log-variance is constant. In this case, it is not surprising that the log transformation is best since the log transforms to i.i.d. normal noise.

The *coefficient of variation* of a random variable is the ratio of its standard deviation to its expected value. When (11.24) holds, the response has a constant coefficient of variation.

A transformation that causes that conditional variance to be constant is called the *variance-stabilizing transformation*. We have just shown that when the coefficient of variation is constant, then the variance-stabilizing transformation is the logarithm.

#### *Example 11.5. Poisson responses*

Assume  $Y_i | \mathbf{X}_i$  is Poisson distributed with mean  $f(\mathbf{X}_i; \boldsymbol{\beta})$ , as might, for example, happen if  $Y_i$  were of the number of companies declaring bankruptcy

---

<sup>4</sup> A Taylor series linearization of the function  $h$  about the point  $x$  is  $h(y) \approx h(x) + h^{(1)}(x)(y - x)$ , where  $h^{(1)}$  is the first derivative of  $h$ . See any calculus textbook for further discussion of Taylor series.

in a year, with  $f(\mathbf{X}_i; \boldsymbol{\beta})$  modeling how that expected number depends on macroeconomic variables in  $\mathbf{X}_i$ . The variance equals the mean for the Poisson distribution, so

$$\text{Var}(Y_i | \mathbf{X}_i) = f(\mathbf{X}_i; \boldsymbol{\beta}).$$

Using the same type of reasoning as in the previous example, it follows that one should use  $\alpha = 1/2$ ; the square-root transformation is the variance-stabilizing transformation for Poisson-distributed responses.  $\square$

## 11.5 Transforming Only the Response

The so-called Box–Cox transformation model is

$$Y_i^{(\alpha)} = \beta_0 + X_{i,1}\beta_1 + \cdots + X_{i,p}\beta_p + \epsilon_i, \quad (11.25)$$

where  $\epsilon_1, \dots, \epsilon_n$  are i.i.d.  $N(0, \sigma_\epsilon^2)$  for some  $\sigma_\epsilon$ . In contrast to the TBS model, only the response is transformed. The goal of transforming the response is to achieve three objectives:

1. a simple model:  $Y_i^{(\alpha)}$  is linear in predictors  $X_{i,1}, \dots, X_{i,p}$  and in the parameters  $\beta_1, \dots, \beta_p$ ;
2. constant residual variance; and
3. Gaussian noise.

In contrast, 2 and 3 but *not* 1 are the goals of the TBS model.

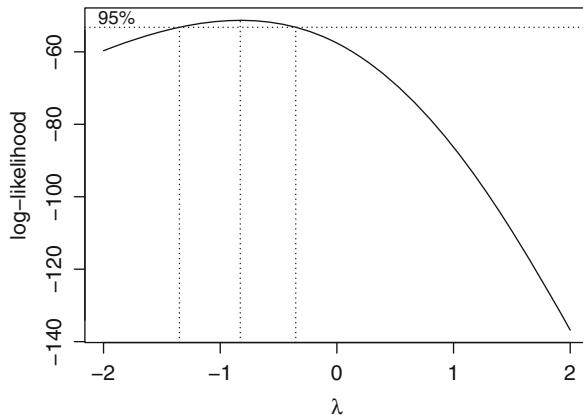
Model (11.25) was introduced by Box and Cox (1964) who suggested estimation of  $\alpha$  by maximum likelihood. The function `boxcox()` in R's MASS package will compute the profile log-likelihood for  $\alpha$  along with a confidence interval. Usually,  $\hat{\alpha}$  is taken to be some round number, e.g.,  $-1, -1/2, 0, 1/2$ , or  $1$ , in the confidence interval. The reason for selecting one of these numbers is that then the transformation is readily interpretable, that is, it is the square root, log, inverse, or some other familiar function. Of course, one can use the value of  $\alpha$  that maximizes the profile log-likelihood if one is not concerned with having a familiar transformation. After  $\hat{\alpha}$  has been selected in this way,  $\beta_0, \dots, \beta_p$  and  $\sigma_\epsilon^2$  can be estimated by regressing  $Y_i^{(\hat{\alpha})}$  on  $X_{i,1}, \dots, X_{i,p}$ .

*Example 11.6. Simulated data—Box Cox transformation*

This example uses the simulated data introduced in Example 10.6. The model is

$$Y_i^{(\alpha)} = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,1}^2 + \beta_3 X_{i,2} + \epsilon_i. \quad (11.26)$$

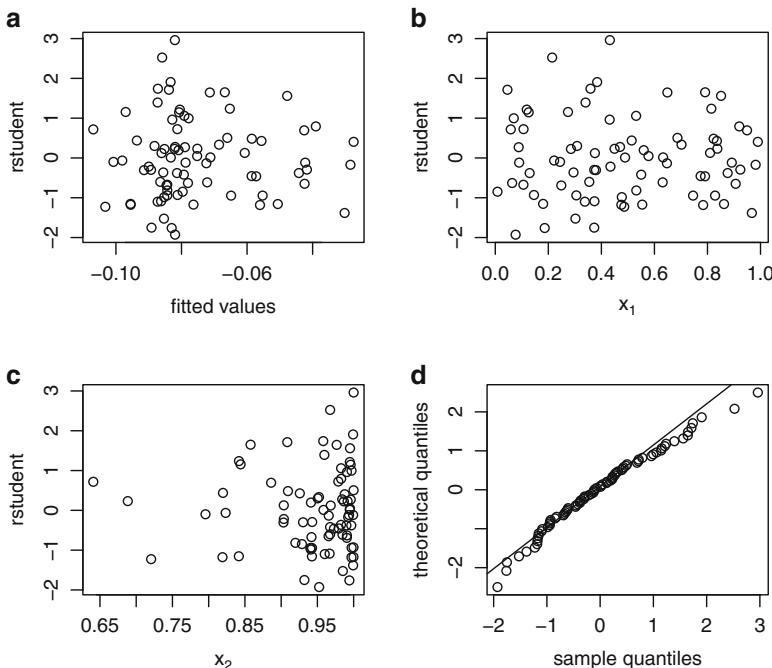
The profile likelihood for  $\alpha$  was produced by the `boxcox()` function in R and is plotted in Fig. 11.7. The code to produce the figure is:



**Fig. 11.7.** Profile likelihood for the Box–Cox model applied to the simulated data.

```
boxcox(y ~ poly(x1,2) + x2, ylab = "log-likelihood")
```

We see that the MLE is near  $-1$  and  $-1$  is well within the confidence interval; these results suggest that we use  $-1/Y_i$  as the response.



**Fig. 11.8.** Residuals for the Box–Cox model applied to the simulated data.

Residual plots with response  $-1/Y_i$  are shown in Fig. 11.8. We see in panel (a) that there is no sign of heteroskedasticity, since the vertical scatter of the residuals does not change from left to right. In panels (b) and (c) we see uniform vertical scatter which shows that the model that is quadratic in  $X_1$  and linear in  $X_2$  fits  $-1/Y_i$  well. Finally, in panel (d), we see that the residuals appear normally distributed.  $\square$

## 11.6 Binary Regression

A binary response  $Y$  can take only two values, 0 or 1, which code two possible outcomes, for example, that a company goes into default on its loans or that it does not default. Binary regression models the conditional probability that a binary response is 1, given the values of the predictors  $X_{i,1}, \dots, X_{i,p}$ . Since a probability is constrained to lie between 0 and 1, a linear model is not appropriate for a binary response. However, linear models are so convenient that one would like a model that has many of the features of a linear model. This has motivated the development of *generalized linear models*, often called GLMs.

Generalized linear models for binary responses are of the form

$$P(Y_i = 1|X_{i,1}, \dots, X_{i,p}) = H(\beta_0 + \beta_1 X_{i,1} + \dots + \beta_p X_{i,p}) = H(\mathbf{x}_i^\top \boldsymbol{\beta}),$$

where  $H(x)$  is a function that increases from 0 to 1 as  $x$  increases from  $-\infty$  to  $\infty$ , so that  $H(x)$  is a CDF, and the last expression uses the vector notation of (11.1). The most common GLMs for a binary responses are probit regression, where  $H(x) = \Phi(x)$ , the  $N(0, 1)$  CDF; and logistic regression, where  $H(x)$  is the logistic CDF, which is  $H(x) = 1/\{1 + \exp(-x)\}$ . The parameter vector  $\boldsymbol{\beta}$  can be estimated by maximum likelihood. Assume that conditional on  $\mathbf{x}_1, \dots, \mathbf{x}_n$  the binary responses  $Y_1, \dots, Y_n$  are mutually independent. Then, using (A.8), the likelihood (conditional on  $\mathbf{x}_1, \dots, \mathbf{x}_n$ ) is

$$\prod_{i=1}^n H(\mathbf{x}_i^\top \boldsymbol{\beta})^{Y_i} \{1 - H(\mathbf{x}_i^\top \boldsymbol{\beta})\}^{1-Y_i}. \quad (11.27)$$

The MLEs can be found by standard software, e.g., the function `glm()` in R.

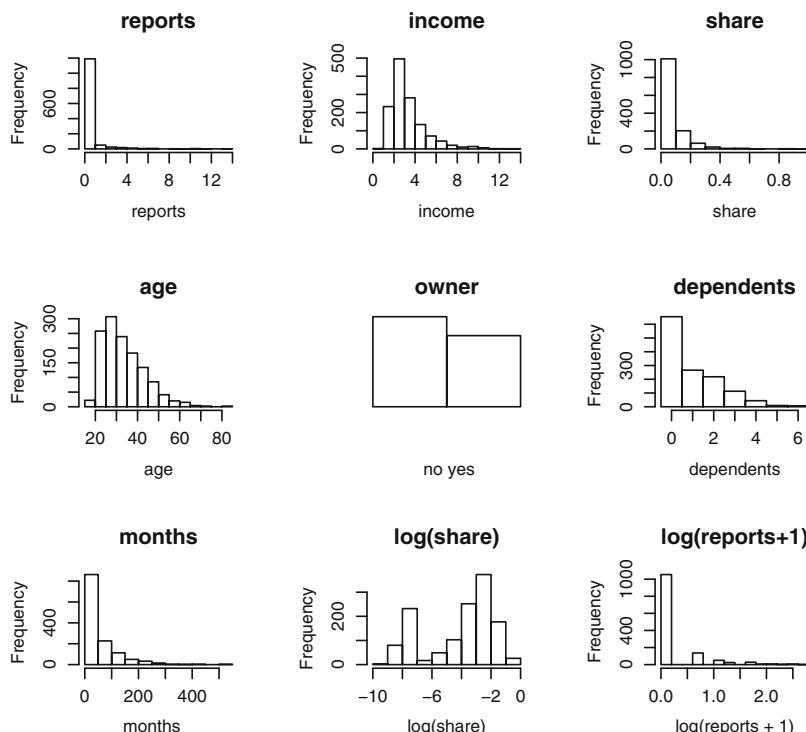
*Example 11.7. Who gets a credit card?*

In this example, we will analyze the data in the `CreditCard` data set in R's `AER` package. The following variables are included in the data set:

1. `card` = Was the application for a credit card accepted?
2. `reports` = Number of major derogatory reports
3. `income` = Yearly income (in USD 10,000)
4. `age` = Age in years plus 12ths of a year

5. **owner** = Does the individual own his or her home?
6. **dependents** = Number of dependents
7. **months** = Months living at current address
8. **share** = Ratio of monthly credit card expenditure to yearly income
9. **selfemp** = Is the individual self-employed?
10. **majorcards** = Number of major credit cards held
11. **active** = Number of active credit accounts
12. **expenditure** = Average monthly credit card expenditure

The first variable, **card**, is binary and will be the response. Variables 2–8 will be used as predictors. The goal of the analysis is to discover which of the predictors influences the probability that an application is accepted. R's documentation mentions that there are some values of the variable **age** under one year. These cases must be in error and they were deleted from the analysis. Figure 11.9 contains histograms of the predictors. The variable **share** is highly right-skewed, so  $\log(\text{share})$  will be used in the analysis. The variable **reports**



**Fig. 11.9.** Histograms of variables for potential use in a model to predict whether a credit card application will be accepted.

is also extremely right-skewed; most values of `reports` are 0 or 1 but the maximum value is 14. To reduce the skewness, `log(reports+1)` will be used instead of `reports`. The “1” is added to avoid taking the logarithm of 0. There are no assumptions in regression about the distributions of the predictors, so skewed predictor variables can, in principle, be used. However, highly skewed predictors have high-leverage points and are less likely to be linearly related to the response. It is a good idea at least to consider transformation of highly skewed predictors. In fact, the logistic model was also fit with `reports` and `share` untransformed, but this increased AIC by more than 3 compared to using the transformed predictors.

First, a logistic regression model is fit with all seven predictors using the `glm()` function. The R code is:

```

library("AER")
library("faraway")
data("CreditCard")
CreditCard_clean = CreditCard[CreditCard$age > 18,]
names(CreditCard)
fit1 = glm(card ~ log(reports + 1) + income + log(share) + age
 + owner + dependents + months,
 family = "binomial", data = CreditCard_clean)
summary(fit1)
stepAIC(fit1)

Call:
glm(formula = card ~ log(reports + 1) + income + log(share) +
 age + owner + dependents + months, family = "binomial",
 data = CreditCard_clean)

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) 21.473930 3.674325 5.844 5.09e-09 ***
log(reports + 1) -2.908644 1.097604 -2.650 0.00805 **
income 0.903315 0.189754 4.760 1.93e-06 ***
log(share) 3.422980 0.530499 6.452 1.10e-10 ***
age 0.022682 0.021895 1.036 0.30024
owneryes 0.705171 0.533070 1.323 0.18589
dependents -0.664933 0.267404 -2.487 0.01290 *
months -0.005723 0.003988 -1.435 0.15130

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1398.53 on 1311 degrees of freedom
Residual deviance: 139.79 on 1304 degrees of freedom
AIC: 155.79

Number of Fisher Scoring iterations: 11

```

Several of the regressors have large  $p$ -values, so `stepAIC()` was used to find a more parsimonious model. The final step where no more variables were deleted is

```
Step: AIC=154.22
card ~ log(reports + 1) + income + log(share) + dependents
```

|                    | Df | Deviance | AIC     |
|--------------------|----|----------|---------|
| <none>             |    | 144.22   | 154.22  |
| - dependents       | 1  | 150.28   | 158.28  |
| - log(reports + 1) | 1  | 164.18   | 172.18  |
| - income           | 1  | 173.62   | 181.62  |
| - log(share)       | 1  | 1079.61  | 1087.61 |

Below is the fit using the model selected by `stepAIC()`. For convenience later, each of the regressors was mean-centered; “\_c” appended to a variable name indicates centering.

```
glm(formula = card ~ log_reports_c + income_c + log_share_c +
dependents_c, family = "binomial", data = CreditCard_clean)
```

Coefficients:

|               | Estimate | Std. Error | z value | Pr(> z )     |
|---------------|----------|------------|---------|--------------|
| (Intercept)   | 9.5238   | 1.7213     | 5.533   | 3.15e-08 *** |
| log_reports_c | -2.8953  | 1.0866     | -2.664  | 0.00771 **   |
| income_c      | 0.8717   | 0.1724     | 5.056   | 4.28e-07 *** |
| log_share_c   | 3.3102   | 0.4942     | 6.698   | 2.11e-11 *** |
| dependents_c  | -0.5506  | 0.2505     | -2.198  | 0.02793 *    |
| ---           |          |            |         |              |

(Dispersion parameter for binomial family taken to be 1)

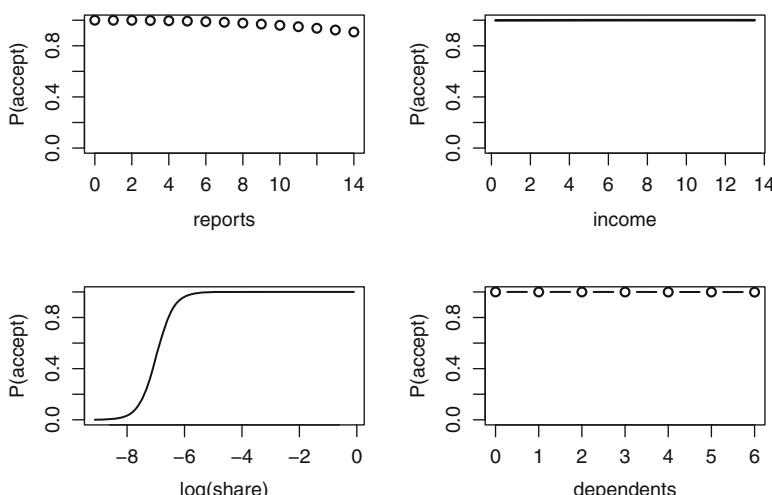
```
Null deviance: 1398.53 on 1311 degrees of freedom
Residual deviance: 144.22 on 1307 degrees of freedom
AIC: 154.22
```

Number of Fisher Scoring iterations: 11

It is important to understand what the logistic regression model is telling us about the probability of an application being accepted. Qualitatively, we see that the probability of having an application accepted increases with `income` and `share` and decreases with `reports` and `dependents`. To understand these effects quantitatively, first consider the intercept. Since the predictors have been mean-centered, the probability of an application being accepted when all variables are at their mean is simply  $H(9.5238) = 0.999927$ . Since `reports` and `dependents` are integer-valued and cannot exactly equal their means, this probability only provides an idea of what the intercept 9.5238 signifies. Figure 11.10 plots the probability that a credit card application is accepted as

functions of `reports`, `income`, `log(share)`, and `dependents`. In each plot, the other variables are fixed at their means. Clearly, the variable with the largest effect is `share`, the ratio of monthly credit card expenditure to yearly income. We see that applicants who spend little of their income through credit cards are unlikely to have their applications accepted.

In Fig. 11.11, panel (a) is a plot of `card`, which takes value 0 if an application is rejected and 1 if it is accepted, versus `log(share)`. It should be emphasized that panel (a) is a plot of the data, not a fit from the model. We see that an application is always accepted if `log(share)` exceeds  $-6$ , which translates into `share` exceeding 0.0025. Thus, in this data set, among the group of applicants whose average monthly credit card expenses exceeded 0.25 % of yearly income, all credit card applications were accepted. How do

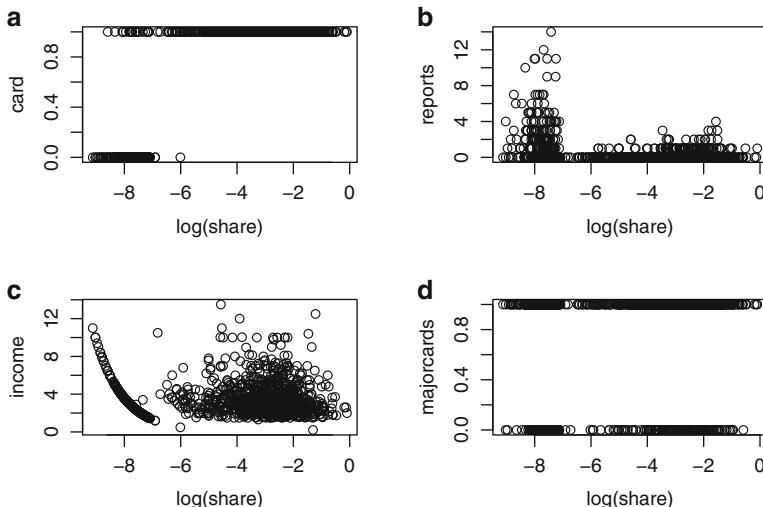


**Fig. 11.10.** Plots of probabilities of a credit card application being accepted as functions of single predictors with other predictors fixed at their means. The variables vary over their ranges in the data.

these applicants look on the other variables? Panels (b)–(d) plot `reports`, `income`, and `majorcards` versus `log(share)`. The variable `majorcards` was not used in the logistic regression analysis but is included here.

An odd feature in Fig. 11.11c is a group of points following a smooth curve. This is a group of 316 applications who had the product of `share` times `income` exactly equal to 0.0012, the minimum value of this product. Oddly, `share` is never 0. Perhaps because of some coding artifact, these 316 had 0 credit card expenditures rather than the reported values. Another interesting feature of the data is that among these 316 applications, only 21 were accepted. Among the remaining 996 applications, all were accepted.

Besides illustrating logistic regression, this example demonstrates that real-world data often contain errors, or perhaps we should call them idiosyncrasies, and that a thorough graphical analysis of the data is always a good thing.  $\square$



**Fig. 11.11.** Plots of  $\log(\text{share})$  versus other variables.

## 11.7 Linearizing a Nonlinear Model

Sometimes a nonlinear model can be linearized by applying a transformation to both the model and the response. In such cases, should one use a linearizing transformation or, instead, apply nonlinear regression to the original model? The answer is that linearization can sometimes be a good thing, but not always. Fortunately, residual analysis can help us decide whether a linearizing transformation should be used.

For example, consider the model

$$Y_i = \beta_1 \exp(\beta_2 X_i). \quad (11.28)$$

This model is “equivalent” to the linear model

$$\log(Y_i) = \alpha + \beta_2 X_i, \quad (11.29)$$

where  $\alpha = \log(\beta_1)$ . “Equivalent” is in quotes, because the two models are no longer equivalent when noise is present.

Suppose (11.28) has i.i.d. additive noise, so that

$$Y_i = \beta_1 \exp(\beta_2 X_i) + \epsilon_i, \quad (11.30)$$

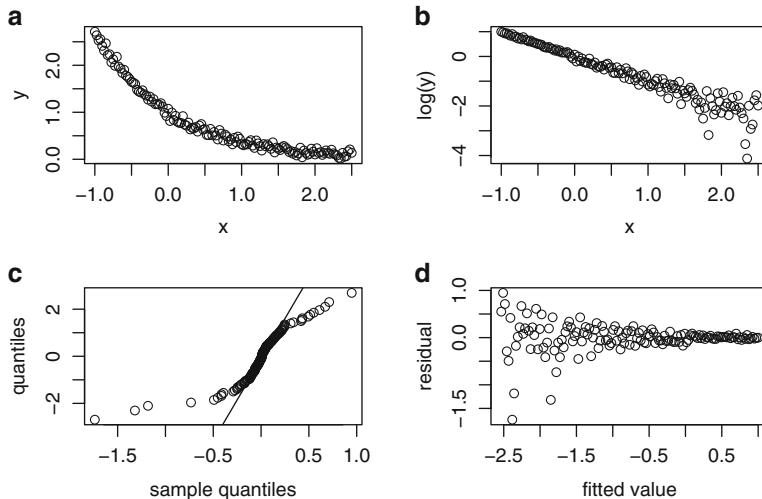
where  $\epsilon_1, \dots, \epsilon_n$  are i.i.d. Then applying the log transformation to (11.29) gives us the model

$$\log(Y_i) = \log\{\beta_1 \exp(\beta_2 X_i) + \epsilon_i\} \quad (11.31)$$

with nonadditive noise. Because the noise is not additive, the variation of  $\log(Y_i)$  about the model  $\log\{\beta_1 \exp(\beta_2 X_i)\}$  will have nonconstant variation and skewness, even if  $\epsilon_1, \dots, \epsilon_n$  are i.i.d. Gaussian.

*Example 11.8. Linearizing transformation—Simulated data*

Figure 11.12a shows a simulated sample from model (11.28) with  $\beta_1 = 1$ ,  $\beta_2 = -1$ , and  $\sigma_\epsilon = 0.02$ . The  $X_i$  are equally spaced from  $-1$  to  $2.5$  by increments of  $0.025$ . Panel (b) shows  $\log(Y_i)$  plotted against  $X_i$ . One can see that the transformation has linearized the relationship between the variables but has introduced nonconstant residual variation. Panels (c) and (d) show residual plots using the linearized model. Notice the nonlinear normal plot and the severe nonconstant variance.  $\square$



**Fig. 11.12.** Example where the log transformation linearizes a model but induces substantial heteroskedasticity and skewness. (a) Raw data. (b) Data after log transformation of the response. (c) Normal plot of residuals after linearization. (d) Absolute residual plot after linearization.

Linearizing is not always a bad thing. Suppose the noise is multiplicative and lognormal so that (11.28) becomes

$$Y_i = \beta_1 \exp(\beta_2 X_i) \exp(\epsilon_i) = \beta_1 \exp(\beta_2 X_i + \epsilon_i), \quad (11.32)$$

where  $\epsilon_1, \dots, \epsilon_n$  are i.i.d. Gaussian. Then the log transformation converts (11.32) to

$$\log(Y_i) = \alpha + \beta_2 X_i + \epsilon_i, \quad (11.33)$$

which is a linear model satisfying all of the usual assumptions.

In summary, a linearizing transformation may or may not cause the data to better follow the assumptions of regression analysis. Residual analysis can help one decide whether a transformation is appropriate.

## 11.8 Robust Regression

A robust regression estimator should be relatively immune to two types of outliers. The first are *bad data*, meaning *contaminants* that are not part of the population, for example, due to undetected recording errors. The second are outliers due to the noise distribution having heavy tails. There are a large number of robust regression estimators, and their sheer number has been an impediment to their use. Many data analysts are confused as to which robust estimator is best and consequently are reluctant to use any. Rather than describe many of these estimators, which might contribute to this problem, we mention just one, the *least-trimmed sum of squares estimator*, often called the *LTS*.

Recall the trimmed mean, a robust estimator of location for a univariate sample. The trimmed mean is simply the mean of the sample after a certain percentage of the largest observations and the same percentage of the smallest observations have been removed. This trimming removes some non-outliers, which, under the ideal conditions of no outliers, causes some loss of precision, but not an unacceptable amount. The trimming also removes outliers, and this causes the estimator to be robust. Trimming is easy for a univariate sample because we know which observations to trim, the very largest and the very smallest. This is not the case in regression. Consider the data in Fig. 11.13. There are 26 observations that fall closely along a line plus two *residual outliers* that are far from this line. Notice that the residual outliers have neither extreme  $X$ -values nor extreme  $Y$ -values. They are outlying only relative to the linear regression fit to the other data.

The residual outliers are obvious in Fig. 11.13 because there is only a single predictor. When there are many predictors, outliers can only be identified when we have a model *and* good estimates of the parameters in that model. The difficulty, then, is that estimation of the parameters requires the identification of the outliers, and vice versa. One can see from the figure that the least-squares line is changed by including the residual outliers in the data used

for estimation. In some cases, e.g., Fig. 10.1b, the effect of a residual outlier can be so severe that it totally changes the least-squares estimates. This is likely to happen if the residual outlier occurs at a high-leverage point.

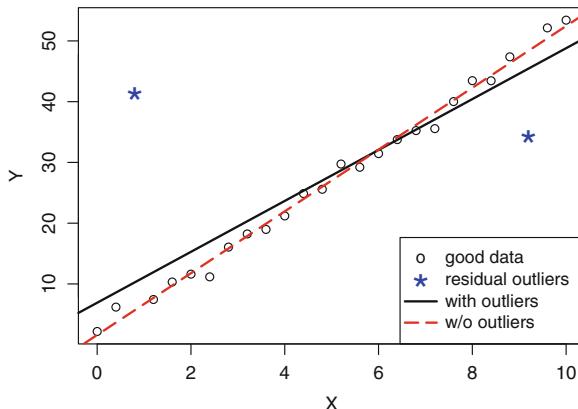
The LTS estimator simultaneously identifies residual outliers and estimates robustly the parameters of a model. Let  $0 < \alpha \leq 1/2$  be the trimming proportion and let  $k$  equal  $n\alpha$  rounded to an integer. The trimmed sum of squares about a set of values of the regression parameters is defined as follows: Form the residuals from the model evaluated at these parameters, square the residuals, then order the squared residuals and remove the  $k$  largest, and finally sum the remaining squared residuals. The LTS estimates are the set of parameter values that minimize the trimmed sum of squares. The LTS estimator can be computed using the `ltsReg()` function in R's `robust` package.

If the noise distribution is heavy-tailed, then an alternative to a robust regression analysis is to use a heavy-tailed distribution as a model for the noise and then to estimate the parameters by maximum likelihood. For example, one could assume that the noise has a double-exponential or  $t$ -distribution. In the latter case, one could either estimate the degrees of freedom or simply fix the degrees of freedom at a low value, which implies heavier tails; see Lange, Little, and Taylor (1989). This strategy is called *robust modeling* rather than robust estimation. The distinction is that in robust estimation one assumes a fairly restrictive model such as a normal noise distribution, but finds a robust alternative to maximum likelihood. In robust modeling, one uses a more flexible model so that maximum likelihood estimation is itself robust. When there is a single gross residual outlier, particularly at a high-leverage point, robust regression is a better alternative than the MLE with a heavy-tailed noise distribution; see the next example.

Another possibility is that residual outliers are due to nonconstant standard deviations, with the outliers mainly in the data with a higher noise standard deviation. The remedy to this problem is to apply a variance stabilization transformation or to model the nonconstant standard deviation, say by one of the GARCH models discussed in Chap. 14.

*Example 11.9. Simulated data in Example 10.1—Robust regression*

Figure 11.14 compares least-squares fit, the LTS fit, and the MLE assuming  $t$ -distributed noise for the simulated data in Example 10.1. In panel (a) with no residual outliers, the three fits coincide. In panels (b) and (c), the LTS fits are not affected by the residual outliers and fit the nonoutlying data very well. In these panels, the LS and MLE fits are highly affected by the outlier and nearly identical. For these examples, the MLE assuming  $t$ -distributed noise is not robust.  $\square$



**Fig. 11.13.** Straight-line regression with two residual outliers showing least-squares fits with and without the outliers.

## 11.9 Regression and Best Linear Prediction

### 11.9.1 Best Linear Prediction

Often we observe a random variable  $X$  and we want to predict an unobserved random variable  $Y$  that is related to  $X$ . For example,  $Y$  could be the future price of an asset and  $X$  might be the most recent change in that asset's price. Prediction has many practical uses, and it is also important in theoretical studies.

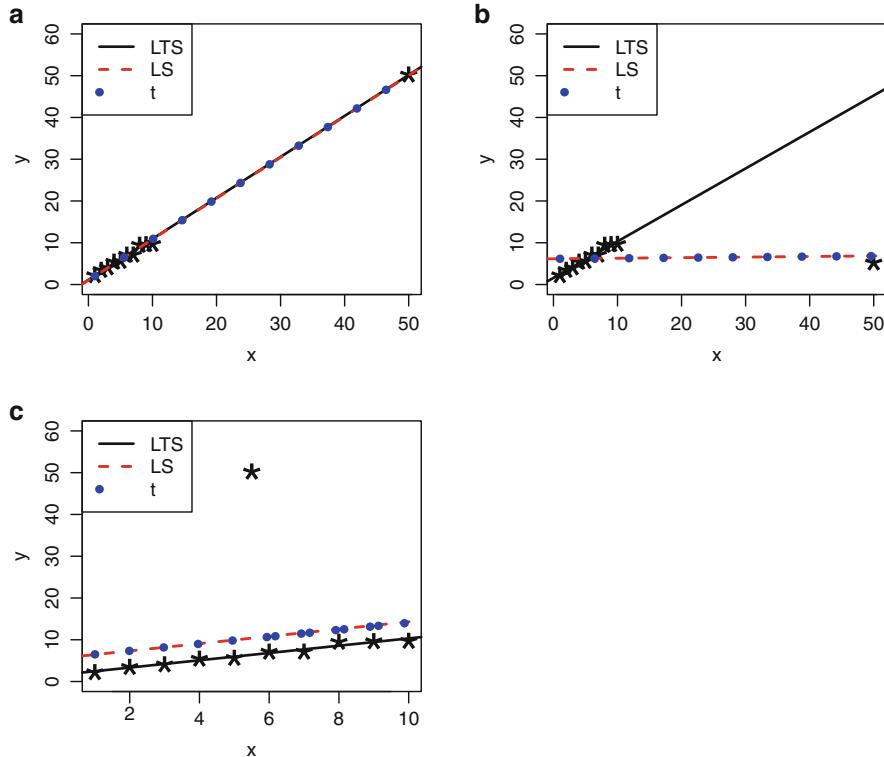
The predictor of  $Y$  that minimizes the expected squared prediction error is  $E(Y|X)$  (see Appendix A.19), but  $E(Y|X)$  is often a nonlinear function of  $X$  and difficult to compute. A common solution to this difficulty is to consider only linear functions of  $X$  as possible predictors. This is called *linear prediction*. In this section, we will show that linear prediction is closely related to linear regression.

A linear predictor of  $Y$  based on  $X$  is a function  $\beta_0 + \beta_1 X$  where  $\beta_0$  and  $\beta_1$  are parameters that we can choose. *Best linear prediction* means finding  $\beta_0$  and  $\beta_1$  so that expected squared prediction error, which is given by

$$E\{Y - (\beta_0 + \beta_1 X)\}^2, \quad (11.34)$$

is minimized. Doing this makes the predictor as close as possible, on average, to  $Y$ . The expected squared prediction error can be rewritten as

$$\begin{aligned} & E\{Y - (\beta_0 + \beta_1 X)\}^2 \\ &= E(Y^2) - 2\beta_0 E(Y) - 2\beta_1 E(XY) + \beta_0^2 + 2\beta_0\beta_1 E(X) + \beta_1^2 E(X^2). \end{aligned}$$



**Fig. 11.14.** Simulated data in Example 10.1 with LS fits (dashed red) and LTS fits (solid black) and MLEs assuming  $t$ -distributed noise (dotted blue). In (a) the three fits are too close together to distinguish between them. In (b) and (c) the LS and  $t$  fits are nearly identical and difficult to distinguish.

To find the minimizers, we set the partial derivatives of this expression to zero to obtain

$$0 = -E(Y) + \beta_0 + \beta_1 E(X), \quad (11.35)$$

$$0 = -E(XY) + \beta_0 E(X) + \beta_1 E(X^2). \quad (11.36)$$

After some algebra we find that

$$\beta_1 = \sigma_{XY}/\sigma_X^2 \quad (11.37)$$

and

$$\beta_0 = E(Y) - \beta_1 E(X) = E(Y) - \sigma_{XY}/\sigma_X^2 E(X). \quad (11.38)$$

One can check that the matrix of second derivatives of (11.34) is positive definite so that the solution  $(\beta_0, \beta_1)$  to (11.35) and (11.36) minimizes (11.34). Thus, the best linear predictor of  $Y$  is

$$\hat{Y}^{\text{Lin}}(X) = \beta_0 + \beta_1 X = E(Y) + \frac{\sigma_{XY}^2}{\sigma_X^2} \{X - E(X)\}. \quad (11.39)$$

In practice, (11.39) cannot be used directly unless  $E(X)$ ,  $E(Y)$ ,  $\sigma_{XY}$ , and  $\sigma_X^2$  are known, which is often not the case. Linear regression analysis is essentially the use of (11.39) with these unknown parameters replaced by least-squares estimates—see Sect. 11.9.3.

### 11.9.2 Prediction Error in Best Linear Prediction

In this section, assume that  $\hat{Y}$  is the best linear predictor of  $Y$ . The *prediction error* is  $Y - \hat{Y}$ . It is easy to show that  $E\{Y - \hat{Y}\} = 0$  so that the prediction is unbiased. With a little algebra we can show that the expected squared prediction error is

$$E\{Y - \hat{Y}\}^2 = \sigma_Y^2 - \frac{\sigma_{XY}^2}{\sigma_X^2} = \sigma_Y^2(1 - \rho_{XY}^2). \quad (11.40)$$

How much does  $X$  help us predict  $Y$ ? To answer this question, notice first that if we do not observe  $X$ , then we must predict  $Y$  using a constant, which we denote by  $c$ . It is easy to show that the best predictor has  $c$  equal to  $E(Y)$ . Notice first that the expected squared prediction error is  $E(Y - c)^2$ . Some algebra shows that

$$E(Y - c)^2 = \text{Var}(Y) + \{c - E(Y)\}^2, \quad (11.41)$$

which, since  $\text{Var}(Y)$  does not depend on  $c$ , shows that the expected squared prediction error is minimized by  $c = E(Y)$ . Thus, when  $X$  is unobserved, the best predictor of  $Y$  is  $E(Y)$  and the expected squared prediction error is  $\sigma_Y^2$ , but when  $X$  is observed, then the expected squared prediction error is smaller,  $\sigma_Y^2(1 - \rho_{XY}^2)$ . Therefore,  $\rho_{XY}^2$  is the fraction by which the prediction error is reduced when  $X$  is known. This is an important fact that we will see again.

**Result 11.1** Prediction when  $Y$  is independent of all available information:

*If  $Y$  is independent of all presently available information, that is,  $Y$  is independent of all random variables that have been observed, then the best predictor of  $Y$  is  $E(Y)$  and the expected value of the squared prediction error is  $\sigma_Y^2$ . We say that  $Y$  “cannot be predicted” when there exists no predictor better than its expected value.*

### 11.9.3 Regression Is Empirical Best Linear Prediction

For the case of a single predictor, note the similarity between the best linear predictor,

$$\hat{Y} = E(Y) + \frac{\sigma_{XY}}{\sigma_X^2} \{X - E(X)\},$$

and the least-squares line,

$$\hat{Y} = \bar{Y} + \frac{s_{XY}}{s_X^2} (X - \bar{X}).$$

The least-squares line is a sample version of the best linear predictor. Also,  $\rho_{XY}^2$ , the squared correlation between  $X$  and  $Y$ , is the fraction of variation in  $Y$  that can be predicted using the linear predictor, and the sample version of  $\rho_{XY}^2$  is  $R^2 = r_{XY}^2 = r_{\hat{Y}Y}^2$ .

### 11.9.4 Multivariate Linear Prediction

So far we have assumed that there is only a single random variable,  $X$ , available to predict  $Y$ . More commonly,  $Y$  is predicted using a set of observed random variables,  $X_1, \dots, X_n$ .

Let  $\mathbf{Y}$  and  $\mathbf{X}$  by  $p \times 1$  and  $q \times 1$  random vectors. As before in Sect. 7.3.1, define

$$\boldsymbol{\Sigma}_{Y,X} = E\{\mathbf{Y} - E(\mathbf{Y})\}\{\mathbf{X} - E(\mathbf{X})\}^\top,$$

so that the  $i, j$ th element of  $\boldsymbol{\Sigma}_{Y,X}$  is the covariance between  $Y_i$  and  $X_j$ . Then the best linear predictor of  $\mathbf{Y}$  given  $\mathbf{X}$  is

$$\hat{\mathbf{Y}} = E(\mathbf{Y}) + \boldsymbol{\Sigma}_{Y,X} \boldsymbol{\Sigma}_X^{-1} \{\mathbf{X} - E(\mathbf{X})\}. \quad (11.42)$$

Note the similarity between (11.39) and (11.42), the best linear predictors in the univariate and multivariate cases.

The sample analog of multivariate linear prediction is multiple regression.

## 11.10 Regression Hedging

An interesting application of regression is determining the optimal hedge of a bond position. Market makers buy securities at a *bid price* and make a profit by selling them at a higher *ask price*. Suppose a market maker has just purchased a bond from a pension fund. Ideally, the market maker would sell the bond immediately after purchasing it. However, many bonds are illiquid, so it may take some time before the bond can be sold. During the period that a market maker is holding a bond, the market maker is at risk that the bond price could drop due to a change in interest rates. The change could wipe out the profit due to the small bid–ask spread. The market maker would prefer to

hedge this risk by assuming another risk which is likely to be in the opposite direction. To hedge the interest-rate risk of the bond being held, the market maker can sell other, more liquid, bonds short. Suppose that the market maker decides to sell short a 30-year Treasury bond, which is more liquid.

*Regression hedging* determines the optimal amount of the 30-year Treasury bonds to sell short to hedge the risk of the bond just purchased. The goal is that the price of the portfolio long in the first bond and short in the Treasury bond changes as little as possible as yields change. Suppose the first bond has a maturity of 25 years. One can determine the sensitivity of price to yield changes using results from Sect. 3.8. Let  $y_{30}$  be the yield on 30-year bonds, let  $P_{30}$  be the price of \$1 in face amount of 30-year bonds, and let  $DUR_{30}$  be the duration. The change in price,  $\Delta P_{30}$ , and the change in yield,  $\Delta y_{30}$ , are related by

$$\Delta P_{30} \approx -P_{30} DUR_{30} \Delta y_{30}$$

for small values of  $\Delta y_{30}$ . A similar result holds for 25-year bonds.

Consider a portfolio that holds face amount  $F_{25}$  in 25-year bonds and is short face amount  $F_{30}$  in 30-year bonds. The value of the portfolio is

$$F_{25}P_{25} - F_{30}P_{30}.$$

If  $\Delta y_{25}$  and  $\Delta y_{30}$  are the changes in the yields, then the change in value of the portfolio is approximately

$$\{F_{30}P_{30} DUR_{30} \Delta y_{30} - F_{25}P_{25} DUR_{25} \Delta y_{25}\}. \quad (11.43)$$

Suppose that the regression of  $\Delta y_{30}$  on  $\Delta y_{25}$  is

$$\Delta y_{30} = \hat{\beta}_0 + \hat{\beta}_1 \Delta y_{25} \quad (11.44)$$

and  $\hat{\beta}_0 \approx 0$ , as is usually the case for regression of changes in interest rates, as in Example 9.1. Substituting (11.44) into (11.43), the change in price of the portfolio is approximately

$$\{F_{30}P_{30} DUR_{30}\hat{\beta}_1 - F_{25}P_{25} DUR_{25}\} \Delta y_{25}. \quad (11.45)$$

This change is approximately zero for all values of  $\Delta y_{25}$  if

$$F_{30} = F_{25} \frac{P_{25} DUR_{25}}{P_{30} DUR_{30} \hat{\beta}_1}. \quad (11.46)$$

Equation (11.46) tells us how much face value of the 30-year bond to sell short in order to hedge  $F_{25}$  face value of the 25-year bond. All quantities on the right-hand side of (11.46) are known or readily calculated:  $F_{25}$  is the current position in the 25-year bond,  $P_{25}$  and  $P_{30}$  are known bond prices, calculation of  $DUR_{25}$  and  $DUR_{30}$  is discussed in Chap. 3, and  $\hat{\beta}_1$  is the slope of the regression of  $\Delta y_{30}$  on  $\Delta y_{25}$ .

The higher the  $R^2$  of the regression, the better the hedge works. Hedging with two or more liquid bonds, say a 30-year and a 10-year, can be done by multiple regression and might produce a better hedge.

## 11.11 Bibliographic Notes

Atkinson (1985) has nice coverage of transformations and residual plotting and many good examples. For more information on nonlinear regression, see Bates and Watts (1988) and Seber and Wild (1989). Graphical methods for detecting a nonconstant variance, transform-both-sides regression, and weighting are discussed in Carroll and Ruppert (1988). Hosmer and Lemeshow (2000) is an in-depth treatment of logistic regression. Faraway (2006) covers generalized linear models including logistic regression. See Tuckman (2002) for more discussion of regression hedging.

The Nelson–Siegel and Svensson models are from Nelson and Siegel (1985) and Svensson (1994).

## 11.12 R Lab

### 11.12.1 Nonlinear Regression

In this section, you will be fitting short-rate models. Let  $r_t$  be the short rate (the risk-free rate for short-term borrowing) at time  $t$ . It is assumed that the short rate satisfies the stochastic differential equation

$$dr_t = \mu(t, r_t) dt + \sigma(t, r_t) dW_t, \quad (11.47)$$

where  $\mu(t, r_t)$  is a drift function,  $\sigma(t, r_t)$  is a volatility function, and  $W_t$  is a standard Brownian motion. We will use a discrete approximation to (11.47):

$$(r_t - r_{t-1}) = \mu(t-1, r_{t-1}) + \sigma(t-1, r_{t-1}) \epsilon_{t-1} \quad (11.48)$$

where  $\epsilon_1, \dots, \epsilon_{n-1}$  are i.i.d.  $N(0, 1)$ .

We will start with the Chan, Karolyi, Longstaff, and Sanders (1992) (CKLS) model, which assumes that

$$\mu(t, r) = \mu(r) = a(\theta - r) \quad (11.49)$$

for some unknown parameters  $a$  and  $\theta$ , and

$$\sigma(t, r) = \sigma r^\gamma \quad (11.50)$$

for some  $\sigma$  and  $\gamma$ . Be careful to distinguish between the volatility function  $\sigma(t, r)$  and the constant volatility parameter  $\sigma$ .

We will use the `Irates` data set in the `Ecdat` package. This data set has interest rates for maturities from 1 to 120 months. We will use the first column, which has the one-month maturity rates, since we want the short rate.

Run the following code to input the data, compute the lagged and differenced short-rate series, and construct some basic plots.

```

library(Ecdat)
data(Irates)
r1 = Irates[,1]
n = length(r1)
lag_r1 = lag(r1)[-n]
delta_r1 = diff(r1)
n = length(lag_r1)
par(mfrow = c(3, 2))
plot(r1, main = "(a)")
plot(delta_r1, main = "(b)")
plot(delta_r1^2, main = "(c)")
plot(lag_r1, delta_r1, main = "(d)")
plot(lag_r1, delta_r1^2, main = "(e)")

```

**Problem 1** What is the maturity of the interest rates in the first column? What is the sampling frequency of this data set—daily, weekly, monthly, or quarterly? What country are the data from? Are the rates expressed as percentages or fractions (decimals)?

In the plot you have just created, panels (a), (b), and (c) show how the short rate, changes in the short rate, and squared changes in the short rate depend on time. The plots of changes in the short rate are useful for choosing the drift  $\mu(t - 1, r_{t-1})$  while squared changes in the short rate are helpful for selecting the volatility  $\sigma(t - 1, r_{t-1})$ .

**Problem 2** Model (11.49) states that  $\mu(t, r) = \mu(r)$ , that is, that the drift does not depend on  $t$ . Use your plots to discuss whether this assumption seems valid. Assuming for the moment that this assumption is valid, any trend in the plot in panel (d) would give us information about the form of  $\mu(r)$ . Do you see any trend?

Now run the following code to fit model (11.49) and fill in the first two panels of a figure. This figure will be continued next.

```

CKLS (Chan, Karolyi, Longstaff, Sanders)

nlmod_CKLS = nls(delta_r1 ~ a * (theta-lag_r1),
 start=list(theta = 5, a = 0.01),
 control = list(maxiter = 200))
param = summary(nlmod_CKLS)$parameters[, 1]
par(mfrow = c(2, 2))
t = seq(from = 1946, to = 1991 + 2 / 12, length = n)
plot(lag_r1, ylim = c(0, 16), ylab = "rate and theta",
 main = "(a)", type = "l")
abline(h = param[1], lwd = 2, col = "red")

```

**Problem 3** What are the estimates of  $a$  and  $\theta$  and their 95% confidence intervals?

Note that the nonlinear regression analysis estimates  $\sigma^2(r)$ , not  $\sigma(r)$ , since the response variable is the squared residual. Here  $A = \sigma^2$  and  $B = 2\gamma$ .

```
res_sq = residuals(nlmod_CKLS)^2
nlmod_CKLS_res <- nls(res_sq ~ A*lag_r1^B,
 start = list(A = 0.2, B = 1/2))
param2 = summary(nlmod_CKLS_res)$parameters[, 1]
plot(lag_r1, sqrt(res_sq), pch = 5, ylim = c(0, 6),
 main = "(b)")
attach(as.list(param2))
curve(sqrt(A * x^B), add = T, col = "red", lwd = 3)
```

**Problem 4** What are the estimates of  $\sigma$  and  $\gamma$  and their 95 % confidence intervals?

Finally, refit model (11.49) using weighted least squares.

```
nlmod_CKLS_wt = nls(delta_r1 ~ a * (theta-lag_r1),
 start = list(theta = 5, a = 0.01),
 control = list(maxiter = 200),
 weights = 1 / fitted(nlmod_CKLS_res))

plot(lag_r1, ylim = c(0, 16), ylab = "rate and theta",
 main = "(c)", type = "l")
param3 = summary(nlmod_CKLS_wt)$parameters[, 1]
abline(h = param3[1], lwd = 2, col = "red")
```

**Problem 5** How do the unweighted estimate of  $\theta$  shown in panel (a) and the weighted estimate plotted in panel (d) differ? Why do they differ in this manner?

### 11.12.2 Response Transformations

This section uses the `HousePrices` data set in the `AER` package. This is a cross-sectional data set on house prices and other features, e.g., the number of bedrooms of houses in Windsor, Ontario. The data were gathered during the summer of 1987. Accurate modeling of house prices is important for the mortgage industry. Run the code below to read the data and regress `price` on the other variables; the period on the right-hand side of the formula “`price~.`” specifies that the predictors should include all variables except, of course, the response.

```
library(AER)
data(HousePrices)
fit1 = lm(price ~ ., data = HousePrices)
summary(fit1)
```

Next construct a profile log-likelihood plot for the transformation parameter  $\alpha$  in model (11.25)

```
library(MASS)
fit2 = boxcox(fit1, xlab = expression(alpha))
```

**Problem 6** What is the MLE of  $\alpha$ ? (Hint: Type `?boxcox` to learn what is returned by this function.)

Next, fit a linear model with `price` transformed by  $\hat{\alpha}$  (the MLE). Here the function `bcPower()` in the `AER` package computes a Box–Cox transformation for a given value of  $\alpha$  and must be distinguished from `boxcox()`, which computes the profile log-likelihood for  $\alpha$ . In the following code, replace  $1/2$  by the MLE of  $\alpha$ .

```
library(car)
alphahat = 1/2
fit3 = lm(bcPower(price, alphahat) ~ ., data = HousePrices)
summary(fit3)
AIC(fit1)
AIC(fit3)
```

**Problem 7** Does the Box–Cox transformation offer a substantial improvement in fit compared to the regression with no transformation of `price`?

**Problem 8** Would it be worthwhile to check the residuals for correlation?

### 11.12.3 Binary Regression: Who Owns an Air Conditioner?

This section uses the `HousePrices` data set used in Sect. 11.12.2. The goal here is to investigate how the presence or absence of air conditioning is related to the other variables. The code below fits a logistic regression model to all potential predictor variables and then uses `stepAIC()` to find a parsimonious model.

```
library(AER)
data(HousePrices)
fit1 = glm(aircon ~ ., family = "binomial",
 data = HousePrices)
summary(fit1)
library(MASS)
fit2 = stepAIC(fit1)
summary(fit2)
```

**Problem 9** Which variables are most useful for predicting whether a home has air conditioning? Describe qualitatively the relationships between these variables and the variable `aircon`. Are there any variables in the model selected by `stepAIC()` that you think might be dropped?

**Problem 10** Estimate the probability that a house will have air conditioning if it has the following characteristics:

|          |         |          |           |         |          |            |    |
|----------|---------|----------|-----------|---------|----------|------------|----|
| price    | lotsize | bedrooms | bathrooms | stories | driveway | recreation |    |
| 42000    | 5850    | 3        |           | 1       | 2        | yes        | no |
| fullbase | gasheat | garage   | prefer    |         |          |            |    |
| yes      | no      | 1        | no        |         |          |            |    |

(Hint: The R function `plogis()` computes the logistic function.)

### 11.13 Exercises

- When we were finding the best linear predictor of  $Y$  given  $X$ , we derived the equations

$$0 = -E(Y) + \beta_0 + \beta_1 E(X)$$

$$0 = -E(XY) + \beta_0 E(X) + \beta_1 E(X^2).$$

Show that their solution is

$$\beta_1 = \frac{\sigma_{XY}}{\sigma_X^2}$$

and

$$\beta_0 = E(Y) - \beta_1 E(X) = E(Y) - \frac{\sigma_{XY}}{\sigma_X^2} E(X).$$

- Suppose one has a long position of  $F_{20}$  face value in 20-year Treasury bonds and wants to hedge this with short positions in both 10- and 30-year Treasury bonds. The prices and durations of 10-, 20-, and 30-year Treasury bonds are  $P_{10}$ ,  $DUR_{10}$ ,  $P_{20}$ ,  $DUR_{20}$ ,  $P_{30}$ , and  $DUR_{30}$  and are assumed to be known. A regression of changes in the 20-year yield on changes in the 10- and 30-year yields is  $\Delta y_{20} = \hat{\beta}_0 + \hat{\beta}_1 \Delta y_{10} + \hat{\beta}_2 \Delta y_{30}$ . The  $p$ -value of  $\hat{\beta}_0$  is large and it is assumed that  $\beta_0$  is close enough to zero to be ignored. What face amounts  $F_{10}$  and  $F_{30}$  of 10- and 30-year Treasury bonds should be shorted to hedge the long position in 20-year Treasury bonds? (Express  $F_{10}$  and  $F_{30}$  in terms of the known quantities  $P_{10}$ ,  $P_{20}$ ,  $P_{30}$ ,  $DUR_{10}$ ,  $DUR_{20}$ ,  $DUR_{30}$ ,  $\hat{\beta}_1$ ,  $\hat{\beta}_2$ , and  $F_{20}$ .)
- The maturities ( $T$ ) in years and prices in dollars of zero-coupon bonds are in file `ZeroPrices.txt` on the book's website. The prices are expressed

as percentages of par. A popular model is the Nelson–Siegel family with forward rate

$$r(T; \theta_1, \theta_2, \theta_3, \theta_4) = \theta_1 + (\theta_2 + \theta_3 T) \exp(-\theta_4 T).$$

Fit this forward rate to the prices by nonlinear regression using R's `optim()` function.

- (a) What are your estimates of  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , and  $\theta_4$ ?
  - (b) Plot the estimated forward rate and estimated yield curve on the same figure. Include the figure with your work.
4. Least-squares estimators are unbiased in linear models, but in nonlinear models they can be biased. Simulation studies (including bootstrap resampling) can be used to estimate the amount of bias. In Example 11.1, the data were simulated with  $r = 0.06$  and  $\hat{r} = 0.0585$ . Do you think this is a sign of bias or simply due to random variability? Justify your answer.

## References

- Atkinson, A. C. (1985) *Plots, Transformations and Regression*, Clarendon, Oxford.
- Bates, D. M., and Watts, D. G. (1988) *Nonlinear Regression Analysis and Its Applications*, Wiley, New York.
- Bluhm, C., Overbeck, L., and Wagner, C. (2003) *An Introduction to Credit Risk Modelling*, Chapman & Hall/CRC, Boca Raton, FL.
- Box, G. E. P., and Dox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, **26** 211–246.
- Carroll, R. J., and Ruppert, D. (1988) *Transformation and Weighting in Regression*, Chapman & Hall, New York.
- Chan, K. C., Karolyi, G. A., Longstaff, F. A., and Sanders, A. B. (1992) An empirical comparison of alternative models of the short-term interest rate. *Journal of Finance*, **47**, 1209–1227.
- Faraway, J. J. (2006) *Extending the Linear Model with R*, Chapman & Hall, Boca Raton, FL.
- Hosmer, D., and Lemeshow, S. (2000) *Applied Logistic Regression*, 2nd ed., Wiley, New York.
- Jarrow, R. (2002) *Modeling Fixed-Income Securities and Interest Rate Options*, 2nd Ed., Stanford University Press, Stanford, CA.
- Lange, K. L., Little, R. J. A., and Taylor, J. M. G. (1989) Robust statistical modeling using the  $t$ -distribution. *Journal of the American Statistical Association*, **84**, 881–896.
- Nelson, C. R., and Siegel, A. F. (1985) Parsimonious modelling of yield curves. *Journal of Business*, **60**, 473–489.

- Seber, G. A. F., and Wild, C. J. (1989) *Nonlinear Regression*, Wiley, New York.
- Svensson, L. E. (1994) Estimating and interpreting forward interest rates: Sweden 1992–94, Working paper. International Monetary Fund, 114.
- Tuckman, B. (2002) *Fixed Income Securities*, 2nd ed., Wiley, Hoboken, NJ.

## Time Series Models: Basics

### 12.1 Time Series Data

A *time series* is a sequence of observations in chronological order, for example, daily log returns on a stock or monthly values of the Consumer Price Index (CPI). A common simplifying assumption is that the data are equally spaced with a discrete-time observation index; however, this may only hold approximately. For example, daily log returns on a stock may only be available for weekdays, with additional gaps on holidays, and monthly values of the CPI are equally spaced by month, but unequally spaced by days. In either case, the consecutive observations are commonly regarded as equally spaced, for simplicity. In this chapter, we study statistical models for time series. These models are widely used in econometrics, business forecasting, and many scientific applications.

A *stochastic process* is a sequence of random variables and can be viewed as the “theoretical” or “population” analog of a time series—conversely, a time series can be considered a sample from a stochastic process. “Stochastic” is a synonym for random. One of the most useful methods for obtaining parsimony in a time series model is to assume some form of distributional invariance over time, or *stationarity*, a property discussed next.

### 12.2 Stationary Processes

When we observe a time series, the fluctuations appear random, but often with the same type of stochastic behavior from one time period to the next. For example, returns on stocks or changes in interest rates can be very different from the previous year, but the mean, standard deviation, and other statistical

properties often are similar from one year to the next.<sup>1</sup> Similarly, the demand for many consumer products, such as sunscreen, winter coats, and electricity, has random as well as seasonal variation, but each summer is similar to past summers, each winter to past winters, at least over shorter time periods. *Stationary stochastic processes* are probability models for time series with time-invariant behavior.

A process is said to be *strictly stationary* if all aspects of its behavior are unchanged by shifts in time. Mathematically, stationarity is defined as the requirement that for every  $m$  and  $n$ , the distributions of  $(Y_1, \dots, Y_n)$  and  $(Y_{1+m}, \dots, Y_{n+m})$  are the same; that is, the probability distribution of a sequence of  $n$  observations does not depend on their time origin. Strict stationarity is a very strong assumption, because it requires that “all aspects” of stochastic behavior be constant in time. Often, it will suffice to assume less, namely, weak stationarity. A process is *weakly stationary* if its mean, variance, and covariance are unchanged by time shifts. More precisely,  $Y_1, Y_2, \dots$  is a *weakly stationary process* if

- $E(Y_t) = \mu$  (a finite constant) for all  $t$ ;
- $\text{Var}(Y_t) = \sigma^2$  (a positive finite constant) for all  $t$ ; and
- $\text{Cov}(Y_t, Y_s) = \gamma(|t - s|)$  for all  $t$  and  $s$  for some function  $\gamma(h)$ .

Thus, the mean and variance do not change with time and the covariance between two observations depends only on the *lag*, the time distance  $|t - s|$  between them, not the indices  $t$  or  $s$  directly. For example, if the process is weakly stationary, then the covariance between  $Y_2$  and  $Y_5$  is the same as the covariance between  $Y_7$  and  $Y_{10}$ , since each pair is separated by three units of time. The adjective “weakly” in “weakly stationary” refers to the fact that we are only assuming that means, variance, and covariances, not other distributional characteristics such as quantiles, skewness, and kurtosis, are stationary. Weakly stationary is also sometimes referred to as *covariance stationary*. The term *stationary* will sometimes be used as a shorthand for strictly stationary.

The function  $\gamma$  is called the *autocovariance function* of the process. Note that  $\gamma(h) = \gamma(-h)$ . Why? Assuming weak stationarity, the correlation between  $Y_t$  and  $Y_{t+h}$  is denoted by  $\rho(h)$ . The function  $\rho$  is called the *autocorrelation function*. Note that  $\gamma(0) = \sigma^2$  and that  $\gamma(h) = \sigma^2 \rho(h)$ . Also,  $\rho(h) = \gamma(h)/\sigma^2 = \gamma(h)/\gamma(0)$ .

As mentioned, many financial time series do not exhibit stationarity, but often the *changes* in them, perhaps after applying a log transformation, are approximately stationary. For this reason, stationary time series models have broad applicability and wide ranging applications. From the viewpoint of

---

<sup>1</sup> It is the returns, not the stock prices, that have time-invariant behavior. Stock prices themselves tend to increase over time, so this year’s stock prices tend to be higher and more variable than those a decade or two ago.

statistical modeling, it is not important whether it is the time series itself or changes in the time series that are stationary, because either way we get a parsimonious model.

The beauty of a stationary process is that it can be modeled with relatively few parameters. For example, we do not need a different expectation for each  $Y_t$ ; rather they all have a common expectation,  $\mu$ . This implies that  $\mu$  can be estimated accurately by  $\bar{Y}$ . If instead we did not assume stationarity and each  $Y_t$  had its own unique expectation,  $\mu_t$ , then it would not be possible to estimate  $\mu_t$  accurately— $\mu_t$  could only be estimated by the single observation  $Y_t$  itself.

When a time series is observed, a natural question is whether it appears to be stationary. This is not an easy question to address, and we can never be absolutely certain of the answer. However, visual inspection of the time series and changes in the time series can be helpful. A *time series plot* is a plot of the series in chronological order. This very basic plot is useful for assessing stationary behavior, though it can be supplemented with other plots, such as the plot of the sample autocorrelation function that will be introduced later. In addition, there are statistical tests of stationarity—these are discussed in Sect. 12.10.

A time series plot of a stationary series should show random oscillation around some fixed level, a phenomenon called *mean-reversion*. If the series wanders without returning repeatedly to some fixed level, then the series should not be modeled as a stationary process.

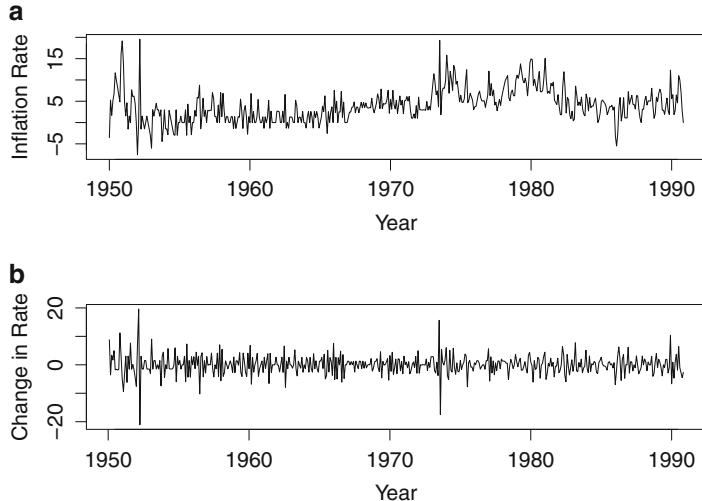
*Example 12.1. Inflation rates and changes in inflation rates—time series plots*

The one-month inflation rate (annual rate, in percent) is shown in Fig. 12.1a. The data come from the `Mishkin` data set in R's `Ecdat` package. The series may be wandering without reverting to a fixed mean, or it may be slowly reverting to a mean of approximately 4%, as would be expected with a stationary time series. In panel (b), the first differences, that is, the changes from one month to the next, are shown. In contrast to the original series, the differenced series certainly oscillate around a fixed mean that is 0%, or nearly so. The differenced series appears stationary, but whether or not the original series is stationary needs further investigation. We will return to this question later.  $\square$

*Example 12.2. Air passengers*

Figure 12.2 is a plot of monthly total international airline passengers for the years 1949 to 1960. The data come from the `AirPassengers` data set in R's `Datasets` package. There are three types of nonstationarity seen in the

plot. First is the obvious upward trend, second is the seasonal variation with local peaks in summer and troughs in winter months, and third is the increase over time in the size of the seasonal oscillations.  $\square$



**Fig. 12.1.** Time series plots of (a) one-month inflation rate (annual rate, in percent) and (b) first differences (changes) in the one-month inflation rate. It is unclear if the series in (a) is stationary, but the differenced series in (b) seems suitable for modeling as stationary.

### 12.2.1 White Noise

White noise is the simplest example of a stationary process. We will define several types of white noise with increasingly restrictive assumptions.

The sequence  $Y_1, Y_2, \dots$  is a *weak white noise process* with mean  $\mu$  and variance  $\sigma^2$ , which will be shortened to “weak WN( $\mu, \sigma^2$ ),” if

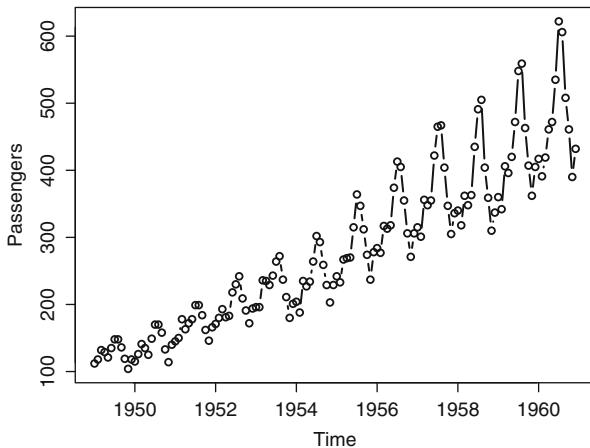
- $E(Y_t) = \mu$  (a finite constant) for all  $t$ ;
- $\text{Var}(Y_t) = \sigma^2$  (a positive finite constant) for all  $t$ ; and
- $\text{Cov}(Y_t, Y_s) = 0$  for all  $t \neq s$ .

If the mean is not specified, then it is assumed that  $\mu = 0$ . A weak white noise process is weakly stationary with

$$\begin{aligned}\gamma(0) &= \sigma^2, \\ \gamma(h) &= 0 \quad \text{if } h \neq 0,\end{aligned}$$

so that

$$\begin{aligned}\rho(0) &= 1, \\ \rho(h) &= 0 \text{ if } h \neq 0.\end{aligned}$$



**Fig. 12.2.** Time series plot of monthly totals of air passengers (in thousands).

If  $Y_1, Y_2, \dots$  is an i.i.d. process, then we call it an *i.i.d. white noise process* or simply *i.i.d. WN*( $\mu, \sigma^2$ ). Weak white noise is weakly stationary, while i.i.d. white noise is strictly stationary. An i.i.d. white noise process with  $\sigma^2$  finite is also a weak white noise process, but not vice versa.

If, in addition,  $Y_1, Y_2, \dots$  is an i.i.d. process with a specific marginal distribution, then this might be noted. For example, if  $Y_1, Y_2, \dots$  are i.i.d. normal random variables, then the process is called a *Gaussian white noise process*. Similarly, if  $Y_1, Y_2, \dots$  are i.i.d.  $t$  random variables with  $\nu$  degrees of freedom, then it is called a  $t_\nu$  white noise process.

### 12.2.2 Predicting White Noise

Because of the lack of dependence, past values of a white noise process contain no information that can be used to predict future values. More precisely, suppose that  $Y_1, Y_2, \dots$  is an i.i.d. WN( $\mu, \sigma^2$ ) process. Then

$$E(Y_{t+h}|Y_1, \dots, Y_t) = \mu \text{ for all } h \geq 1. \quad (12.1)$$

What this equation is saying is that one cannot predict the future deviations of a white noise process from its mean, because its future is independent of its past and present. Therefore, the best predictor of any future value of the

process is simply the mean  $\mu$ , what you would use even if  $Y_1, \dots, Y_t$  had not been observed. For weak white noise, (12.1) need not be true, but it is still true that the best linear predictor<sup>2</sup> of  $Y_{t+h}$  given  $Y_1, \dots, Y_t$  is  $\mu$ .

## 12.3 Estimating Parameters of a Stationary Process

Suppose we observe  $Y_1, \dots, Y_n$  from a weakly stationary process. To estimate the mean  $\mu$  and variance  $\sigma^2$  of the process, we can use the sample mean  $\bar{Y}$  and sample variance  $s^2$ . To estimate the autocovariance function, we use the *sample autocovariance function*

$$\hat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (Y_{t+h} - \bar{Y})(Y_t - \bar{Y}) = n^{-1} \sum_{t=h+1}^n (Y_t - \bar{Y})(Y_{t-h} - \bar{Y}). \quad (12.2)$$

Equation (12.2) is an example of the usefulness of parsimony induced by the stationarity assumption. Because the covariance between  $Y_t$  and  $Y_{t+h}$  does not depend on  $t$ , all  $n - h$  pairs of data points that are separated by a lag of  $h$  time units can be used to estimate  $\gamma(h)$ . Some authors define  $\hat{\gamma}(h)$  with the factor  $n^{-1}$  in (12.2) replaced by  $(n - h)^{-1}$ , but this change has little effect if  $n$  is reasonably large and  $h$  is small relative to  $n$ , as is typically the case.

To estimate  $\rho(\cdot)$ , we use the *sample autocorrelation function (sample ACF)* defined as

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}.$$

### 12.3.1 ACF Plots and the Ljung–Box Test

Most statistical software will plot a sample ACF with *test bounds*. These bounds are used to test the null hypothesis that an autocorrelation coefficient is 0. The null hypothesis is rejected if the sample autocorrelation is outside the bounds. The usual level of the test is 0.05, so one can expect to see about 1 out of 20 sample autocorrelations outside the test bounds simply by chance.

An alternative to using the bounds to test the autocorrelations one at a time is to use a simultaneous test. A *simultaneous test* is one that tests whether a group of null hypotheses are all true versus the alternative that at least one of them is false. The null hypothesis of the Ljung–Box test is  $H_0 : \rho(1) = \rho(2) = \dots = \rho(K) = 0$  for some  $K$ , say  $K = 5$  or 10. If the Ljung–Box test rejects, then we conclude that one or more of  $\rho(1), \rho(2), \dots, \rho(K)$  is nonzero.

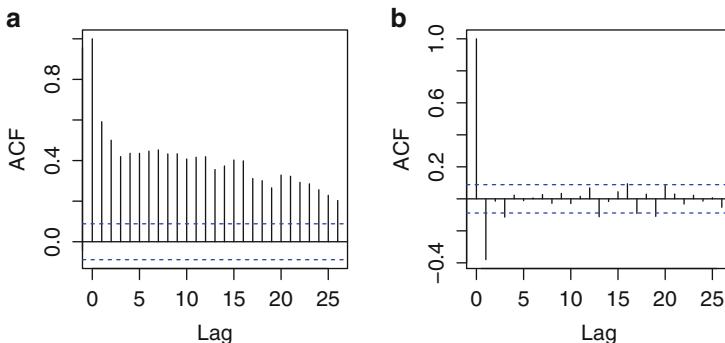
---

<sup>2</sup> Best linear prediction is discussed in Sect. 11.9.1.

If, in fact, the autocorrelations 1 to  $K$  are all zero, then there is only a 1 in 20 chance of falsely concluding that they are not all zero, assuming a level 0.05 test. In contrast, if the autocorrelations are tested one at time, then there is a much higher chance of concluding that one or more is nonzero.

The Ljung–Box test is sometimes called simply the Box test, though the former name is preferable since the test is based on a joint paper of Ljung and Box.

*Example 12.3. Inflation rates and changes in the inflation rate—sample ACF plots and the Ljung–Box test*



**Fig. 12.3.** Sample ACF plots of the one-month inflation rate (a) and changes in the inflation rate (b).

We return to the inflation rate data used in Example 12.1. Figure 12.3 contains plots of (a) the sample ACF of the one-month inflation rate and (b) the sample ACF of changes in the inflation rate.

```

1 data(Mishkin, package = "Ecdat")
2 y = as.vector(Mishkin[,1])
3 par(mfrow=c(1,2))
4 acf(y)
5 acf(diff(y))

```

In (a) we see that the sample ACF decays to zero slowly. This is a sign of either nonstationarity or possibly of stationarity with long-memory dependence, which is discussed in Sect. 13.5. In contrast, the sample ACF in (b) decays to zero quickly, indicating clearly that the differenced series is stationary. Thus, the sample ACF plots agree with the conclusions reached by examining the time series plots in Fig. 12.1, specifically that the differenced series is stationary and the original series might not be. In Sect. 12.10 we will use hypothesis testing to further address the question of whether or not the original series is stationary.

Several of the autocorrelations of the rate change series fall outside the test bounds, which suggests that the series is not white noise. To check, the Ljung–Box test was implemented using R’s `Box.test()` function.  $K$  is called `lag` when `Box.test()` is called and `df` in the output, and we specify `type="Ljung-Box"` for the Ljung–Box test.

```
6 Box.test(diff(y), lag = 10, type = "Ljung-Box")
```

The Ljung–Box test statistic with  $K = 10$  is 79.92, which has an extremely small  $p$ -value, 5.217e–13, so the null hypothesis of white noise is strongly rejected. Other choices of  $K$  give similar results.  $\square$

Although a stationary process is somewhat parsimonious with parameters, at least relative to a general nonstationary process, a stationary process is still not sufficiently parsimonious for most purposes. The problem is that there are still an infinite number of parameters,  $\rho(1), \rho(2), \dots$ . What we need is a class of stationary time series models with only a finite, preferably small, number of parameters. The ARIMA models of this chapter are precisely such a class. The simplest ARIMA models are autoregressive (AR) models, and we turn to these first.

## 12.4 AR(1) Processes

Time series models with correlation can be constructed from white noise. The simplest correlated stationary processes are *autoregressive processes*, where  $Y_t$  is modeled as a weighted average of past observations plus a white noise “error,” which is also called the “noise” or “disturbance.” We start with AR(1) processes, the simplest autoregressive processes.

Let  $\epsilon_1, \epsilon_2, \dots$  be weak WN( $0, \sigma_\epsilon^2$ ). We say that  $Y_1, Y_2, \dots$  is an *AR(1) process* if for some constant parameters  $\mu$  and  $\phi$ ,

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t \quad (12.3)$$

for all  $t$ . The parameter  $\mu$  is the mean of the process, hence  $(Y_t - \mu)$  has mean zero for all  $t$ . We may interpret the term  $\phi(Y_{t-1} - \mu)$  as representing “memory” or “feedback” of the past into the present value of the process. The process  $\{Y_t\}_{t=-\infty}^{+\infty}$  is correlated because the deviation of  $Y_{t-1}$  from its mean is fed back into  $Y_t$ . The parameter  $\phi$  determines the amount of feedback, with a larger absolute value of  $\phi$  resulting in more feedback and  $\phi = 0$  implying that  $Y_t = \mu + \epsilon_t$ , so that  $Y_t$  is weak WN( $\mu, \sigma_\epsilon^2$ ). In applications in finance, one can think of  $\epsilon_t$  as representing the effect of “new information.” For example, if  $Y_t$  is the log return on an asset at time  $t$ , then  $\epsilon_t$  represents the effect on the asset’s price of business and economic information that is revealed at time  $t$ . Information that is truly new cannot be anticipated, so the effects of today’s new information should be independent of the effects of yesterday’s news. This is why we model new information as white noise.

If  $Y_1, Y_2, \dots$  is a weakly stationary process, then  $|\phi| < 1$ . To see this, note that stationarity implies that the variances of  $(Y_t - \mu)$  and  $(Y_{t-1} - \mu)$  in (12.3) are equal, say, to  $\sigma_Y^2$ . Therefore,  $\sigma_Y^2 = \phi^2 \sigma_Y^2 + \sigma_\epsilon^2$ , which requires that  $|\phi| < 1$ . The mean of this process is  $\mu$ . Simple algebra shows that (12.3) can be rewritten as

$$Y_t = (1 - \phi)\mu + \phi Y_{t-1} + \epsilon_t. \quad (12.4)$$

Recall the linear regression model  $Y_t = \beta_0 + \beta_1 X_t + \epsilon_t$  from your statistics courses or see Chap. 9 for an introduction to regression analysis. Equation (12.4) is just a linear regression model with intercept  $\beta_0 = (1 - \phi)\mu$  and slope  $\beta_1 = \phi$ . The term *autoregression* refers to the regression of the process on its own past values.

If  $|\phi| < 1$ , then repeated substitution of (12.3) shows that

$$Y_t = \mu + \epsilon_t + \phi \epsilon_{t-1} + \phi^2 \epsilon_{t-2} + \dots = \mu + \sum_{h=0}^{\infty} \phi^h \epsilon_{t-h}, \quad (12.5)$$

assuming that time index  $t$  of  $Y_t$  and  $\epsilon_t$  can be extended to negative values so that the white noise process is  $\{\epsilon_t\}_{t=-\infty}^{+\infty}$  and (12.3) is true for all integers  $t$ . Equation (12.5) is called *the infinite moving average* [MA( $\infty$ )] representation of the process. This equation shows that  $Y_t$  is a weighted average of *all* past values of the white noise process. This representation should be compared to the AR(1) representation that shows  $Y_t$  as depending only on  $Y_{t-1}$  and  $\epsilon_t$ . Since  $|\phi| < 1$ ,  $\phi^h \rightarrow 0$  as the lag  $h \rightarrow \infty$ . Thus, the weights given to the distant past are small. In fact, they are quite small. For example, if  $\phi = 0.5$ , then  $\phi^{10} = 0.00098$ , so  $\epsilon_{t-10}$  has virtually no effect on  $Y_t$ . For this reason, the sum in (12.5) could be truncated at a finite number of terms, with no practical need to assume that the processes existed in the infinite past.

### 12.4.1 Properties of a Stationary AR(1) Process

When an AR(1) process is weakly stationary, which implies that  $|\phi| < 1$ , then

$$E(Y_t) = \mu \quad \forall t, \quad (12.6)$$

$$\text{Var}(Y_t) = \gamma(0) = \sigma_Y^2 = \frac{\sigma_\epsilon^2}{1 - \phi^2} \quad \forall t, \quad (12.7)$$

$$\text{Cov}(Y_t, Y_{t+h}) = \gamma(h) = \phi^{|h|} \frac{\sigma_\epsilon^2}{1 - \phi^2} \quad \forall t \text{ and } \forall h, \text{ and} \quad (12.8)$$

$$\text{Corr}(Y_t, Y_{t+h}) = \rho(h) = \phi^{|h|} \quad \forall t \text{ and } \forall h. \quad (12.9)$$

It is important to remember that formulas (12.6) to (12.9) hold only if  $|\phi| < 1$  and only for AR(1) processes. Moreover, for  $Y_t$  to be stationary,  $Y_0$  must start in the stationary distribution so that  $E(Y_0) = \mu$  and  $\text{Var}(Y_0) = \sigma_\epsilon^2 / (1 - \phi^2)$ . Otherwise,  $Y_t$  is not stationary though it eventually converges to stationarity.

These formulas can be proved using (12.5). For example, using (7.11) in Sect. 7.3.2,

$$\text{Var}(Y_t) = \text{Var}\left(\sum_{h=0}^{\infty} \phi^h \epsilon_{t-h}\right) = \sigma_\epsilon^2 \sum_{h=0}^{\infty} \phi^{2h} = \frac{\sigma_\epsilon^2}{1-\phi^2}, \quad (12.10)$$

which proves (12.7). In (12.10) the formula for summation of a geometric series was used. This formula is

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r} \quad \text{if } |r| < 1. \quad (12.11)$$

Also, for  $h > 0$ ,

$$\text{Cov}\left(\sum_{i=0}^{\infty} \epsilon_{t-i} \phi^i, \sum_{j=0}^{\infty} \epsilon_{t+h-j} \phi^j\right) = \phi^{|h|} \frac{\sigma_\epsilon^2}{1-\phi^2}, \quad (12.12)$$

thus verifying (12.8). Then (12.9) follows by dividing (12.8) by (12.7).

Be sure to distinguish between  $\sigma_\epsilon^2$ , which is the variance of the white noise process  $\epsilon_1, \epsilon_2, \dots$ , and  $\gamma(0)$ , which is the variance,  $\sigma_Y^2$ , of the stationary AR(1) process  $Y_1, Y_2, \dots$ . We can see from (12.7) that  $\gamma(0)$  is larger than  $\sigma_\epsilon^2$  unless  $\phi = 0$ , in which case  $Y_t = \mu + \epsilon_t$ , such that  $Y_t$  and  $\epsilon_t$  have the same variance.

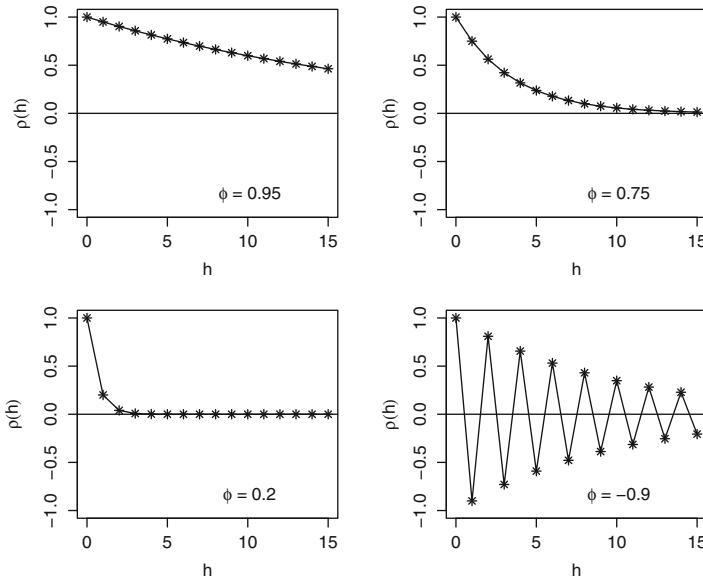
The ACF (autocorrelation function) of an AR(1) process depends upon only one parameter,  $\phi$ . This is a remarkable amount of parsimony, but it comes at a price. The ACF of an AR(1) process has only a limited range of shapes, as can be seen in Fig. 12.4. The magnitude of its ACF decays geometrically to zero, either slowly as when  $\phi = 0.95$ , moderately slowly as when  $\phi = 0.75$ , or rapidly as when  $\phi = 0.2$ . If  $\phi < 0$ , then the sign of the ACF alternates as its magnitude decays geometrically. If the sample ACF of the data does not behave in one of these ways, then an AR(1) model is unsuitable. The remedy is to use more AR parameters, or to switch to another class of models such as the moving average (MA) or autoregressive moving average (ARMA) models. We investigate these alternatives in this chapter.

### 12.4.2 Convergence to the Stationary Distribution

Suppose that  $Y_0$  is an arbitrary starting value not chosen from the stationary distribution and that (12.3) holds for  $t = 1, 2, \dots$ . Then the process is not stationary, but converges to the stationary distribution satisfying (12.6) to (12.9) as  $t \rightarrow \infty$ .<sup>3</sup> For example, since  $Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t$ , we have

---

<sup>3</sup> However, there is a technical issue here. It must be assumed that  $Y_0$  has a finite mean and variance, since otherwise  $Y_t$  will not have a finite mean and variance for any  $t > 0$ .



**Fig. 12.4.** Autocorrelation functions of AR(1) processes with  $\phi$  equal to 0.95, 0.75, 0.2, and -0.9.

$E(Y_1) - \mu = \phi\{E(Y_0) - \mu\}$ , and  $E(Y_2) - \mu = \phi^2\{E(Y_0) - \mu\}$ , and so forth, so that

$$E(Y_t) = \mu + \phi^t\{E(Y_0) - \mu\} \text{ for all } t > 0. \quad (12.13)$$

Since  $|\phi| < 1$ ,  $\phi^t \rightarrow 0$  and  $E(Y_t) \rightarrow \mu$  as  $t \rightarrow \infty$ . The convergence of  $\text{Var}(Y_t)$  to  $\sigma_\epsilon^2/(1-\phi^2)$  can be proved in a somewhat similar manner. The convergence to the stationary distribution can be very rapid when  $|\phi|$  is not too close to 1. For example, if  $\phi = 0.5$ , then  $\phi^{10} = 0.00097$ , so by (12.13),  $E(Y_{10})$  is very close to  $\mu$  unless  $E(Y_0)$  was extremely far from  $\mu$ .

### 12.4.3 Nonstationary AR(1) Processes

If  $|\phi| \geq 1$ , then the AR(1) process is nonstationary, and the mean, variance, covariances and correlations are not constant.

#### Random Walk ( $\phi = 1$ )

If  $\phi = 1$ , then

$$Y_t = Y_{t-1} + \epsilon_t$$

and the process is *not* stationary. This is the random walk process we saw in Chap. 2.

Suppose we start the process at an arbitrary point  $Y_0$ . It is easy to see that

$$Y_t = Y_0 + \epsilon_1 + \cdots + \epsilon_t.$$

Then  $E(Y_t|Y_0) = Y_0$  for all  $t$ , which is constant but depends entirely on the arbitrary starting point. Moreover,  $\text{Var}(Y_t|Y_0) = t\sigma_\epsilon^2$ , which is not stationary but rather increases linearly with time. The increasing variance makes the random walk “wander” in that  $Y_t$  takes increasingly longer excursions away from its conditional mean of  $Y_0$  and therefore is not mean-reverting.

### AR(1) Processes When $|\phi| > 1$

When  $|\phi| > 1$ , an AR(1) process has explosive behavior. This can be seen in Fig. 12.5. This figure shows simulations of 200 observations from AR(1) processes with various values of  $\phi$ . The explosive case where  $\phi = 1.01$  clearly is different from the other cases where  $|\phi| \leq 1$ . However, the case where  $\phi = 1$  is not that much different from  $\phi = 0.98$  even though the former is nonstationary while the latter is stationary. Longer time series would help distinguish between  $\phi = 0.98$  and  $\phi = 1$ .

## 12.5 Estimation of AR(1) Processes

R has the function `arima()` for fitting AR and other time series models. The function `arima()` and similar functions in other software packages have two primary estimation methods, conditional least-squares and maximum likelihood. The two methods are explained in Sect. 12.5.2. They are similar and generally give nearly the same estimates. In this book, we use the default method in R’s `arima()`, which is the MLE with the conditional least-squares estimate used as the starting value for computing the MLE by iterative nonlinear optimization.

### 12.5.1 Residuals and Model Checking

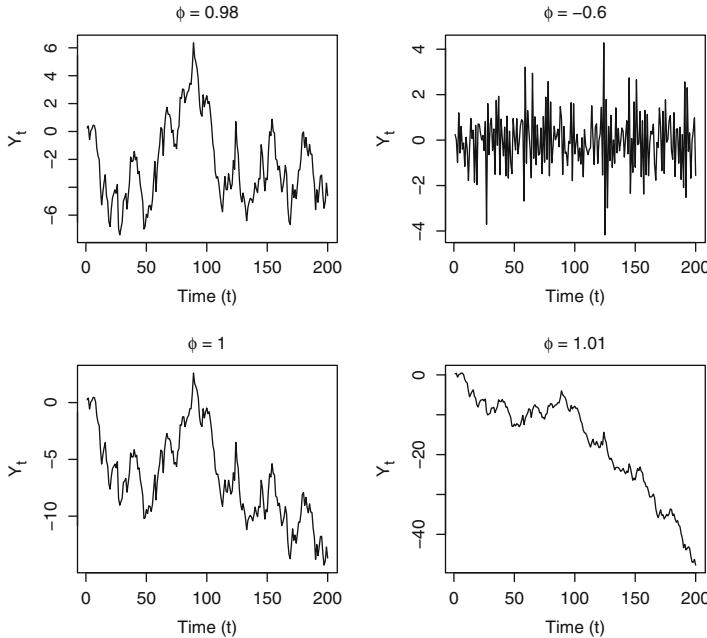
Once  $\mu$  and  $\phi$  have been estimated, one can estimate the white noise process  $\epsilon_1, \dots, \epsilon_n$ . Rearranging Eq. (12.3), we have

$$\epsilon_t = (Y_t - \mu) - \phi(Y_{t-1} - \mu). \quad (12.14)$$

In analogy with (12.14), the residuals,  $\hat{\epsilon}_2, \hat{\epsilon}_3, \dots, \hat{\epsilon}_n$ , are defined as

$$\hat{\epsilon}_t = (Y_t - \hat{\mu}) - \hat{\phi}(Y_{t-1} - \hat{\mu}), \quad t \geq 2, \quad (12.15)$$

and estimate  $\epsilon_2, \dots, \epsilon_n$ . The first noise,  $\epsilon_1$ , cannot be estimated directly since it is assumed that the observations start at  $Y_1$  so that  $Y_0$  is not available.



**Fig. 12.5.** Simulations of 200 observations from AR(1) processes with various values of  $\phi$  and  $\mu = 0$ . The white noise process  $\epsilon_1, \epsilon_2, \dots, \epsilon_{200}$  is the same for all four AR(1) processes.

The residuals can be used to check the assumption that  $Y_1, Y_2, \dots, Y_n$  is an AR(1) process; any autocorrelation in the residuals is evidence against the assumption of an AR(1) process.

To appreciate why residual autocorrelation indicates a possible problem with the model, suppose that we are fitting an AR(1) model,  $Y_t = \mu + \phi(Y_{t-1} - \mu) + \epsilon_t$ , but the true model is an AR(2) process<sup>4</sup> given by

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t.$$

Since we are fitting the incorrect AR(1) model, there is no hope of estimating  $\phi_2$  since it is not in the model. Moreover,  $\hat{\phi}$  does not necessarily estimate  $\phi_1$  because of bias caused by model misspecification. Let  $\phi^*$  be the expected value of  $\hat{\phi}$ . For the purpose of illustration, assume that  $\hat{\mu} \approx \mu$  and  $\hat{\phi} \approx \phi^*$ . This is a sensible approximation if the sample size  $n$  is large enough. Then

$$\begin{aligned}\hat{\epsilon}_t &\approx (Y_t - \mu) - \phi^*(Y_{t-1} - \mu) \\ &= \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t - \phi^*(Y_{t-1} - \mu) \\ &= (\phi_1 - \phi^*)(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t.\end{aligned}$$

---

<sup>4</sup> We discuss higher-order AR models in more detail soon.

Thus, the residuals do not estimate the white noise process as they would if the correct AR(2) model were used. Even if there is no bias in the estimation of  $\phi_1$  by  $\hat{\phi}$  so that  $\phi_1 = \phi^*$  and the term  $(\phi_1 - \phi^*)(Y_{t-1} - \mu)$  drops out, the presence of  $\phi_2(Y_{t-2} - \mu)$  in the residuals causes them to be autocorrelated.

To check for residual autocorrelation, one can use the *test bounds* of ACF plots. Any residual ACF value outside the test bounds is significantly different from 0 at the 0.05 level. As discussed earlier, the danger here is that some sample ACF values will be significant merely by chance, and to guard against this danger, one can use the Ljung–Box test that *simultaneously* tests that all autocorrelations up to a specified lag are zero. When the Ljung–Box test is applied to residuals, a correction is needed to account for the use of  $\hat{\phi}$  in place of the unknown  $\phi$ . Some software makes this correction automatically. In R the correction is not automatic but is done by setting the `fitdf` parameter in `Box.test()` to the number of autoregressive coefficient parameters that were estimated, so for an AR(1) model `fitdf` should be 1.

*Example 12.4. Daily log returns for BMW stock—ACF plots and AR fit*

The daily log returns for BMW stock between January 1973 and July 1996 from the `bmw` data set in R’s `evir` package are shown in Fig. 12.6a. Their sample ACF and quantiles are shown in Fig. 12.6b and c, respectively. The estimated autocorrelation coefficient at lag 1 is well outside the test bounds, so the series has some dependence. Also, the Ljung–Box test that the first `lag` autocorrelations are 0 was performed using R’s `Box.test()` function.

```
7 data(bmw, package = "evir")
8 Box.test(bmw, lag = 5, type = "Ljung-Box")
```

The parameter `lag`, which specifies the number of autocorrelation coefficients to test, was set equal to 5, though other choices give similar results. The output was

**Box-Ljung test**

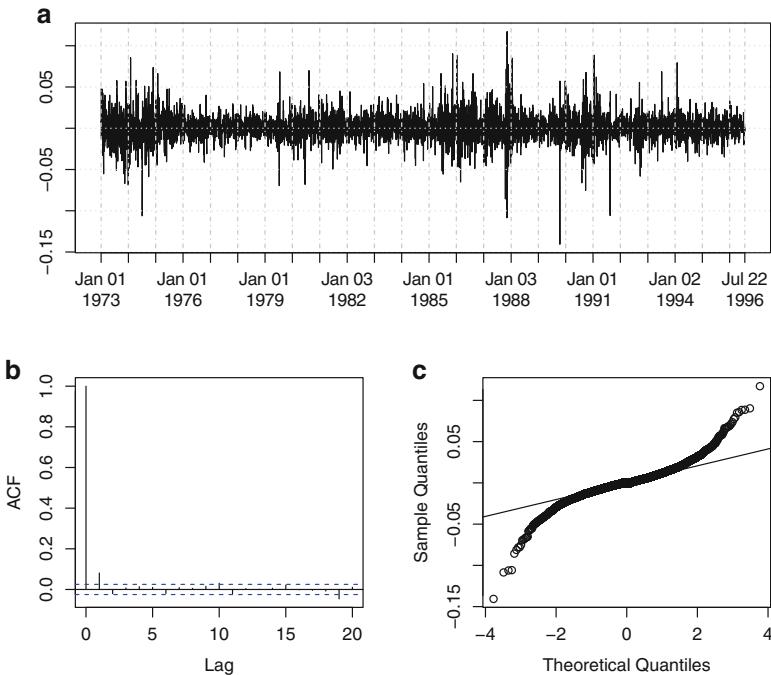
```
data: bmw
X-squared = 44.987, df = 5, p-value = 1.460e-08
```

The *p*-value is very small, indicating that at least one of the first five autocorrelations is nonzero. Whether the amount of dependence is of any practical importance is debatable, but an AR(1) model to account for the small amount of autocorrelation might be appropriate.

Next, an AR(1) model was fit using the `arima()` command in R.

```
9 fitAR1 = arima(bmw, order = c(1,0,0))
10 print(fitAR1)
```

The `order` parameter will be explained later, but for an AR(1) process it should be `c(1,0,0)`. A summary of the output is below.



**Fig. 12.6.** (a) Daily log returns for BMW stock from January 1973 until July 1996, and their (b) sample ACF and (c) sample quantiles relative to the normal distribution.

Call:

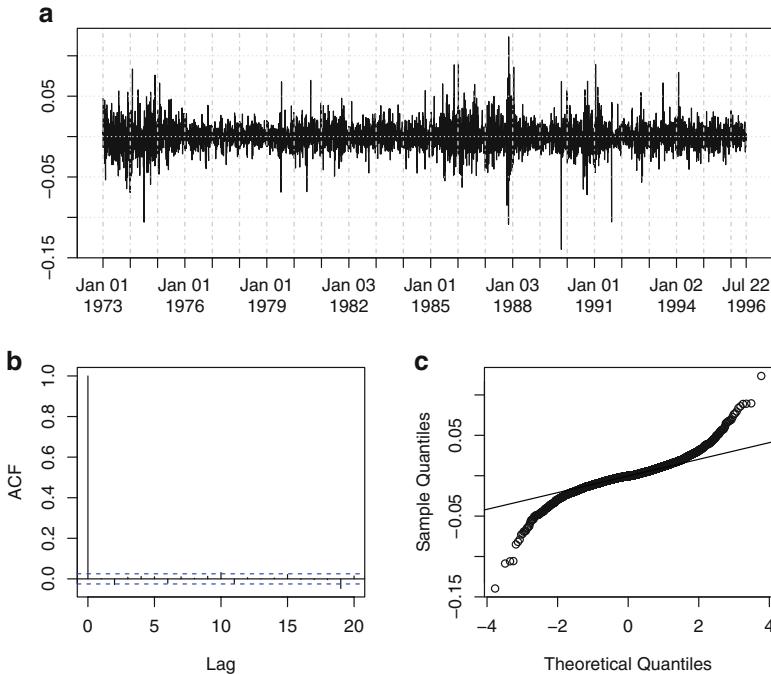
```
arima(x = bmw, order = c(1, 0, 0))
```

Coefficients:

|          |           |
|----------|-----------|
| ar1      | intercept |
| 0.081116 | 0.000340  |
| s.e.     | 0.012722  |
|          | 0.000205  |

```
sigma^2 estimated as 0.000216260: log-likelihood = 17212.34,
aic = -34418.68
```

We see that  $\hat{\phi} = 0.081$  and  $\hat{\sigma}^2 = 0.00022$ . Although  $\hat{\phi}$  is small, it is statistically significant since it is 6.4 times its standard error 0.013, so its *p*-value is near zero. As just mentioned, whether this small, but nonzero, value of  $\hat{\phi}$  is of practical significance is another matter. A non-zero value of  $\phi$  means that there is some information in today's return that could be used for prediction of tomorrow's return, but a small value of  $\phi$  means that the prediction will not be very accurate. The potential for profit might be negated by trading costs.



**Fig. 12.7.** A (a) time series plot, (b) sample ACF and (c) normal quantile plot of residuals from an AR(1) fit to the daily log returns for BMW stock.

The sample ACF of the residuals is plotted in Fig. 12.7b. None of the autocorrelations at low lags are outside the test bounds. A few at higher lags are outside the bounds, but this type of behavior is expected to occur by chance or because, with a large sample size, very small but nonzero true correlations can be detected. The Ljung–Box test was applied, with `lag` equal to 5 and `fitdf=1`.

```
11 Box.test(residuals(fitAR1), lag = 5, type = "Ljung-Box", fitdf = 1)
```

Box-Ljung test

```
data: residuals(fitAR1)
X-squared = 6.8669, df = 4, p-value = 0.1431
```

The large  $p$ -value indicates that we should accept the null hypothesis that the residuals are uncorrelated, at least at small lags. This is a sign that the AR(1) model provides an adequate fit. However, the Ljung–Box test was repeated with `lag` equal to 10, 15, and 20 and the  $p$ -values were 0.041, 0.045, and 0.004, respectively. These values are “statistically significant” using the conventional

cutoff of 0.05. The sample size is 6146, so it is not surprising that even a small amount of autocorrelation can be statistically significant. The practical significance of this autocorrelation is very doubtful.

We conclude that the AR(1) model is adequate for the BMW daily returns, but at longer lags some slight amount of autocorrelation appears to remain. However, the time series plot and normal quantile plot of the AR(1) residuals in Fig. 12.7a and c show volatility clustering and heavy tails. These are common features of economic data and will be modeled in subsequent chapters.  $\square$

### *Example 12.5. Inflation rate—AR(1) fit and checking residuals*

This example uses the inflation rate time series used earlier in Example 12.1. Although there is some doubt as to whether this series is stationary, we will fit an AR(1) model. The ACF of the residuals are shown in Fig. 12.8 and there is considerable residual autocorrelation, which indicates that the AR(1) model is not adequate. A Ljung–Box test confirms this result.

```
12 data(Mishkin, package = "Ecdat")
13 y = as.vector(Mishkin[,1])
14 fit = arima(y, order = c(1,0,0))
15 Box.test(fit$resid, type = "Ljung", lag = 24, fitdf = 1)

Box-Ljung test

data: fit$resid
X-squared = 138.5776, df = 23, p-value < 2.2e-16
```

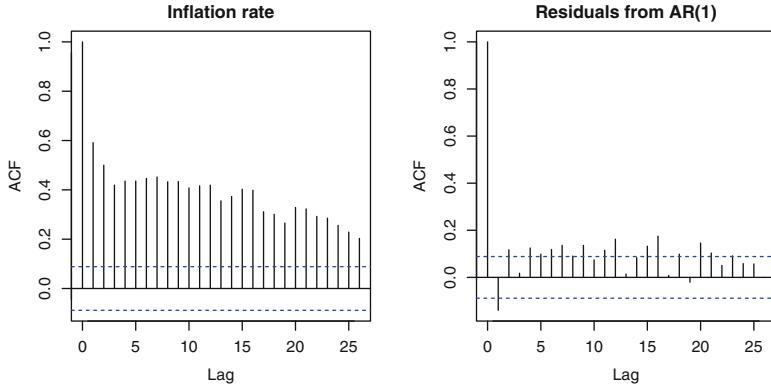
One might try fitting an AR(1) to the changes in the inflation rate, since this series is clearly stationary. However, the AR(1) model also does not fit the changes in the inflation rate. We will return to this example when we have a larger collection of models in our statistics toolbox.  $\square$

### 12.5.2 Maximum Likelihood and Conditional Least-Squares

Estimators for AR processes can be computed automatically by most statistical software packages, and the user need not know what is “under the hood” of the software. Nonetheless, for readers interested in the estimation methodology, this section has been provided.

To find the joint density for  $Y_1, \dots, Y_n$ , we use (A.41) and the fact that

$$f_{Y_t|Y_1, \dots, Y_{t-1}}(y_t|y_1, \dots, y_{t-1}) = f_{Y_t|Y_{t-1}}(y_t|y_{t-1}) \quad (12.16)$$



**Fig. 12.8.** Sample ACF for the inflation rate time series and residual series from an AR(1) fit.

for  $t = 2, 3, \dots, n$ . A stochastic process with property (12.16) is called a *Markov process*. By (A.41) and (12.16), we have

$$f_{Y_1, \dots, Y_n}(y_1, \dots, y_n) = f_{Y_1}(y_1) \prod_{t=2}^n f_{Y_t|Y_{t-1}}(y_t|y_{t-1}). \quad (12.17)$$

If we assume the errors are from a Gaussian white noise process, then by (12.6) and (12.7), we know that  $Y_1$  is  $N\{\mu, \sigma_\epsilon^2/(1-\phi^2)\}$  for a stationary AR(1) process. Given  $Y_{t-1}$ , the only random component of  $Y_t$  is  $\epsilon_t$ , so that  $Y_t$  given  $Y_{t-1}$  is  $N\{\mu + \phi(Y_{t-1} - \mu), \sigma_\epsilon^2\}$ . It then follows that the joint density for  $Y_1, \dots, Y_n$  is

$$\left(\frac{1}{\sqrt{2\pi}\sigma_\epsilon}\right)^n \sqrt{1-\phi^2} \exp\left\{-\frac{(Y_1 - \mu)^2}{2\sigma_\epsilon^2/(1-\phi^2)}\right\} \prod_{i=2}^n \exp\left(-\frac{\left[Y_i - \{\mu + \phi(Y_{i-1} - \mu)\}\right]^2}{2\sigma_\epsilon^2}\right). \quad (12.18)$$

The maximum likelihood estimator maximizes the logarithm of (12.18) over  $(\mu, \phi, \sigma_\epsilon)$ . A somewhat simpler estimator deletes the marginal density of  $Y_1$  from the likelihood and maximizes the logarithm of

$$\left(\frac{1}{\sqrt{2\pi}\sigma_\epsilon}\right)^{n-1} \prod_{t=2}^n \exp\left(-\frac{\left[Y_t - \{\mu + \phi(Y_{t-1} - \mu)\}\right]^2}{2\sigma_\epsilon^2}\right). \quad (12.19)$$

This estimator is called the conditional least-squares estimator. It is “conditional” because it uses the conditional density of  $Y_2, \dots, Y_n$  given  $Y_1$ . It is a least-squares estimator because the estimates of  $\mu$  and  $\phi$  minimize

$$\sum_{t=2}^n \left[ Y_t - \{\mu + \phi(Y_{t-1} - \mu)\} \right]^2. \quad (12.20)$$

The default method for the function `arima()` in R is to use the conditional least-squares estimates as starting values for maximum likelihood. The MLE is returned, along with approximate standard errors. The default option is used in the examples in this book.

## 12.6 AR( $p$ ) Models

We have seen that the ACF of an AR(1) process decays geometrically to zero if  $|\phi| < 1$  and also alternates in sign if  $\phi < 0$ . This is a limited range of behavior and many time series do not behave in this way. To get a more flexible class of models, but one that is still parsimonious, we can use a model that regresses the current value of the process on several of the recent past values, not just the most recent. Thus, we let the last  $p$  values of the process,  $Y_{t-1}, \dots, Y_{t-p}$ , feed back into the current value  $Y_t$ .

Here is a formal definition. The stochastic process  $Y_t$  is an *AR( $p$ ) process* if

$$Y_t - \mu = \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) + \epsilon_t,$$

where  $\epsilon_1, \epsilon_2, \dots$  is weak WN( $0, \sigma_\epsilon^2$ ).

This is a multiple linear regression<sup>5</sup> model with lagged values of the time series as the “ $x$ -variables.” The model can also be expressed as

$$Y_t = \beta_0 + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t,$$

where  $\beta_0 = \{1 - (\phi_1 + \cdots + \phi_p)\}\mu$ . The parameter  $\beta_0$  is called the “constant” or “intercept” as in an AR(1) model. It can be shown that  $\{1 - (\phi_1 + \cdots + \phi_p)\} > 0$  for a stationary process, so  $\mu = 0$  if and only if  $\beta_0$  is zero.

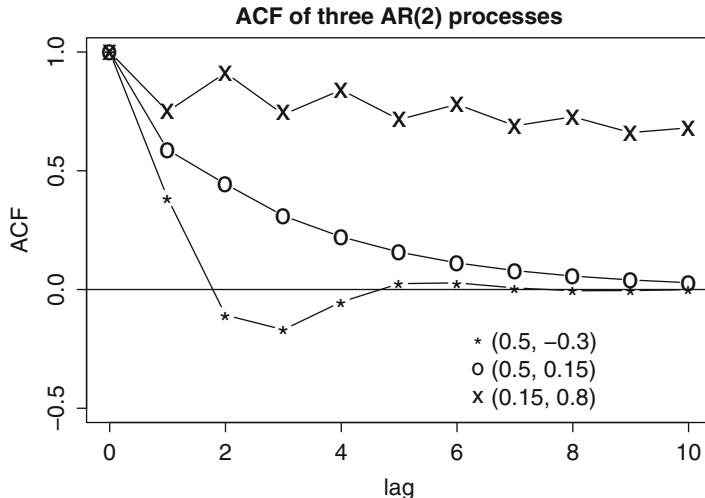
Formulas for the ACFs of AR( $p$ ) processes with  $p > 1$  are more complicated than for an AR(1) process and can be found in the time series textbooks listed in the “References” section. However, software is available for computing and plotting the ACF of any AR processes, as well as for the MA and ARMA processes to be introduced soon. Figure 12.9 is a plot of the ACFs of three AR(2) process. The ACFs were computed using R’s `ARMAacf()` function. Notice the wide variety of ACFs that are possible with two AR parameters.

Most of the concepts we have discussed for AR(1) models generalize easily to AR( $p$ ) models. The conditional least squares or maximum likelihood estimators can be calculated using software such as R’s `arima()` function. The residuals are defined by

$$\hat{\epsilon}_t = Y_t - \{\hat{\beta}_0 + \hat{\phi}_1 Y_{t-1} + \cdots + \hat{\phi}_{t-p} Y_{t-p}\}, \quad t \geq p+1.$$

---

<sup>5</sup> See Chap. 9 for an introduction to multiple regression.



**Fig. 12.9.** ACF of three AR(2) processes; the legend gives the values of  $\phi_1$  and  $\phi_2$ .

If the AR( $p$ ) model fits the time series well, then the residuals should look like white noise. Residual autocorrelation can be detected by examining the sample ACF of the residuals and using the Ljung–Box test. Any significant residual autocorrelation is a sign that the AR( $p$ ) model does not fit well.

One problem with AR models is that they often need a rather large value of  $p$  to fit a data set. The problem is illustrated by the following two examples.

*Example 12.6. Changes in the inflation rate—AR( $p$ ) models*

Figure 12.10 is a plot of AIC and BIC versus  $p$  for AR( $p$ ) fits to the changes in the inflation rate. Both criteria suggest that  $p$  should be large. AIC decreases steadily as  $p$  increases from 1 to 19, though there is a local minimum at 8. Even the conservative BIC criterion indicates that  $p$  should be as large as 6. Thus, AR models are not parsimonious for this example. The remedy is to use a MA or ARMA model, which are the topics of the next sections.

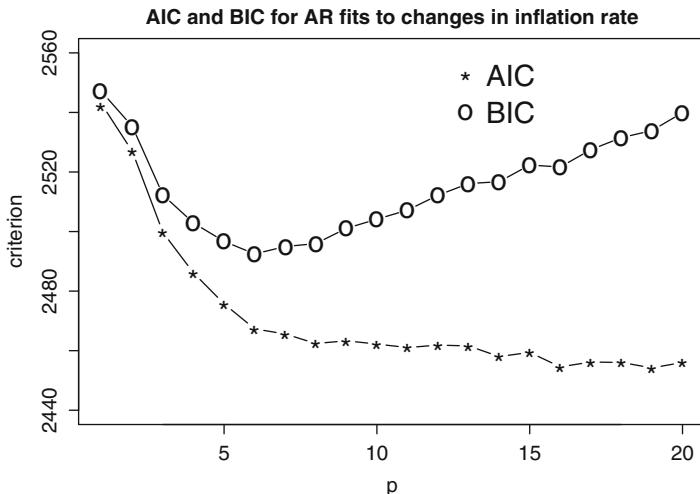
Many statistical software packages have functions to automate the search for the AR model that optimizes AIC or other criteria. The `auto.arima` function in R's `forecast` package found that  $p = 8$  is the first local minimum of AIC.

```

16 library(forecast)
17 auto.arima(diff(y), max.p = 20, max.q = 0, d = 0, ic = "aic")

Series: diff(y)
ARIMA(8,0,0) with zero mean

```



**Fig. 12.10.** Fitting AR( $p$ ) models to changes in the one-month inflation rate; AIC and BIC plotted against  $p$ .

Coefficients:

|      | ar1     | ar2     | ar3     | ar4     | ar5     | ar6     |
|------|---------|---------|---------|---------|---------|---------|
|      | -0.6274 | -0.4977 | -0.5158 | -0.4155 | -0.3443 | -0.2560 |
| s.e. | 0.0456  | 0.0536  | 0.0576  | 0.0606  | 0.0610  | 0.0581  |
|      | ar7     | ar8     |         |         |         |         |
|      | -0.1557 | -0.1051 |         |         |         |         |
| s.e. | 0.0543  | 0.0459  |         |         |         |         |

sigma^2 estimated as 8.539: log likelihood=-1221.2

AIC=2460.4 AICc=2460.78 BIC=2498.15

The first local minimum of BIC is at  $p = 6$ .

```
18 auto.arima(diff(y), max.p = 20, max.q = 0, d = 0, ic = "bic")
```

Series: diff(y)

ARIMA(6,0,0) with zero mean

Coefficients:

|      | ar1     | ar2     | ar3     | ar4     | ar5     | ar6     |
|------|---------|---------|---------|---------|---------|---------|
|      | -0.6057 | -0.4554 | -0.4558 | -0.3345 | -0.2496 | -0.1481 |
| s.e. | 0.0454  | 0.0522  | 0.0544  | 0.0546  | 0.0526  | 0.0457  |

sigma^2 estimated as 8.699: log likelihood=-1225.67

AIC=2465.33 AICc=2465.56 BIC=2494.69

We will see later that a more parsimonious fit can be obtained by going beyond AR models.  $\square$

*Example 12.7. Inflation rates—AR( $p$ ) models*

Since it is uncertain whether or not the inflation rates are stationary, one might fit an AR model to the inflation rates themselves, rather than their differences. An AR( $p$ ) models was fit to the inflation rates with  $p$  selected via an information criterion by `auto.arima()`. The BIC method chose  $p = 2$  and AIC selected  $p = 7$ . The results for  $p = 7$  are below.

```
Series: y
ARIMA(7,0,0) with non-zero mean

Coefficients:
 ar1 ar2 ar3 ar4 ar5 ar6 ar7 intercept
 0.366 0.129 -0.020 0.099 0.065 0.080 0.119 3.987
 s.e. 0.045 0.048 0.048 0.048 0.049 0.048 0.046 0.784

sigma^2 estimated as 8.47: log likelihood=-1221.8
AIC=2461.6 AICc=2461.9 BIC=2499.3
```

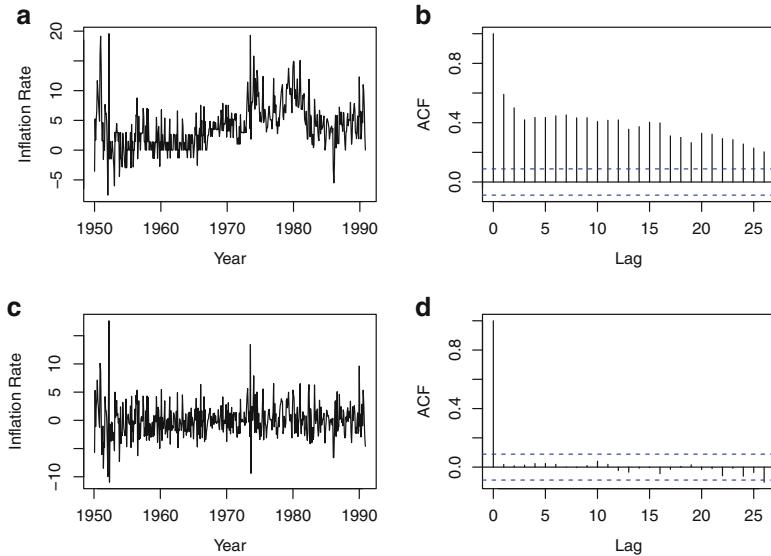
The inflation rate and its residual series from an AR(7) fit and their sample ACFs are shown in Fig. 12.11.  $\square$

## 12.7 Moving Average (MA) Processes

As we saw in Example 12.6, there is a potential need for large values of  $p$  when fitting AR processes. A remedy for this problem is to add a moving average component to an AR( $p$ ) process. The result is an *autoregressive-moving average process*, often called an *ARMA process*. Before introducing ARMA processes, we start with pure moving average (MA) processes.

### 12.7.1 MA(1) Processes

The idea behind AR processes is to feed past data back into the current value of the process. This induces correlation between the past and present. The effect is to have at least some correlation at *all* lags. Sometimes data show correlation at only short lags, for example, only at lag 1 or only at lags 1 and 2. See, for example, Fig. 12.3b where the sample ACF of changes in the inflation rate is approximately  $-0.4$  at lag 1, but then is approximately 0.1 or less in magnitude after one lag. AR processes do not behave this way and, as already seen in Example 12.6, do not provide a parsimonious fit. In such situations, a useful alternative to an AR model is a moving average (MA) model. A process  $Y_t$  is a *moving average process* if  $Y_t$  can be expressed as a weighted average (moving average) of the past values of the white noise process  $\{\epsilon_t\}$ .



**Fig. 12.11.** (a) The inflation rate series and (b) its sample ACF; (c) the residuals series from an AR(7) fit to the inflation rates (d) ACF of residuals.

The **MA(1)** (moving average of order 1) process is

$$Y_t - \mu = \epsilon_t + \theta\epsilon_{t-1}, \quad (12.21)$$

where as before the  $\epsilon_t$  are weak  $WN(0, \sigma_\epsilon^2)$ .<sup>6</sup>

One can show that

$$E(Y_t) = \mu, \quad \text{Var}(Y_t) = \sigma_\epsilon^2(1 + \theta^2),$$

$$\gamma(1) = \theta\sigma_\epsilon^2,$$

$$\gamma(h) = 0 \text{ if } |h| > 1,$$

$$\rho(1) = \frac{\theta}{1 + \theta^2}, \quad (12.22)$$

$$\rho(h) = 0 \text{ if } |h| > 1. \quad (12.23)$$

Notice the implication of (12.22) and (12.23)—an MA(1) model has zero correlation at all lags except lag 1 (and of course lag 0). It is relatively easy to derive these formulas and this is left as an exercise for the reader.

<sup>6</sup> Some textbooks and some software write MA models with the signs reversed so that model (12.21) is written as  $Y_t - \mu = \epsilon_t - \theta\epsilon_{t-1}$ . We have adopted the same form of MA models as R's `arima()` function. These remarks apply as well to the general MA and ARMA models given by Eqs. (12.24) and (12.25).

### 12.7.2 General MA Processes

The **MA**( $q$ ) process is

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}. \quad (12.24)$$

One can show that  $\gamma(h) = 0$  and  $\rho(h) = 0$  if  $|h| > q$ . Formulas for  $\gamma(h)$  and  $\rho(h)$  when  $|h| \leq q$  are given in time series textbooks and these functions can be computed in R by the function **ARMAacf()**.

Unlike AR( $p$ ) models where the “constant” in the model is not the same as the mean, in an MA( $q$ ) model  $\mu$ , the mean of the process, is the same as  $\beta_0$ , the “constant” in the model. This fact can be appreciated by examining the right-hand side of Eq. (12.24), where  $\mu$  is the “intercept” or “constant” in the model and is also the mean of  $Y_t$  because  $\epsilon_t, \dots, \epsilon_{t-q}$  have mean zero. MA( $q$ ) models can be fit easily using, for example, the **arima()** function in R.

*Example 12.8. Changes in the inflation rate—MA models*

MA( $q$ ) models were fit to the changes in the inflation rate. Figure 12.12 shows plots of AIC and BIC versus  $q$ . BIC suggests that an MA(2) model is adequate, while AIC suggests an MA(3) model. We fit the MA(3) model. The Ljung–Box test was applied to the residuals with **fitdf** = 3 and **lag** equal to 5, 10, and 15 and gave  $p$ -values of 0.65, 0.76, and 0.32, respectively. The MA(2) also provided an adequate fit with the  $p$ -values from the Ljung–Box test all above 0.08. The output for the MA(3) model is below.

```

19 fitMA3 = arima(diff(y), order = c(0,0,3))
20 fitMA3

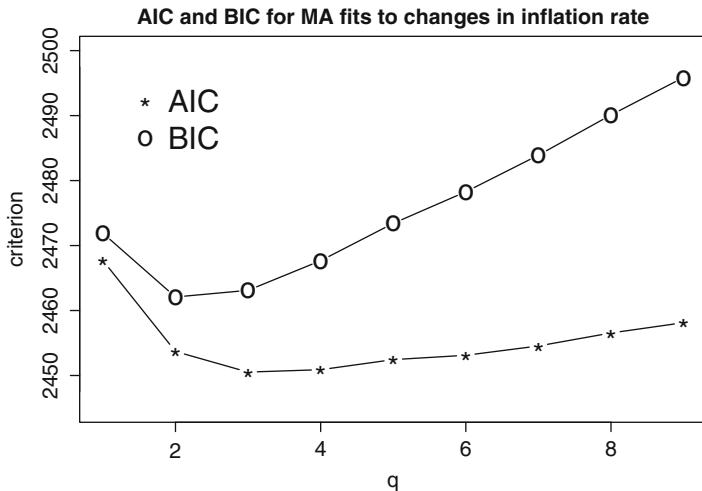
Series: diff(y)
ARIMA(0,0,3) with non-zero mean

Coefficients:
 ma1 ma2 ma3 intercept
 -0.633 -0.103 -0.108 0.000
 s.e. 0.046 0.051 0.047 0.021

sigma^2 estimated as 8.5: log likelihood=-1220.3
AIC=2450.5 AICc=2450.7 BIC=2471.5

```

Thus, if an MA model is used, then only two or three MA parameters are needed. This is a strong contrast with AR models, which require far more parameters, perhaps as many as six.  $\square$



**Fig. 12.12.** Fitting  $MA(q)$  models to changes in the one-month inflation rate; AIC and BIC plotted against  $q$ .

## 12.8 ARMA Processes

Stationary time series with complex autocorrelation behavior often are more parsimoniously modeled by mixed autoregressive and moving average (ARMA) processes than by either a pure AR or pure MA process. For example, it is sometimes the case that a model with one AR and one MA parameter, called an ARMA(1, 1) model, will provide a more parsimonious fit than a pure AR or pure MA model. This section introduces ARMA processes.

### 12.8.1 The Backwards Operator

The *backwards operator*  $B$  is a simple notation with a fancy name. It is useful for describing ARMA (and ARIMA) models. The backwards operator is defined by

$$B Y_t = Y_{t-1}$$

and, more generally,

$$B^h Y_t = Y_{t-h}.$$

Thus,  $B$  backs up time one unit while  $B^h$  does this repeatedly so that time is backed up  $h$  time units. Note that  $B c = c$  for any constant  $c$ , since a constant does not change with time. The backwards operator is sometimes called the *lag operator*.

### 12.8.2 The ARMA Model

An  $ARMA(p, q)$  model combines both AR and MA terms and is defined by the equation

$$(Y_t - \mu) = \phi_1(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) + \epsilon_t + \theta_1\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q}, \quad (12.25)$$

which shows how  $Y_t$  depends on lagged values of itself and lagged values of the white noise process. Equation (12.25) can be written more succinctly with the backwards operator as

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(Y_t - \mu) = (1 + \theta_1 B + \cdots + \theta_q B^q)\epsilon_t. \quad (12.26)$$

A white noise process is  $ARMA(0,0)$  since if  $p = q = 0$ , then (12.26) reduces to

$$(Y_t - \mu) = \epsilon_t.$$

### 12.8.3 ARMA(1,1) Processes

The  $ARMA(1,1)$  model is commonly used in practice and is simple enough to study theoretically. In this section, formulas for its variance and ACF will be derived. Without loss of generality, one can assume that  $\mu = 0$  when computing the variance and ACF. Multiplying the model

$$Y_t = \phi Y_{t-1} + \theta \epsilon_{t-1} + \epsilon_t \quad (12.27)$$

by  $\epsilon_t$  and taking expectations, one has

$$\text{Cov}(Y_t, \epsilon_t) = E(Y_t \epsilon_t) = \sigma_\epsilon^2, \quad (12.28)$$

since  $\epsilon_t$  is independent of  $\epsilon_{t-1}$  and  $Y_{t-1}$ . From (12.27) and (12.28),

$$\gamma(0) = \phi^2 \gamma(0) + (1 + \theta^2) \sigma_\epsilon^2 + 2\phi\theta\sigma_\epsilon^2, \quad (12.29)$$

and then solving (12.29) for  $\gamma(0)$  gives us the formula

$$\gamma(0) = \frac{(1 + \theta^2 + 2\phi\theta)\sigma_\epsilon^2}{1 - \phi^2}. \quad (12.30)$$

By similar calculations, multiplying (12.27) by  $Y_{t-h}$  and taking expectations yields a formula for  $\gamma(h)$ . Dividing this formula by the right-hand side of (12.29) gives us

$$\rho(h) = \frac{(1 + \phi\theta)(\phi + \theta)}{1 + \theta^2 + 2\phi\theta}. \quad (12.31)$$

For  $h \geq 2$ , multiplying (12.27) by  $Y_{t-h}$  and taking expectations results in the formula

$$\rho(h) = \phi\rho(h-1), \quad h \geq 2. \quad (12.32)$$

By (12.32), after one lag, the ACF of an  $ARMA(1,1)$  process decays in the same way as the ACF of an  $AR(1)$  process with the same  $\phi$ .

### 12.8.4 Estimation of ARMA Parameters

The parameters of ARMA models can be estimated by maximum likelihood or conditional least-squares. These methods were introduced for AR(1) processes in Sect. 12.5. The estimation methods for AR( $p$ ) models are very similar to those for AR(1) models. For MA and ARMA, because the noise terms  $\epsilon_1, \dots, \epsilon_n$  are unobserved, there are complications that are best left for advanced time series texts.

#### *Example 12.9. Changes in risk-free returns–ARMA models*

This example uses the monthly changes in the risk-free returns shown in Fig. 4.3. In Table 12.1, AIC and BIC are shown for ARMA models with  $p, q = 0, 1, 2$ . We see that AIC and BIC are both minimized by the ARMA(1,1) model, though the MA(2) model is a very close second. The ARMA(1,1) and MA(2) fit nearly equally well, and it is difficult to decide between them.

Sample ACF, normal, and time series plots of the residuals from the ARMA(1,1) model are shown in Fig. 12.13. The ACF plot shows no short-term autocorrelation, which is another sign that the ARMA(1,1) model is satisfactory. However, the normal plot shows heavy tails and the residual time series plot shows volatility clustering. These problems will be addressed in later chapters.  $\square$

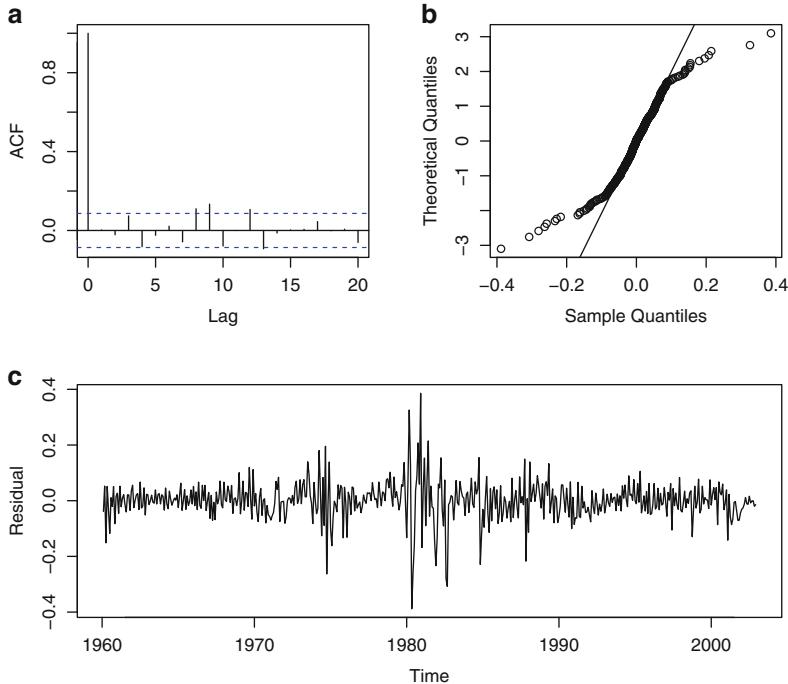
**Table 12.1.** *AIC and BIC for ARMA models fit to the monthly changes in the risk-free interest returns. The minimum values of both criteria are shown in boldface. To improve the appearance of the table, 1290 was added to all AIC and BIC values.*

| $p$ | $q$ | AIC         | BIC         |
|-----|-----|-------------|-------------|
| 0   | 0   | 29.45       | 37.8        |
| 0   | 1   | 9.21        | 21.8        |
| 0   | 2   | 3.00        | 19.8        |
| 1   | 0   | 14.86       | 27.5        |
| 1   | 1   | <b>2.67</b> | <b>19.5</b> |
| 1   | 2   | 4.67        | 25.7        |
| 2   | 0   | 5.61        | 22.4        |
| 2   | 1   | 6.98        | 28.0        |
| 2   | 2   | 4.89        | 30.1        |

### 12.8.5 The Differencing Operator

The *differencing operator* is another useful notation and is defined as  $\Delta = 1 - B$ , where  $B$  is the backwards operator, so that

$$\Delta Y_t = Y_t - B Y_t = Y_t - Y_{t-1}.$$



**Fig. 12.13.** Residual sample ACF, normal quantile, and time series plots for the ARMA(1, 1) fit to the monthly changes in the risk-free returns.

For example, if  $p_t = \log(P_t)$  is the log price, then the log return is

$$r_t = \Delta p_t.$$

Differencing can be iterated. For example,

$$\begin{aligned} \Delta^2 Y_t &= \Delta(\Delta Y_t) = \Delta(Y_t - Y_{t-1}) = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) \\ &= Y_t - 2Y_{t-1} + Y_{t-2}. \end{aligned}$$

$\Delta^k$  is called the  $k$ th-order differencing operator. A general formula for  $\Delta^k$  can be derived from a binomial expansion:

$$\Delta^k Y_t = (1 - B)^k Y_t = \sum_{\ell=0}^k \binom{k}{\ell} (-1)^\ell Y_{t-\ell}. \quad (12.33)$$

## 12.9 ARIMA Processes

Often the first or perhaps second differences of nonstationary time series are stationary. For example, the first differences of a random walk (nonstationary) are white noise (stationary). In this section, *autoregressive integrated moving average* (ARIMA) processes are introduced. They include stationary as well as nonstationary processes.

A time series  $Y_t$  is said to be an  $ARIMA(p, d, q)$  process if  $\Delta^d Y_t$  is  $ARMA(p, q)$ . For example, if log returns on an asset are  $ARMA(p, q)$ , then the log prices are  $ARIMA(p, 1, q)$ . An  $ARIMA(p, d, q)$  is stationary only if  $d = 0$ . Otherwise, only its differences of order  $d$  or above are stationary.

Notice that an  $ARIMA(p, 0, q)$  model is the same as an  $ARMA(p, q)$  model.  $ARIMA(p, 0, 0)$ ,  $ARMA(p, 0)$ , and  $AR(p)$  models are the same. Similarly,  $ARIMA(0, 0, q)$ ,  $ARMA(0, q)$ , and  $MA(q)$  models are the same. A random walk is an  $ARIMA(0, 1, 0)$  model, and white noise is an  $ARIMA(0, 0, 0)$  model.

The inverse of differencing is “integrating.” The integral of a process  $Y_t$  is the process  $w_t$ , where

$$w_t = w_{t_0} + Y_{t_0+1} + \cdots + Y_t. \quad (12.34)$$

Here  $t_0$  is an arbitrary starting time point and  $w_{t_0}$  is the starting value of the  $w_t$  process. It is easy to check that

$$\Delta w_t = Y_t, \quad (12.35)$$

so integrating and differencing are inverse processes.<sup>7</sup>

We will say that a process is  $I(d)$  if it is stationary after being differenced  $d$  times. For example, a stationary process is  $I(0)$ . An  $ARIMA(p, d, q)$  process is  $I(d)$ . An  $I(d)$  process is said to be “integrated to order  $d$ .”

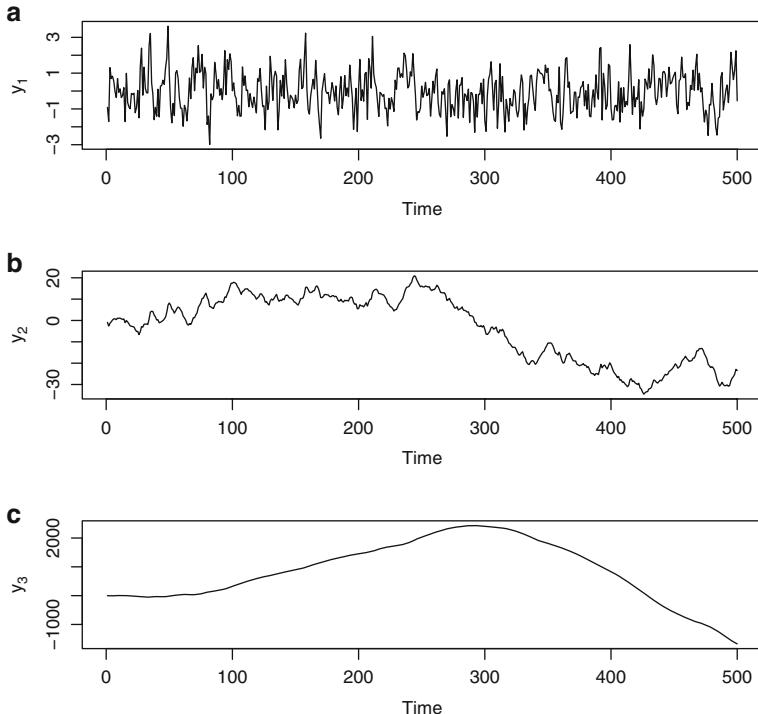
Figure 12.14 shows an  $AR(1)$  process, its integral, and its second integral, meaning the integral of its integral. These three processes are  $I(0)$ ,  $I(1)$ , and  $I(2)$ , respectively. The three processes behave in entirely different ways. The  $AR(1)$  process is stationary and varies randomly about its mean, which is 0; one says that the process *reverts* to its mean. The integral of this process behaves much like a random walk in having no fixed level to which it reverts. The second integral has *momentum*. Once the process starts moving upward or downward, it tends to continue in that direction. If data show momentum like this, then the momentum is an indication that  $d = 2$ . The  $AR(1)$  process was generated by the R function `arima.sim()`. This process was integrated twice with R’s `cumsum()` function.

```
21 set.seed(4631)
22 y1 = arima.sim(n = 500, list(ar = c(0.4)))
23 y2 = cumsum(y1)
24 y3 = cumsum(y2)
```

*Example 12.10. Fitting an ARIMA model to CPI data*

---

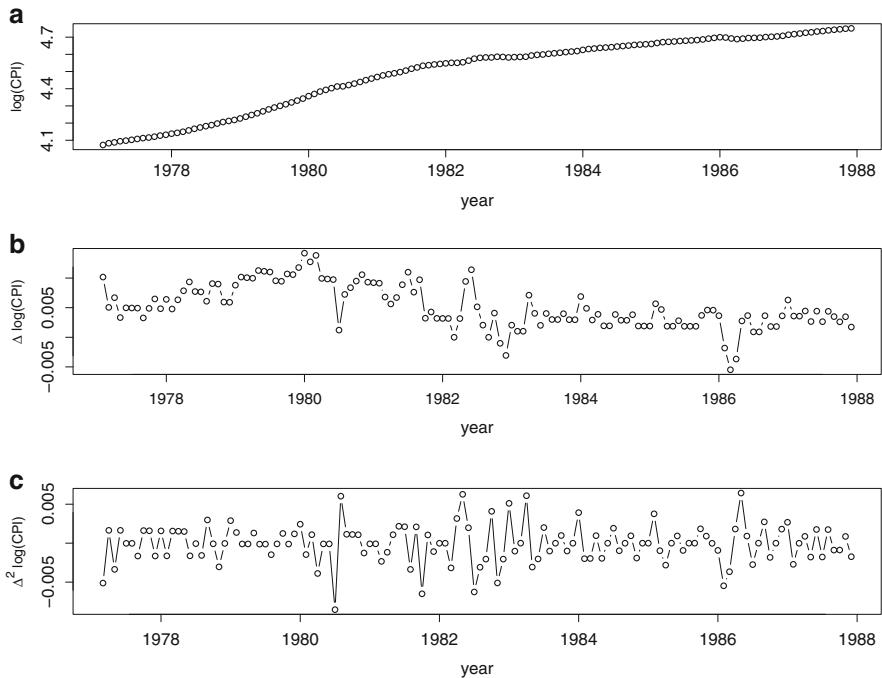
<sup>7</sup> An analog is, of course, differentiation and integration in calculus, which are inverses of each other.



**Fig. 12.14.** The (a) top plot is of an AR(1) process with  $\mu = 0$  and  $\phi = 0.4$ . The (b) middle and (c) bottom plots are, respectively, the integral and second integral of this AR(1) process. Thus, from top to bottom, the series are I(0), I(1), and I(2), respectively.

This example uses the `CPI.dat.csv` data set. CPI is a seasonally adjusted U.S. Consumer Price Index. The data are monthly. Only data from January 1977 to December 1987 are used in this example. Figure 12.15 shows time series plots of  $\log(\text{CPI})$  and the first and second differences of this series. The original series shows the type of momentum that is characteristic of an I(2) series. The first differences show no momentum, but they do not appear to be mean-reverting and so they may be I(1). The second differences appear to be mean-reverting and therefore seem to be I(0). ACF plots in Fig. 12.16a,b, and c provide additional evidence that the  $\log(\text{CPI})$  is I(2).

Notice that the ACF of  $\Delta^2 \log(\text{CPI})$  has large correlations at the first two lags and then small autocorrelations after that. This suggests using an MA(2) for  $\Delta^2 \log(\text{CPI})$  or, equivalently, an ARIMA(0,2,2) model for  $\log(\text{CPI})$ . The ACF of the residuals from this fit is shown in Fig. 12.16d. The residual ACF has small correlations at short lags, which is an indication that the ARIMA(0,2,2) model fits well. Also, the residuals pass Ljung–Box tests for various choices of `lag`, for example, with a  $p$ -value of 0.08 at `lag = 20`, with `fitdf = 2`.  $\square$



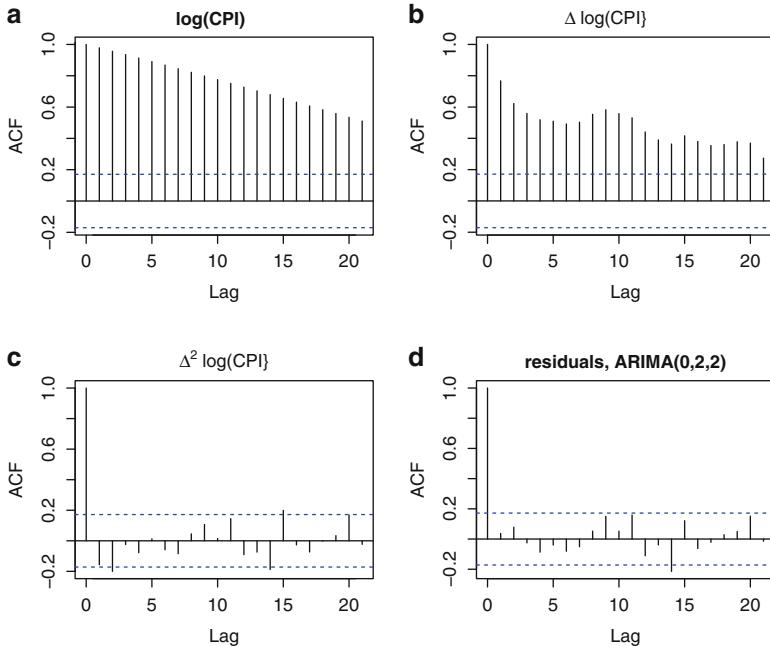
**Fig. 12.15.** (a)  $\log(\text{CPI})$ , (b) first differences of  $\log(\text{CPI})$ , and (c) second differences of  $\log(\text{CPI})$ .

*Example 12.11. Fitting an ARIMA model to industrial production (IP) data*

This example uses the IP.dat data set. The variable, IP, is a seasonally adjusted U.S. industrial production index. Figure 12.17 panels (a) and (b) show time series plots of  $\log(\text{IP})$  and  $\Delta \log(\text{IP})$  and panel (c) has the sample ACF of  $\Delta \log(\text{IP})$ . The  $\log(\text{IP})$  series appears to be I(1), implying that we should fit an ARMA model to  $\Delta \log(\text{IP})$ . AR(1), AR(2), and ARMA(1,1) each fit  $\Delta \log(\text{IP})$  reasonably well and the ARMA(1,0) model is selected using the BIC criterion with R's `auto.arima()` function. The ACF of the residuals in Fig. 12.17d indicates a satisfactory fit to the ARMA(1,0) model since it shows virtually no short-term autocorrelation. In summary,  $\log(\text{IP})$  is well fit by an ARIMA(1,1,0) model.  $\square$

### 12.9.1 Drifts in ARIMA Processes

If a nonstationary process has a constant mean, then the first differences of this process have mean zero. For this reason, it is often assumed that a differenced process has mean zero. The `arima()` function in R makes this assumption.



**Fig. 12.16.** Sample ACF of (a)  $\log(\text{CPI})$ , (b) first differences of  $\log(\text{CPI})$ , (c) second differences of  $\log(\text{CPI})$ , and (d) residuals from an ARIMA(0,2,2) model fit to  $\log(\text{CPI})$ .

Instead of a constant mean, sometimes a nonstationary process has a mean with a deterministic linear trend, e.g.,  $E(Y_t) = \beta_0 + \beta_1 t$ . Then,  $\beta_1$  is called the *drift* of  $Y_t$ . Note that  $E(\Delta Y_t) = \beta_1$ , so if  $Y_t$  has a nonzero drift then  $\Delta Y_t$  has a nonzero mean. The R function `auto.arima()` discussed in Sect. 12.11 allows a differenced process to have a nonzero mean, which is called the *drift* in the output.

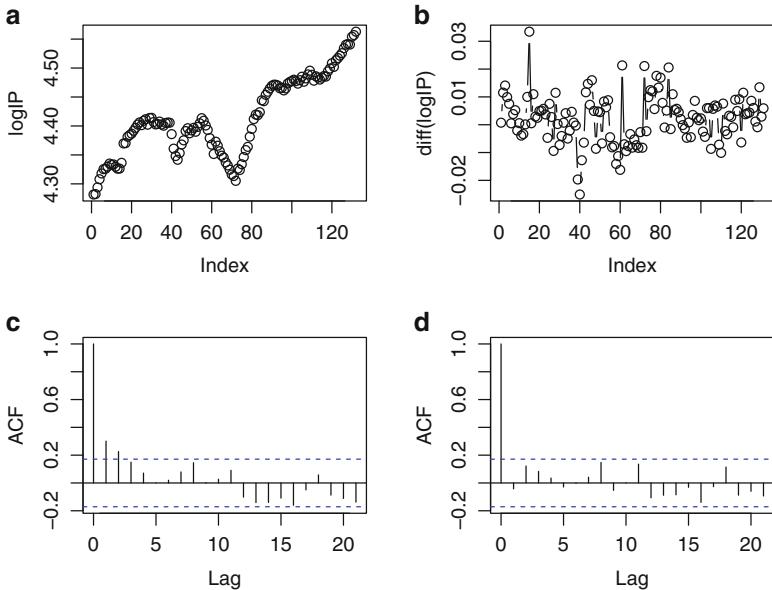
These ideas can be extended to higher-degree polynomial trends and higher-order differencing. If  $E(Y_t)$  has an  $m$ th-degree polynomial trend, then the mean of  $E(\Delta^d Y_t)$  has an  $(m - d)$ th-degree trend for  $d \leq m$ . For  $d > m$ ,  $E(\Delta^d Y_t) = 0$ .

## 12.10 Unit Root Tests

We have seen that it can be difficult to tell whether a time series is best modeled as stationary or nonstationary. To help decide between these two possibilities, it can be helpful to use hypothesis testing.

What is meant by a unit root? Recall that an ARMA( $p, q$ ) process can be written as

$$(Y_t - \mu) = \phi_1(Y_{t-1} - \mu) + \cdots + \phi_p(Y_{t-p} - \mu) + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}. \quad (12.36)$$



**Fig. 12.17.** Time series plots of (a)  $\log(IP)$  and (b)  $\Delta \log(IP)$ , and sample ACF plots of (c)  $\Delta \log(IP)$  and (d) residual from ARMA(1,0) fit to  $\Delta \log(IP)$ .

The condition for  $\{Y_t\}$  to be stationary is that all roots of the polynomial

$$1 - \phi_1 x - \cdots - \phi_p x^p \quad (12.37)$$

have absolute values greater than one. (See Appendix A.21 for information about complex roots of polynomials and the absolute value of a complex number.) For example, when  $p = 1$ , then (12.37) is

$$1 - \phi x$$

and has one root,  $1/\phi$ . We know that the process is stationary if  $|\phi| < 1$ , which, of course, is equivalent to  $|1/\phi| > 1$ .

If there is a unit root, that is, a root with an absolute value equal to 1, then the ARMA process is nonstationary and behaves much like a random walk. Not surprisingly, this is called the unit root case. The explosive case is when a root has an absolute value less than 1.

#### Example 12.12. Inflation rates

Recall from Examples 12.1 and 12.3 that we have had difficulty deciding whether the inflation rates are stationary or not. If we fit stationary ARMA models to the inflation rates, then `auto.arima()` selects an ARMA(2,1) model and the AR coefficients are  $\hat{\phi}_1 = 1.229$  and  $\hat{\phi}_2 = -0.233$ . The roots of

$$1 - \hat{\phi}_1 x - \hat{\phi}_2 x^2$$

can be found easily using R's `polyroot()` function and are 1.0053 and 4.2694.

```
25 polyroot(c(1, -1.229, +0.233))
```

Both roots have absolute values greater than 1, indicating possible stationarity; however, the first is very close to 1, and since the roots are estimated with error, there is reason to suspect that this series may be nonstationary.  $\square$

Unit root tests are used to decide if an AR model has an absolute root equal to 1. One popular unit root test is the augmented Dickey–Fuller test, often called the ADF test. The null hypothesis is that there is a unit root. The usual alternative is that the process is stationary, but one can instead use the alternative that the process is explosive.

Another unit root test is the Phillips–Perron test. It is similar to the Dickey–Fuller test, but differs in some details.

A third test is the KPSS test. The null hypothesis for the KPSS test is stationarity and the alternative is a unit root, just the opposite of the hypotheses for the Dickey–Fuller and Phillips–Perron tests.

*Example 12.13. Inflation rates—unit root tests*

Recall that we were undecided as to whether or not the inflation rate time series was stationary. The unit root tests might help resolve this issue, but unfortunately they do not provide unequivocal evidence in favor of stationarity. Both the augmented Dickey–Fuller and Phillips–Perron tests, which were implemented in R with the functions `adf.test()` and `pp.test()`, respectively, have small  $p$ -values, 0.016 for the former and less than 0.01 for the latter; see the output below. The functions `adf.test()`, `pp.test()`, and `kpss.test()` (used below) are in R's `tseries` package. Therefore, at level 0.05 the null hypothesis of a unit root is rejected by both tests in favor of the alternative of stationarity, the default alternative hypothesis for both `adf.test()` and `pp.test()`.

```
26 library(tseries)
27 adf.test(y)

Augmented Dickey-Fuller Test
data: y
Dickey-Fuller = -3.8651, Lag order = 7, p-value = 0.01576
alternative hypothesis: stationary

28 pp.test(y)

Phillips-Perron Unit Root Test
data: y
Dickey-Fuller Z(alpha) = -248.75, Truncation lag parameter = 5,
p-value = 0.01
alternative hypothesis: stationary
```

Although the augmented Dickey–Fuller and Phillips–Perron tests suggest that the inflation rate series is stationary since the null hypothesis of a unit root is rejected, the KPSS test leads one to the opposite conclusion. The null hypothesis for the KPSS is stationarity and it is rejected with a  $p$ -value smaller than 0.01. Here is the R output.

```
29 kpss.test(y)

 KPSS Test for Level Stationarity
 data: y
 KPSS Level = 2.51, Truncation lag parameter = 5, p-value = 0.01
```

Thus, the unit root tests are somewhat contradictory. Perhaps the inflation rates are stationary with long-term memory. Long-memory processes will be introduced in Sect. 13.5.  $\square$

### 12.10.1 How Do Unit Root Tests Work?

A full discussion of the theory behind unit root tests is beyond the scope of this book. Here, only the basic idea will be mentioned. See Sect. 12.14 for more information. The Dickey–Fuller test is based on the AR(1) model

$$Y_t = \phi Y_{t-1} + \epsilon_t. \quad (12.38)$$

The null hypothesis ( $H_0$ ) is that there is a unit root, that is,  $\phi = 1$ , and the alternative ( $H_1$ ) is stationarity, which is equivalent to  $\phi < 1$ , assuming, as seems reasonable, that  $\phi > -1$ . The AR(1) model (12.38) is equivalent to  $\Delta Y_t = (\phi - 1)Y_{t-1} + \epsilon_t$ , or

$$\Delta Y_t = \pi Y_{t-1} + \epsilon_t, \quad (12.39)$$

where  $\pi = \phi - 1$ . Stated in terms of  $\pi$ ,  $H_0$  is  $\pi = 0$  and  $H_1$  is  $\pi < 0$ . The Dickey–Fuller test regresses  $\Delta Y_t$  on  $Y_{t-1}$  and tests  $H_0$ . Because  $Y_{t-1}$  is nonstationary under  $H_0$ , the  $t$ -statistic for  $\pi$  has a nonstandard distribution so special tables need to be developed in order to compute  $p$ -values.

The augmented Dickey–Fuller test expands model (12.39) by adding a time trend and lagged values of  $\Delta Y_t$ . Typically, the time trend is linear so that the expanded model is

$$\Delta Y_t = \beta_0 + \beta_1 t + \pi Y_{t-1} + \sum_{j=1}^p \gamma_j \Delta Y_{t-j} + \epsilon_t. \quad (12.40)$$

The hypotheses are still  $H_0: \pi = 0$  and  $H_1: \pi < 0$ . There are several methods for selecting  $p$ . The `adf.test()` function has a default value of  $p$  equal to `trunc((length(y)-1)^(1/3))`, where  $y$  is the input series ( $Y_t$  in our notation).

## 12.11 Automatic Selection of an ARIMA Model

It is useful to have an automatic method for selecting an ARIMA model. As always, an automatically selected model should not be accepted blindly, but it makes sense to start model selection with something chosen quickly and by an objective criterion.

The R function `auto.arima()` can select all three parameters,  $p$ ,  $d$ , and  $q$ , for an ARIMA model. The differencing parameter  $d$  is selected using the KPSS test. If the null hypothesis of stationarity is accepted when the KPSS is applied to the original time series, then  $d = 0$ . Otherwise, the series is differenced until the KPSS accepts the null hypothesis. After that,  $p$  and  $q$  are selected using either AIC or BIC.

*Example 12.14. Inflation rates—automatic selection of an ARIMA model*

In this example, `auto.arima()` is applied to the inflation rates. The ARIMA (1,1,1) model is selected by `auto.arima()` using either AIC or BIC to select  $p$  and  $q$  after  $d = 1$  is selected by the KPSS test.

```
30 auto.arima(y, max.p = 5, max.q = 5, ic = "bic", trace = FALSE)

Series: y
ARIMA(1,1,1)

Coefficients:
 ar1 ma1
 0.238 -0.877
 s.e. 0.055 0.027

sigma^2 estimated as 8.55: log likelihood=-1221.6
AIC=2449.2 AICc=2449.3 BIC=2461.8
```

This is a very parsimonious model and residual diagnostics (not shown) show that it fits well.

AICc in the output from `auto.arima()` is the value of the corrected AIC criterion defined by (5.34). The sample size is 491 so, not surprisingly, AICc is equal to AIC, at least after rounding to the nearest integer.  $\square$

## 12.12 Forecasting

Forecasting means predicting future values of a time series using the current *information set*, which is the set of present and past values of the time series. In some contexts, the information set could include other variables related to the time series, but in this section the information set contains only the past and present values of the time series that is being predicted.

ARIMA models are often used for forecasting. Consider forecasting using an AR(1) process. Suppose that we have data  $Y_1, \dots, Y_n$  and estimates  $\hat{\mu}$  and  $\hat{\phi}$ . We know that

$$Y_{n+1} = \mu + \phi(Y_n - \mu) + \epsilon_{n+1}. \quad (12.41)$$

Since  $\epsilon_{n+1}$  is independent of the past and present, by Result 11.1 in Sect. 11.9.2 the best predictor of  $\epsilon_{n+1}$  is its expected value, which is 0. We know, of course, that  $\epsilon_{n+1}$  is not 0, but 0 is our best guess at its value. On the other hand, we know or have estimates of all other quantities in (12.41). Therefore, we predict  $Y_{n+1}$  by

$$\hat{Y}_{n+1} = \hat{\mu} + \hat{\phi}(Y_n - \hat{\mu}).$$

By the same reasoning we forecast  $Y_{n+2}$  by

$$\hat{Y}_{n+2} = \hat{\mu} + \hat{\phi}(\hat{Y}_{n+1} - \hat{\mu}) = \hat{\mu} + \hat{\phi}\{\hat{\phi}(Y_n - \hat{\mu})\}, \quad (12.42)$$

and so forth. Notice that in (12.42) we do not use  $Y_{n+1}$ , which is unknown at time  $n$ , but rather the forecast  $\hat{Y}_{n+1}$ . Continuing in this way, we find the general formula for the  $k$ -step-ahead forecast:

$$\hat{Y}_{n+k} = \hat{\mu} + \hat{\phi}^k(Y_n - \hat{\mu}). \quad (12.43)$$

If  $|\hat{\phi}| < 1$ , as is true for a stationary series, then as  $k$  increases, the forecasts will converge geometrically fast to  $\hat{\mu}$ .

Formula (12.43) is valid only for AR(1) processes, but forecasting other AR( $p$ ) processes is similar. For an AR(2) process,

$$\hat{Y}_{n+1} = \hat{\mu} + \hat{\phi}_1(Y_n - \hat{\mu}) + \hat{\phi}_2(Y_{n-1} - \hat{\mu})$$

and

$$\hat{Y}_{n+2} = \hat{\mu} + \hat{\phi}_1(\hat{Y}_{n+1} - \hat{\mu}) + \hat{\phi}_2(Y_n - \hat{\mu}),$$

and so on.

Forecasting ARMA and ARIMA processes is similar to forecasting AR processes. Consider the MA(1) process,  $Y_t - \mu = \epsilon_t + \theta\epsilon_{t-1}$ . Then the next observation will be

$$Y_{n+1} = \mu + \epsilon_{n+1} + \theta\epsilon_n. \quad (12.44)$$

In the right-hand side of (12.44) we replace  $\mu$  and  $\theta$  by estimates and  $\epsilon_n$  by the residual  $\hat{\epsilon}_n$ . Also, since  $\epsilon_{n+1}$  is independent of the observed data, it is replaced by its mean 0. Then the forecast is

$$\hat{Y}_{n+1} = \hat{\mu} + \hat{\theta}\hat{\epsilon}_n.$$

The two-step-ahead forecast of  $Y_{n+2} = \mu + \epsilon_{n+2} + \theta\epsilon_{n+1}$  is simply  $\hat{Y}_{n+2} = \hat{\mu}$ , since  $\epsilon_{n+1}$  and  $\epsilon_{n+2}$  are independent of the observed data. Similarly,  $\hat{Y}_{n+k} = \hat{\mu}$  for all  $k > 2$ .

To forecast the ARMA(1,1) process

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t + \theta\epsilon_{t-1},$$

we use

$$\hat{Y}_{n+1} = \hat{\mu} + \hat{\phi}(Y_n - \hat{\mu}) + \hat{\theta}\hat{\epsilon}_n$$

as the one-step-ahead forecast and

$$\hat{Y}_{n+k} = \hat{\mu} + \hat{\phi}(\hat{Y}_{n+k-1} - \hat{\mu}), k \geq 2$$

for forecasting two or more steps ahead.

As a final example, suppose that  $Y_t$  is ARIMA(1,1,0), so that  $\Delta Y_t$  is AR(1). To forecast  $Y_{n+k}$ ,  $k \geq 1$ , one first fits an AR(1) model to the  $\Delta Y_t$  process and forecasts  $\Delta Y_{n+k}$ ,  $k \geq 1$ . Let the forecasts be denoted by  $\widehat{\Delta Y}_{n+k}$ ,  $k \geq 1$ . Then, since

$$Y_{n+1} = Y_n + \Delta Y_{n+1},$$

the forecast of  $Y_{n+1}$  is

$$\hat{Y}_{n+1} = Y_n + \widehat{\Delta Y}_{n+1},$$

and similarly

$$\hat{Y}_{n+2} = \hat{Y}_{n+1} + \widehat{\Delta Y}_{n+2} = Y_n + \widehat{\Delta Y}_{n+1} + \widehat{\Delta Y}_{n+2},$$

and so on.

Most time series software packages offer functions to automate forecasting. R's `predict()` function forecasts using an "object" returned by the `arima()` fitting function.

### 12.12.1 Forecast Errors and Prediction Intervals

When making forecasts, one would of course like to know the uncertainty of the predictions. To this end, one first computes the variance of the forecast error. Then a  $(1 - \alpha)100\%$  prediction interval is the forecast itself plus or minus the forecast error's standard deviation times  $z_{\alpha/2}$  (the normal upper quantile). The use of  $z_{\alpha/2}$  assumes that  $\epsilon_1, \epsilon_2, \dots$  is Gaussian white noise. If the residuals are heavy-tailed, then we might be reluctant to make the Gaussian assumption. One way to avoid this assumption is discussed in Sect. 12.12.2.

Computation of the forecast error variance and the prediction interval is automated by modern statistical software, so we need not present general formulas for the forecast error variance. However, to gain some understanding of general principles, we will look at two special cases, one stationary and the other nonstationary.

### Stationary AR(1) Forecast Errors

We will first consider the errors made when forecasting a stationary AR(1) process. The error in the first prediction is

$$\begin{aligned} Y_{n+1} - \hat{Y}_{n+1} &= \{\mu + \phi(Y_n - \mu) + \epsilon_{n+1}\} - \{\hat{\mu} + \hat{\phi}(Y_n - \hat{\mu})\} \\ &= (\mu - \hat{\mu}) + (\phi - \hat{\phi})Y_n - (\phi\mu - \hat{\phi}\hat{\mu}) + \epsilon_{n+1} \end{aligned} \quad (12.45)$$

$$\approx \epsilon_{n+1}. \quad (12.46)$$

Here (12.45) is the exact error and (12.46) is a “large-sample” approximation. The basis for (12.46) is that as the sample size increases  $\hat{\mu} \rightarrow \mu$  and  $\hat{\phi} \rightarrow \phi$ , so the first three terms in (12.45) converge to 0, but the last term remains unchanged. The large-sample approximation simplifies formulas and helps us focus on the main components of the forecast error. Using the large-sample approximation again, so  $\hat{\mu}$  is replaced by  $\mu$  and  $\hat{\phi}$  by  $\phi$ , the error in the two-steps-ahead forecast is

$$\begin{aligned} Y_{n+2} - \hat{Y}_{n+2} &= \{\mu + \phi(Y_{n+1} - \mu) + \epsilon_{n+2}\} - \{\mu + \phi(\hat{Y}_{n+1} - \mu)\} \\ &= \phi(Y_{n+1} - \hat{Y}_{n+1}) + \epsilon_{n+1} \\ &= \phi\epsilon_{n+1} + \epsilon_{n+2}. \end{aligned} \quad (12.47)$$

Continuing in this manner, we find that the  $k$ -step-ahead forecasting error is

$$\begin{aligned} Y_{n+k} - \hat{Y}_{n+k} &\approx \{\mu + \phi(Y_{n+k-1} - \mu) + \epsilon_{n+k}\} - \{\mu + \phi(\hat{Y}_{n+k-1} - \mu)\} \\ &= \phi^{k-1}\epsilon_{n+1} + \phi^{k-2}\epsilon_{n+2} + \cdots + \phi\epsilon_{n+k-1} + \epsilon_{n+k}. \end{aligned} \quad (12.48)$$

By the formula for the sum of a finite geometric series, the variance of the right-hand side of (12.48) is

$$\begin{aligned} \left\{ \phi^{2(k-1)} + \phi^{2(k-2)} + \cdots + \phi^2 + 1 \right\} \sigma_\epsilon^2 &= \left( \frac{1 - \phi^{2k}}{1 - \phi^2} \right) \sigma_\epsilon^2 \\ &\rightarrow \frac{\sigma_\epsilon^2}{1 - \phi^2} \text{ as } k \rightarrow \infty. \end{aligned} \quad (12.49)$$

An important point here is that the variance of the forecast error does not diverge as  $k \rightarrow \infty$ , but rather the variance converges to  $\gamma(0)$ , the marginal covariance of the AR(1) process given by (12.7). This is an example of the general principle that for any stationary ARMA process, the variance of the forecast error converges to the marginal variance.

### Forecasting a Random Walk

For the random walk process,  $Y_{n+1} = \mu + Y_n + \epsilon_{n+1}$ , many of the formulas just derived for the AR(1) process still hold, but with  $\phi = 1$ . An exception is

that the last result in (12.49) does not hold because the summation formula for a geometric series does not apply when  $\phi = 1$ . One result that does still hold is

$$Y_{n+k} - \hat{Y}_{n+k} = \epsilon_{n+1} + \epsilon_{n+2} + \cdots + \epsilon_{n+k-1} + \epsilon_{n+k}$$

so the variance of the  $k$ -step-ahead forecast error is  $k\sigma_\epsilon^2$  and, unlike for the stationary AR(1) case, the forecast error variance diverges to  $\infty$  as  $k \rightarrow \infty$ .

## Forecasting ARIMA Processes

As mentioned before, in practice we do not need general formulas for the forecast error variance of ARIMA processes, since statistical software can compute the variance. However, it is worth repeating a general principle: For stationary ARMA processes, the variance of the  $k$ -step-ahead forecast error variance converges to a finite value as  $k \rightarrow \infty$ , but for a nonstationary ARIMA process this variance converges to  $\infty$ . The result of this principle is that for a nonstationary process, the forecast limits diverge away from each other as  $k \rightarrow \infty$ , but for a stationary process the forecast limits converge to parallel horizontal lines.

*Example 12.15. Forecasting the one-month inflation rate*

We saw in Example 12.8 that an MA(3) model provided a parsimonious fit to the changes in the one-month inflation rate. This implies that an ARIMA(0,1,3) model will be a good fit to the inflation rates themselves. The two models are, of course, equivalent, but they forecast different series. The first model gives forecasts and confidence limits for the changes in the inflation rate, while the second model provides forecasts and confidence limits for the inflation rate itself.

Figures 12.18 and 12.19 plot forecasts and forecast limits from the two models out to 100 steps ahead. One can see that the forecast limits diverge for the second model and converge to parallel horizontal lines for the first model.  $\square$

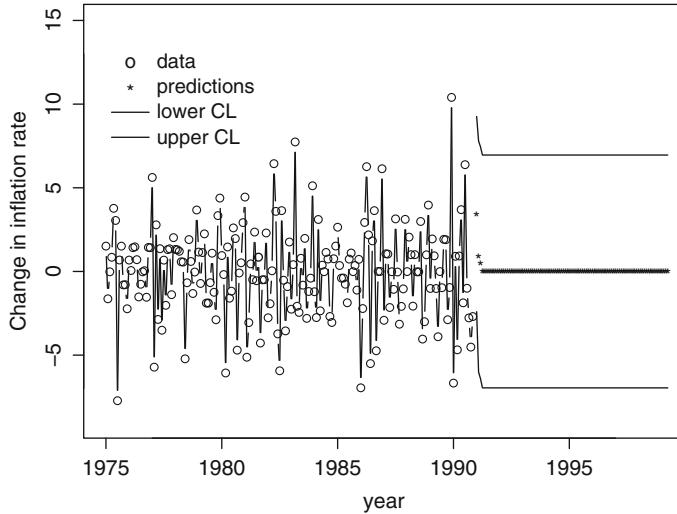
### 12.12.2 Computing Forecast Limits by Simulation

Simulation can be used to compute forecasts limits. This is done by simulating random forecasts and finding their  $\alpha/2$ -upper and -lower sample quantiles. A set of random forecasts up to  $m$  time units ahead is generated for an ARMA process by recursion:

$$\begin{aligned}\hat{Y}_{n+t} &= \hat{\mu} + \hat{\phi}_1(\hat{Y}_{n+t-1} - \hat{\mu}) + \cdots + \hat{\phi}_p(\hat{Y}_{n+t-p} - \hat{\mu}) \\ &\quad + \hat{\epsilon}_{n+t} + \hat{\theta}_1\hat{\epsilon}_{n+t-1} + \cdots + \hat{\theta}_q\hat{\epsilon}_{n+t-q}, \quad t = 1, \dots, m,\end{aligned}\quad (12.50)$$

in which

- $\hat{\epsilon}_k$  is the  $k$ th residual if  $k \leq n$ ,
- $\{\hat{\epsilon}_k : k = n+1, \dots, n+m\}$  is a resample from the residuals.



**Fig. 12.18.** Forecasts of changes in inflation rate.

Thus,  $\hat{Y}_{n+1}$  is generated from  $Y_{n+1-p}, \dots, Y_n, \hat{\epsilon}_{n+1-q}, \dots, \hat{\epsilon}_{n+1}$ , then  $\hat{Y}_{n+2}$  is generated from  $Y_{n+2-p}, \dots, Y_n, \hat{Y}_{n+1}, \hat{\epsilon}_{n+2-q}, \dots, \hat{\epsilon}_{n+2}$ , then  $\hat{Y}_{n+3}$  is generated from  $Y_{n+3-p}, \dots, Y_n, \hat{Y}_{n+1}, \hat{Y}_{n+2}, \hat{\epsilon}_{n+3-q}, \dots, \hat{\epsilon}_{n+3}$ , and so forth.

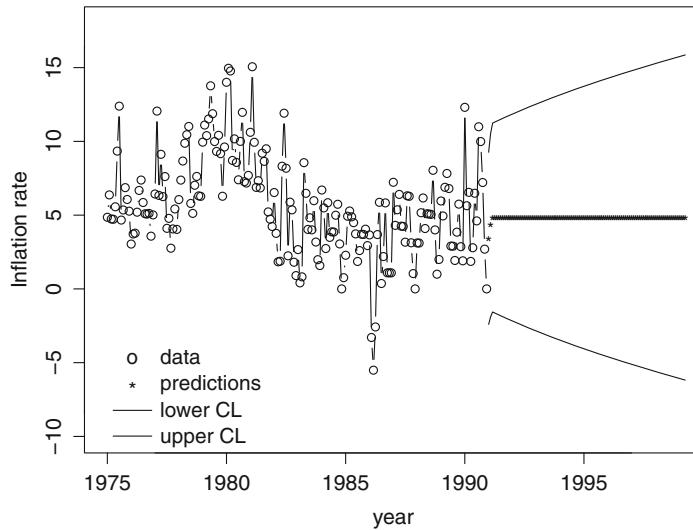
A large number, call it  $B$ , of sets of random forecasts are generated in this way. They differ because their sets of future errors generated in step two are mutually independent. For each  $t = 1, \dots, m$ , the  $\alpha/2$ -upper and -lower sample quantiles of the  $B$  random values of  $\hat{Y}_{n+h}$  are the forecast limits for  $Y_{n+h}$ .

To obtain forecasts, rather than forecast limits, one uses  $\hat{\epsilon}_k = 0$ , for  $k = n+1, \dots, n+m$ , in step two. The forecasts are nonrandom, conditional given the data, and therefore need to be computed only once.

If  $Y_t = \Delta W_t$  for some nonstationary series  $\{W_1, \dots, W_n\}$ , then random forecasts of  $\{W_{n+1}, \dots\}$  can be obtained as partial sums of  $\{W_n, \hat{Y}_{n+1}, \dots\}$ . For example,

$$\begin{aligned}\hat{W}_{n+1} &= W_n + \hat{Y}_{n+1}, \\ \hat{W}_{n+2} &= \hat{W}_{n+1} + \hat{Y}_{n+2} = W_n + \hat{Y}_{n+1} + \hat{Y}_{n+2}, \\ \hat{W}_{n+3} &= \hat{W}_{n+2} + \hat{Y}_{n+3} = W_n + \hat{Y}_{n+1} + \hat{Y}_{n+2} + \hat{Y}_{n+3},\end{aligned}$$

and so forth. Then, upper and lower quantiles of the randomly generated  $\hat{W}_{n+h}$  can be used as forecast limits for  $W_{n+h}$ .

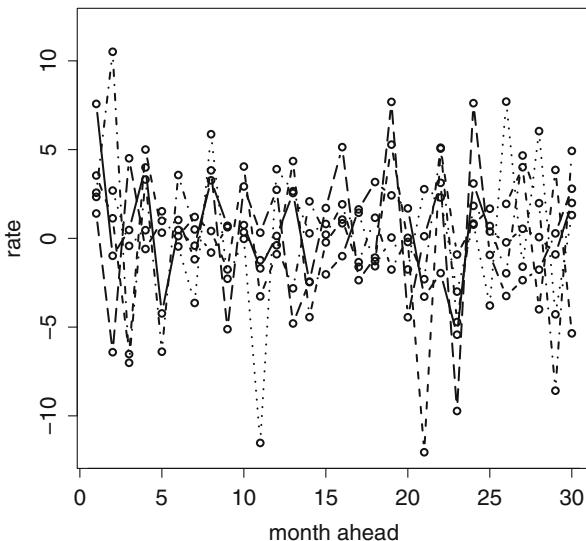


**Fig. 12.19.** Forecasts of inflation rate.

*Example 12.16.* Forecasting the one-month inflation rate and changes in the inflation rate by simulation

To illustrate the amount of random variation in the forecasts, a small number (five) of sets of random forecasts of the changes in the inflation rate were generated out to 30 months ahead. These are plotted in Fig. 12.20. Notice the substantial random variation between the random forecasts. Because of this large variation, to calculate forecasts limits a much larger number of random forecasts should be used. In this example,  $B = 50,000$  sets of random forecasts are generated. Figure 12.21 shows the forecast limits, which are the 2.5 % upper and lower sample quantiles. For comparison, the forecast limits generated by R's function `arima()` are also shown. The two sets of forecast limits are very similar even though the `arima()` limits assume Gaussian noise, but the residuals are heavy-tailed. Thus, the presence of heavy tails does not invalidate the Gaussian limits in this example with 95 % forecast limits. If a larger confidence coefficient were used, that is, one very close to 1, then the forecast intervals based on sampling heavy-tailed residuals would be wider than those based on a Gaussian assumption.

As described above, forecasts for future inflation rates were obtained by taking partial sums of random forecasts of changes in the inflation rate and the forecast limits (upper and lower quantiles) are shown in Fig. 12.22. As expected for a nonstationary process, the forecast limits diverge.  $\square$



**Fig. 12.20.** Five random sets of forecasts of changes in the inflation rate computed by simulation.

There are two important advantages to using simulation for forecasting:

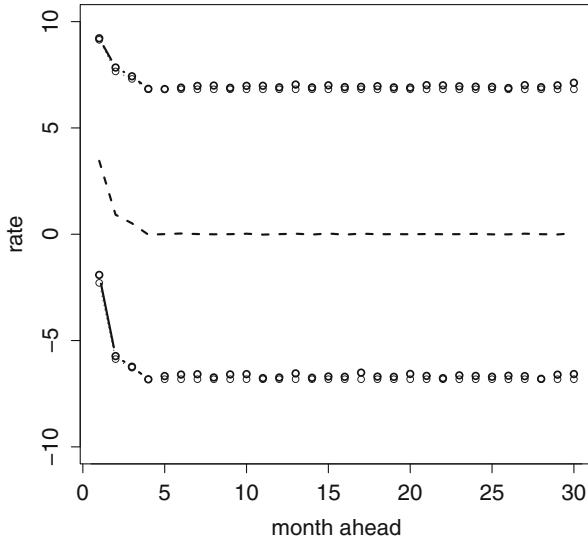
1. simulation can be used in situations where standard software does not compute forecast limits, and
2. simulation does not require that the noise series be Gaussian.

The first advantage will be important in some future examples, such as, multivariate AR processes fit by R's `ar()` function. The second advantage is less important if one is generating 90 % or 95 % forecast limits, but if one wishes more extreme quantiles, say 99 % forecast limits, then the second advantage could be more important since in most applications the noise series has heavier than Gaussian tails.

## 12.13 Partial Autocorrelation Coefficients

The partial autocorrelation function (PACF) can be useful for identifying the order of an AR process. The  $k$ th partial autocorrelation, denoted by  $\phi_{k,k}$ , for a stationary process  $Y_t$  is the correlation between  $Y_t$  and  $Y_{t+k}$ , conditional on  $Y_{t+1}, \dots, Y_{t+k-1}$ . For  $k = 1$ ,  $Y_{t+1}, \dots, Y_{t+k-1}$  is an empty set, so the partial autocorrelation coefficient is simply equal to the autocorrelation coefficient, that is,  $\phi_{1,1} = \rho(1)$ . Let  $\hat{\phi}_{k,k}$  denote the estimate of  $\phi_{k,k}$ .  $\hat{\phi}_{k,k}$  can be calculated by fitting the regression model

$$Y_t = \phi_{0,k} + \phi_{1,k}Y_{t-1} + \cdots + \phi_{k,k}Y_{t-k} + \epsilon_{k,t}.$$



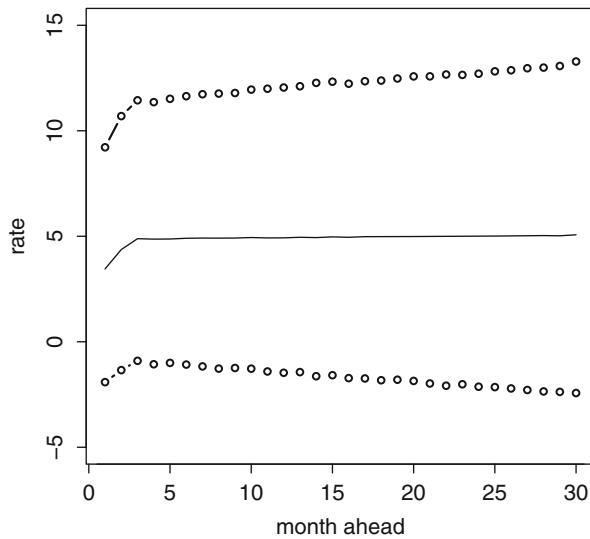
**Fig. 12.21.** Forecast limits of changes in the inflation rate computed by simulation (solid), computed by `arima()` (dotted), and the mean of the forecast (dashed). Notice that the two sets of future limits are very similar and nearly overprint each other, so they are difficult to distinguish visually.

If  $Y_t$  is an AR( $p$ ) process, then  $\phi_{k,k} = 0$  for  $k > p$ . Therefore, a sign that a time series can be fit by an AR( $p$ ) model is that the sample PACF will be nonzero up to  $p$  and then will be nearly zero for larger lags.

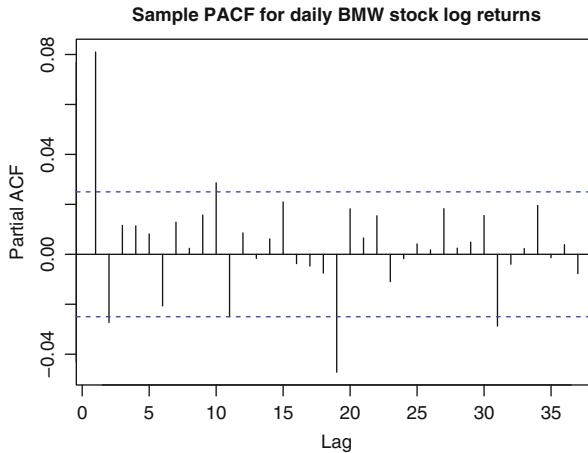
#### Example 12.17. PACF for BMW log returns

Figure 12.23 is the sample PACF for the BMW log returns computed using the R function `pacf()`. The large value of  $\hat{\phi}_{1,1}$  and the smaller values of  $\hat{\phi}_{k,k}$  for  $k = 2, \dots, 9$  are a sign that this time series can be fit by an AR(1) model, in agreement with the results in Example 12.4. Note that  $\hat{\phi}_{k,k}$  is outside the test bounds for some values of  $k > 9$ , particularly for  $k = 19$ . This is likely due to random variation.  $\square$

When computing resources were expensive, the standard practice was to identify a tentative ARMA model using the sample ACF and PACF, fit this model, and then check the ACF and PACF of the residuals. If the residual ACF and PACF revealed some lack of fit, then the model could be enlarged. As computing has become much cheaper and faster and the use of information-based model selection criteria has become popular, this practice has changed. Now many data analysts prefer to start with a relatively large set of models and compare them with selection criteria such as AIC and BIC. This can be done automatically by `auto.arima()` in R or similar functions in other software packages.



**Fig. 12.22.** Forecast limits for the inflation rate computed by simulation.



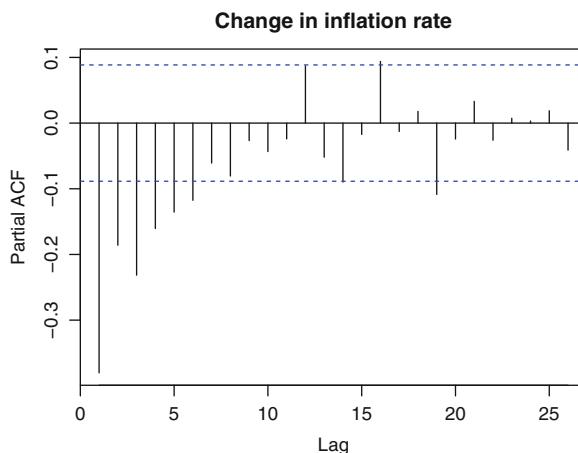
**Fig. 12.23.** Sample PACF for the daily BMW stock log returns.

*Example 12.18.* PACF for changes in the inflation rate

Figure 12.24 is the sample PACF for the changes in the inflation rate. The sample PACF decays slowly to zero, rather than dropping abruptly to zero as for an AR process. This is an indication that this time series should not be fit by a pure AR process. An MA or ARMA process would be preferable. In fact, we saw previously that an MA(2) or MA(3) model provides a parsimonious fit.  $\square$

## 12.14 Bibliographic Notes

There are many books on time series analysis and only a few will be mentioned. Box, Jenkins, and Reinsel (2008) did so much to popularize ARIMA models that these are often called “Box–Jenkins models.” Hamilton (1994) is a comprehensive treatment of time series. Brockwell and Davis (1991) is particularly recommended for those with a strong mathematical preparation wishing to understand the theory of time series analysis. Brockwell and Davis (2003) is a gentler introduction to time series and is suited for those wishing



**Fig. 12.24.** Sample PACF for changes in the inflation rate.

to concentrate on applications. Enders (2004) and Tsay (2005) are time series textbooks concentrating on economic and financial applications; Tsay (2005) is written at a somewhat more advanced level than Enders (2004). Gourieroux and Jasiak (2001) has a chapter on the applications of univariate time series in financial econometrics, and Alexander (2001) has a chapter on time series models. Pfaff (2006) covers both the theory and application of unit root tests.

## 12.15 R Lab

### 12.15.1 T-bill Rates

Run the following code to input the `TbGdpPi.csv` data set and plot the three quarterly time series, as well as their auto- and cross-correlation functions. The last three lines of code run augmented Dickey–Fuller tests on the three series.

```

1 TbGdpPi = read.csv("TbGdpPi.csv", header=TRUE)
2 # r = the 91-day treasury bill rate
3 # y = the log of real GDP
4 # pi = the inflation rate
5 TbGdpPi = ts(TbGdpPi, start = 1955, freq = 4)
6 library(tseries)
7 plot(TbGdpPi)
8 acf(TbGdpPi)
9 adf.test(TbGdpPi[,1])
10 adf.test(TbGdpPi[,2])
11 adf.test(TbGdpPi[,3])

```

### Problem 1

- (a) *Describe the signs of nonstationarity seen in the time series and ACF plots.*
- (b) *Use the augmented Dickey–Fuller tests to decide which of the series are nonstationary. Do the tests corroborate the conclusions of the time series and ACF plots?*

Next run the augmented Dickey–Fuller test on the differenced series and plot the differenced series using the code below. Notice that the `pairs()` function creates a scatterplot matrix, but the `plot()` function applied to time series creates time series plots. [The `plot()` function would create a scatterplot matrix if the data were in a `data.frame` rather than having “class” time series (`ts`). Check the class of `diff_rate` with `attr(diff_rate, "class")`.] Both types of plots are useful. The former shows cross-sectional associations, while the time series plots are helpful when deciding whether differencing once is enough to induce stationarity. You should see that the first-differenced data look stationary.

```

12 diff_rate = diff(TbGdpPi)
13 adf.test(diff_rate[,1])
14 adf.test(diff_rate[,2])
15 adf.test(diff_rate[,3])
16 pairs(diff_rate) # scatterplot matrix
17 plot(diff_rate) # time series plots

```

Next look at the autocorrelation functions of the differenced series. These will be on the diagonal of a  $3 \times 3$  matrix of plots. The off-diagonal plots are cross-correlation functions, which will be discussed in Chap. 13 and can be ignored for now.

```
18 acf(diff_rate) # auto- and cross-correlations
```

**Problem 2**

1. Do the differenced series appear stationary according to the augmented Dickey–Fuller tests?
2. Do you see evidence of autocorrelations in the differenced series? If so, describe these correlations.

For the remainder of this lab, we will focus on the analysis of the 91-day T-bill rate. Since the time series are quarterly, it is good to see whether the mean depends on the quarter. One way to check for such effects is to compare boxplots of the four quarters. The following code does this. Note the use of the `cycle()` function to obtain the quarterly period of each observation; this information is embedded in the data and `cycle()` simply extracts it.

```
19 par(mfrow=c(1,1))
20 boxplot(diff_rate[,1] ~ cycle(diff_rate))
```

**Problem 3** Do you see any seasonal differences in the boxplots? If so, describe them.

Regardless of whether seasonal variation is present, for now we will look at nonseasonal models. Seasonal models are introduced in Sect. 13.1. Next, use the `auto.arima()` function in the `forecast` package to find a “best-fitting” nonseasonal ARIMA model for the T-bill rates. The specifications `max.P=0` and `max.Q=0` force the model to be nonseasonal, since `max.P` and `max.Q` are the number of seasonal AR and MA components.

```
21 library(forecast)
22 auto.arima(TbGdpPi[,1], max.P=0, max.Q=0, ic="aic")
```

**Problem 4**

1. What order of differencing is chosen? Does this result agree with your previous conclusions?
2. What model was chosen by AIC?
3. Which goodness-of-fit criterion is being used here?
4. Change the criterion to BIC. Does the best-fitting model then change?
5. Carefully express the fitted model chosen by the BIC criterion in mathematical notation.

Finally, refit the best-fitting model with the following code, and check for any residual autocorrelation. You will need to replace the three question marks by the appropriate numerical values for the best-fitting model.

```
23 fit1 = arima(TbGdpPi[,1], order=c(?, ?, ?))
24 acf(residuals(fit1))
25 Box.test(residuals(fit1), lag = 12, type="Ljung", fitdf=?)
```

**Problem 5** Do you think that there is residual autocorrelation? If so, describe this autocorrelation and suggest a more appropriate model for the T-bill series.

GARCH effects, that is, volatility clustering, can be detected by looking for auto-correlation in the mean-centered squared residuals. Another possibility is that some quarters are more variable than others. This can be detected for quarterly data by autocorrelation in the squared residuals at time lags that are a multiple of 4. Run the following code to look at autocorrelation in the mean-centered squared residuals.

```
26 resid2 = (residuals(fit1) - mean(residuals(fit1)))^2
27 acf(resid2)
28 Box.test(resid2, lag = 12, type="Ljung")
```

**Problem 6** Do you see evidence of GARCH effects?

### 12.15.2 Forecasting

This example shows how to forecast a time series using R. Run the following code to fit a nonseasonal ARIMA model to the quarterly inflation rate. The code also uses the `predict()` function to forecast 36 quarters ahead. The standard errors of the forecasts are also returned by `predict()` and can be used to create prediction intervals. Note the use of `col` to specify colors. Replace `c(?, ?, ?)` by the specification of the ARIMA model that minimizes BIC.

```
1 TbGdpPi = read.csv("TbGdpPi.csv", header=TRUE)
2 attach(TbGdpPi)
3 # r = the 91-day treasury bill rate
4 # y = the log of real GDP
5 # pi = the inflation rate
6 # fit the non-seasonal ARIMA model found by auto.arima()
7 # quarterly observations from 1955-1 to 2013-4
8 year = seq(1955,2013.75, by=0.25)
9 library(forecast)
10 auto.arima(pi, max.P=0, max.Q=0, ic="bic")
11 fit = arima(pi, order=c(?, ?, ?))
12 forecasts = predict(fit, 36)
13 plot(year,pi,xlim=c(1980,2023), ylim=c(-7,12), type="b")
14 lines(seq(from=2014, by=.25, length=36), forecasts$pred, col="red")
15 lines(seq(from=2014, by=.25, length=36),
16 forecasts$pred + 1.96*forecasts$se, col="blue")
17 lines(seq(from=2014, by=.25, length=36),
18 forecasts$pred - 1.96*forecasts$se, col="blue")
```

**Problem 7** Include the plot with your work.

- (a) Why do the prediction intervals (blue curves) widen as one moves farther into the future?
- (b) Why are the predictions (red) constant throughout?

## 12.16 Exercises

1. This problem and the next use CRSP daily returns. First, get the data and plot the ACF in two ways:

```

1 library(Ecdat)
2 data(CRSPday)
3 crsp=CRSPday[,7]
4 acf(crsp)
5 acf(as.numeric(crsp))

```

- (a) Explain what “lag” means in the two ACF plots. Why does lag differ between the plots?
  - (b) At what values of lag are there significant autocorrelations in the CRSP returns? For which of these values do you think the statistical significance might be due to chance?
2. Next, fit AR(1) and AR(2) models to the CRSP returns:
- ```

6 arima(crsp,order=c(1,0,0))
7 arima(crsp,order=c(2,0,0))

```
- (a) Would you prefer an AR(1) or an AR(2) model for this time series? Explain your answer.
 - (b) Find a 95 % confidence interval for ϕ for the AR(1) model.
3. Consider the AR(1) model

$$Y_t = 5 - 0.55Y_{t-1} + \epsilon_t$$

and assume that $\sigma_\epsilon^2 = 1.2$.

- (a) Is this process stationary? Why or why not?
 - (b) What is the mean of this process?
 - (c) What is the variance of this process?
 - (d) What is the covariance function of this process?
4. Suppose that Y_1, Y_2, \dots is an AR(1) process with $\mu = 0.5$, $\phi = 0.4$, and $\sigma_\epsilon^2 = 1.2$.
 - (a) What is the variance of Y_1 ?
 - (b) What are the covariances between Y_1 and Y_2 and between Y_1 and Y_3 ?
 - (c) What is the variance of $(Y_1 + Y_2 + Y_3)/2$?
 5. An AR(3) model has been fit to a time series. The estimates are $\hat{\mu} = 104$, $\hat{\phi}_1 = 0.4$, $\hat{\phi}_2 = 0.25$, and $\hat{\phi}_3 = 0.1$. The last four observations were $Y_{n-3} = 105$, $Y_{n-2} = 102$, $Y_{n-1} = 103$, and $Y_n = 99$. Forecast Y_{n+1} and Y_{n+2} using these data and estimates.

6. Let Y_t be an MA(2) process,

$$Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}.$$

Find formulas for the autocovariance and autocorrelation functions of Y_t .

7. Let Y_t be a stationary AR(2) process,

$$(Y_t - \mu) = \phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t.$$

- (a) Show that the ACF of Y_t satisfies the equation

$$\rho(k) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2)$$

for all values of $k > 0$. (These are a special case of the Yule–Walker equations.)

[Hint: $\gamma(k) = \text{Cov}(Y_t, Y_{t-k}) = \text{Cov}\{\phi_1(Y_{t-1} - \mu) + \phi_2(Y_{t-2} - \mu) + \epsilon_t, Y_{t-k}\}$ and ϵ_t and Y_{t-k} are independent if $k > 0$.]

- (b) Use part (a) to show that (ϕ_1, ϕ_2) solves the following system of equations:

$$\begin{pmatrix} \rho(1) \\ \rho(2) \end{pmatrix} = \begin{pmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}.$$

- (c) Suppose that $\rho(1) = 0.4$ and $\rho(2) = 0.2$. Find ϕ_1 , ϕ_2 , and $\rho(3)$.

8. Use (12.11) to verify Eq. (12.12).

9. Show that if w_t is defined by (12.34) then (12.35) is true.

10. For a univariate, discrete time process, what is the difference between a strictly stationary process and a weakly stationary process?

11. The time series in the middle and bottom panels of Fig. 12.14 are both nonstationary, but they clearly behave in different manners. The time series in the bottom panel exhibits “momentum” in the sense that once it starts moving upward or downward, it often moves consistently in that direction for a large number of steps. In contrast, the series in the middle panel does not have this type of momentum and a step in one direction is quite likely to be followed by a step in the opposite direction. Do you think the time series model with momentum would be a good model for the price of a stock? Why or why not?

12. The MA(2) model $Y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$ was fit to data and the estimates are

Parameter	Estimate
μ	45
θ_1	0.3
θ_2	-0.15

The last two values of the observed time series and residuals are

t	Y_t	$\hat{\epsilon}_t$
$n - 1$	39.8	-4.3
n	42.7	1.5

Find the forecasts of Y_{n+1} and Y_{n+2} .

13. The ARMA(1,2) model $Y_t = \mu + \phi_1 Y_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$ was fit to data and the estimates are

Parameter	Estimate
μ	103
ϕ_1	0.2
θ_1	0.4
θ_2	-0.25

The last two values of the observed time series and residuals are

t	Y_t	$\hat{\epsilon}_t$
$n - 1$	120.1	-2.3
n	118.3	2.6

Find the forecasts of Y_{n+1} and Y_{n+2} .

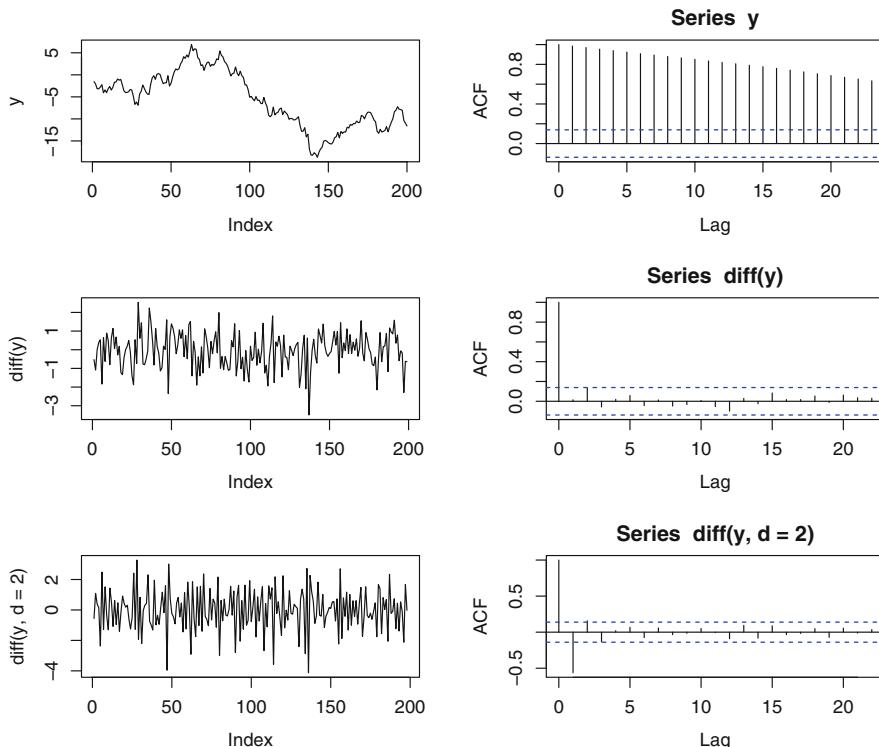
14. To decide the value of d for an ARIMA(p, d, q) model for a time series y , plots were created using the R program:

```

8 par(mfrow=c(3,2))
9 plot(y,type="l")
10 acf(y)
11 plot(diff(y),type="l")
12 acf(diff(y))
13 plot(diff(y,d=2),type="l")
14 acf(diff(y,d=2))

```

The output was the following figure:



What value of d do you recommend? Why?

15. This problem fits an ARIMA model to the logarithms monthly one-month T-bill rates in the data set `Mishkin` in the `Ecdat` package. Run the following code to get the variable:


```

15 library(Ecdat)
16 data(Mishkin)
17 tb1 = log(Mishkin[,3])
      
```

 (a) Use time series and ACF plots to determine the amount of differencing needed to obtain a stationary series.
 (b) Next use `auto.arima` to determine the best-fitting nonseasonal ARIMA models. Use both AIC and BIC and compare the results.
 (c) Examine the ACF of the residuals for the model you selected. Do you see any problems?
16. Suppose you fit an AR(2) model to a time series Y_t , $t = 1, \dots, n$, and the estimates were $\hat{\mu} = 100.1$, $\hat{\phi}_1 = 0.5$, and $\hat{\phi}_2 = 0.1$. The last three observations were $Y_{n-2} = 101.0$, $Y_{n-1} = 99.5$, and $Y_n = 102.3$. What are the forecasts of Y_{n+1} , Y_{n+2} , and Y_{n+3} ?
17. In Sect. 12.9.1, it was stated that “if $E(Y_t)$ has an m th-degree polynomial trend, then the mean of $E(\Delta^d Y_t)$ has an $(m-d)$ th-degree trend for $d \leq m$. For $d > m$, $E(\Delta^d Y_t) = 0$.” Prove these assertions.

References

- Alexander, C. (2001) *Market Models: A Guide to Financial Data Analysis*, Wiley, Chichester.
- Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (2008) *Time Series Analysis: Forecasting and Control*, 4th ed., Wiley, Hoboken, NJ.
- Brockwell, P. J. and Davis, R. A. (1991) *Time Series: Theory and Methods*, 2nd ed., Springer, New York.
- Brockwell, P. J. and Davis, R. A. (2003) *Introduction to Time Series and Forecasting*, 2nd ed., Springer, New York.
- Enders, W. (2004) *Applied Econometric Time Series*, 2nd Ed., Wiley, New York.
- Gouriéroux, C., and Jasiak, J. (2001) *Financial Econometrics*, Princeton University Press, Princeton, NJ.
- Hamilton, J. D. (1994) *Time Series Analysis*, Princeton University Press, Princeton, NJ.
- Pfaff, B (2006) *Analysis of Integrated and Cointegrated Time Series with R*, Springer, New York.
- Tsay, R. S. (2005) *Analysis of Financial Time Series*, 2nd ed., Wiley, New York.

Time Series Models: Further Topics

13.1 Seasonal ARIMA Models

Economic time series often exhibit strong seasonal variation. For example, an investor in mortgage-backed securities might be interested in predicting future housing starts, and these are usually much lower in the winter months compared to the rest of the year. Figure 13.1a is a time series plot of the logarithms of quarterly urban housing starts in Canada from the first quarter of 1960 to final quarter of 2001. The data are in the data set `Hstarts` in R's `Ecdat` package.

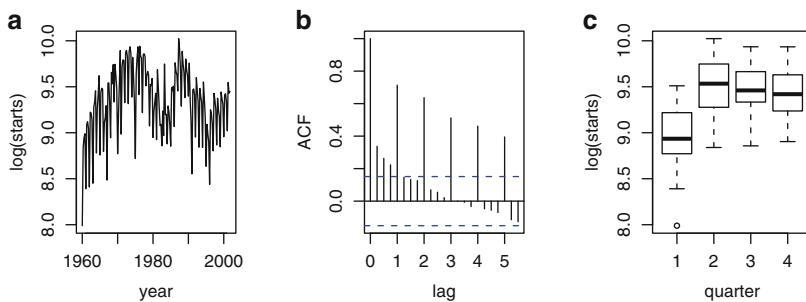


Fig. 13.1. Logarithms of quarterly urban housing starts in Canada: (a) time series plot; (b) sample ACF; (c) boxplots by quarter.

Figure 13.1 shows one and perhaps two types of nonstationarity: (1) There is strong seasonality, and (2) it is unclear whether the seasonal sub-series revert to a fixed mean and, if not, then this is a second type of nonstationarity because the process is integrated. These effects can also be seen in the ACF plot in Fig. 13.1b. At lags that are multiples of four, the autocorrelations

are large, and decay slowly to zero. At other lags, the autocorrelations are smaller but also decay somewhat slowly. The boxplots in Fig. 13.1c give us a better picture of the seasonal effects. Housing starts are much lower in the first quarter than other quarters, jump to a peak in the second quarter, and then drop off slightly in the last two quarters.

Other time series might have only seasonal nonstationarity. For example, monthly average temperatures in a city with a temperate climate will show a strong seasonal effect, but if we plot temperatures for any single month of the year, say July, we will see mean-reversion.

13.1.1 Seasonal and Nonseasonal Differencing

Nonseasonal differencing is the type of differencing that we have been using so far. The series Y_t is replaced by $\Delta Y_t = Y_t - Y_{t-1}$ if the differencing is first-order, and so forth for higher-order differencing. Nonseasonal differencing does not remove seasonal nonstationarity and does not alone create a stationary series; see the top row of Fig. 13.2.

To remove seasonal nonstationarity, one uses seasonal differencing. Let s be the period. For example, $s = 4$ for quarterly data and $s = 12$ for monthly data. Define $\Delta_s = 1 - B^s$ so that $\Delta_s Y_t = Y_t - Y_{t-s}$.

Be careful to distinguish between $\Delta_s = 1 - B^s$ and $\Delta^s = (1 - B)^s$. Note that $\Delta_s = 1 - B^s$ is the first-order seasonal differencing operator and $\Delta^s = (1 - B)^s$ is the s th-order nonseasonal differencing operator. For example, $\Delta_2 Y_t = Y_t - Y_{t-2}$ but $\Delta^2 Y_t = \Delta(\Delta Y_t) = Y_t - 2Y_{t-1} + Y_{t-2}$.

The series $\Delta_s Y_t$ is called the seasonally differenced series. See the middle row of Fig. 13.2 for the seasonally differenced logarithm of housing starts and its Sample ACF.

One can combine seasonal and nonseasonal differencing by using, for example, for first-order differences

$$\Delta(\Delta_s Y_t) = \Delta(Y_t - Y_{t-s}) = (Y_t - Y_{t-s}) - (Y_{t-1} - Y_{t-s-1}).$$

The order in which the seasonal and nonseasonal difference operators are applied does not matter, since one can show that

$$\Delta(\Delta_s Y_t) = \Delta_s(\Delta Y_t).$$

For a seasonal time series, seasonal differencing is necessary, but whether also to use nonseasonal differencing will depend on the particular time series. For the housing starts data, the seasonally differenced series appears stationary so only seasonal differencing is absolutely needed, but combining seasonal and nonseasonal differencing might be preferred since it results in a simpler model.

13.1.2 Multiplicative ARIMA Models

One of the simplest seasonal models is the ARIMA $\{(1, 1, 0) \times (1, 1, 0)_s\}$ model, which puts together the nonseasonal ARIMA(1,1,0) model

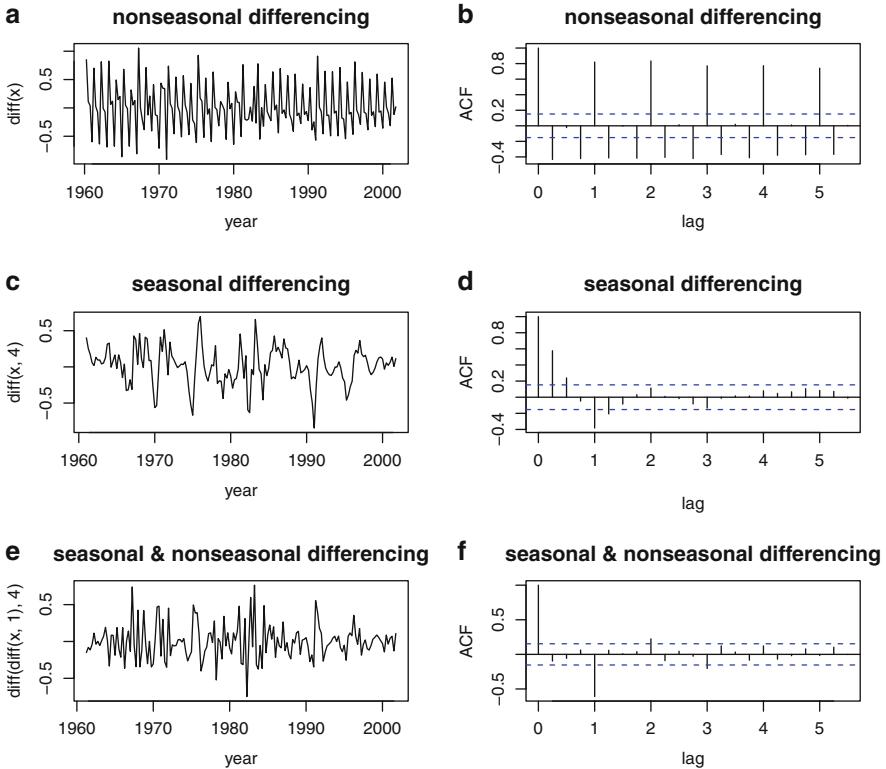


Fig. 13.2. Time series (left column) and sample ACF plots (right column) of the logarithms of quarterly urban housing starts with nonseasonal differencing (top row), seasonal differencing (middle row), and both seasonal and nonseasonal differencing (bottom row). Note: in the sample ACF plots, lag = 1 means a lag of one year, which is four observations for quarterly data.

$$(1 - \phi B)(\Delta Y_t - \mu) = \epsilon_t \quad (13.1)$$

and a purely seasonal ARIMA(1,1,0)_s model

$$(1 - \phi^* B^s)(\Delta_s Y_t - \mu) = \epsilon_t \quad (13.2)$$

to obtain the multiplicative model

$$(1 - \phi B)(1 - \phi^* B^s)\{\Delta(\Delta_s Y_t) - \mu\} = \epsilon_t. \quad (13.3)$$

Model (13.2) is called “purely seasonal” and has a subscript “s” since it uses only B^s and Δ_s ; it is obtained from the ARIMA(1,1,0) by replacing B and Δ by B^s and Δ_s . For a monthly time series ($s = 12$), model (13.2) gives 12 independent processes, one for Januaries, a second for Februaries, and so forth. Model (13.3) uses the components from (13.1) to tie these 12 series together.

The ARIMA $\{(p, d, q) \times (p_s, d_s, q_s)_s\}$ process is

$$\begin{aligned} & (1 - \phi_1 B - \cdots - \phi_p B^p) \{1 - \phi_1^* B^s - \cdots - \phi_{p_s}^* (B^s)^{p_s}\} \{\Delta^d (\Delta_s^{d_s} Y_t) - \mu\} \\ & = (1 + \theta_1 B + \cdots + \theta_q B^q) \{1 + \theta_1^* B^s + \cdots + \theta_{q_s}^* (B^s)^{q_s}\} \epsilon_t. \end{aligned} \quad (13.4)$$

This process multiplies together the AR components, the MA components, and the differencing components of two processes: the nonseasonal ARIMA (p, d, q) process

$$(1 - \phi_1 B - \cdots - \phi_p B^p) \{(\Delta^d Y_t) - \mu\} = (1 + \theta_1 B + \cdots + \theta_q B^q) \epsilon_t$$

and the seasonal ARIMA $(p_s, d_s, q_s)_s$ process

$$\{1 - \phi_1^* B^s - \cdots - \phi_{p_s}^* (B^s)^{p_s}\} \{(\Delta_s^{d_s} Y_t) - \mu\} = \{1 + \theta_1^* B^s + \cdots + \theta_{q_s}^* (B^s)^{q_s}\} \epsilon_t.$$

Example 13.1. ARIMA $\{(1, 1, 1) \times (0, 1, 1)_4\}$ model for housing starts

We return to the housing starts data. The first question is whether to difference only seasonally, or both seasonally and nonseasonally. The seasonally differenced quarterly series in the middle row of Fig. 13.2 is possibly stationary, so perhaps seasonal differencing is sufficient. However, the ACF of the seasonally and nonseasonally differenced series in the bottom row has a simpler ACF than the data that are only seasonally differenced. By differencing both ways, we should be able find a more parsimonious ARMA model.

Two models with seasonal and nonseasonal differencing were tried, ARIMA $\{(1, 1, 1) \times (1, 1, 1)_4\}$ and ARIMA $\{(1, 1, 1) \times (0, 1, 1)_4\}$. Both provided good fits and had residuals that passed the Ljung–Box test. The second of the two models was selected, because it has one fewer parameter than the first, though the other model would have been a reasonable choice. The results from fitting the chosen model are below.

```

1 data(Hstarts, package="Ecdat")
2 x = ts(Hstarts[,1], start=1960, frequency=4)
3 fit2 = arima(x, c(1,1,1), seasonal = list(order = c(0,1,1),
4   period = 4))
5 fit2

Call:
arima(x = hst, order = c(1, 1, 1), seasonal
= list(order = c(0, 1, 1), period = 4))

Coefficients:
      ar1      ma1     sma1
      0.675   -0.890   -0.822
  s.e.  0.142    0.105    0.051

sigma^2 estimated as 0.0261: log-likelihood = 62.9,
  aic = -118

```

Thus, the fitted model is

$$(1 - 0.675 B)Y_t^* = (1 - 0.890 B)(1 - 0.822 B_4) \epsilon_t$$

where $Y_t^* = \Delta(\Delta_4 Y_t)$ and ϵ_t is white noise with mean zero and variance 0.0261. Figure 13.3 shows forecasts from this model for the four years following the end of the time series. \square

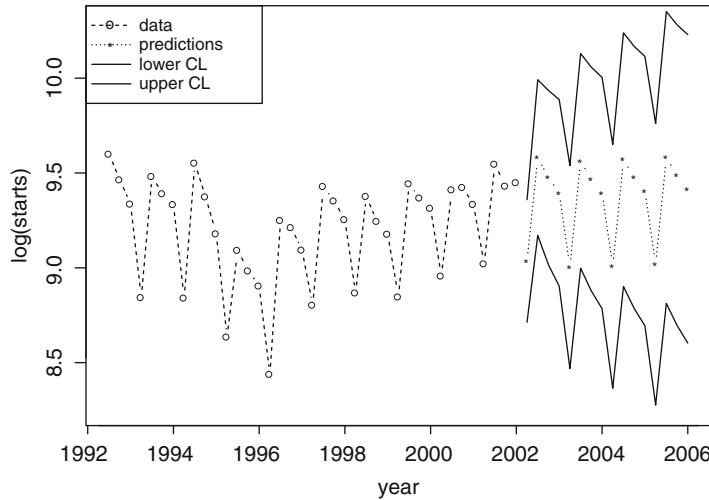


Fig. 13.3. Forecasting logarithms of quarterly urban housing starts using the ARIMA $\{(1, 1, 1) \times (0, 1, 1)_4\}$ model. The dashed line connects the data, the dotted line connects the forecasts, and the solid lines are the forecast limits.

When the size of the seasonal oscillations increases, as with the air passenger data in Fig. 12.2, some type of preprocessing is needed before differencing. Often, taking logarithms stabilizes the size of the oscillations. This can be seen in Fig. 13.4. Box, Jenkins, and Reinsel (2008) obtain a parsimonious fit to the log passengers with an ARIMA $(0, 1, 1) \times (0, 1, 1)_{12}$ model.

For the housing starts series, the data come as logarithms in the Ecdat package. If they had come untransformed, then we would have needed to apply some type of transformation.

13.2 Box–Cox Transformation for Time Series

As just discussed, it is often desirable to transform a time series to stabilize the size of the variability, both seasonal and random. Although a transformation can be selected by trial-and-error, another possibility is automatic selection by maximum likelihood estimation using the model

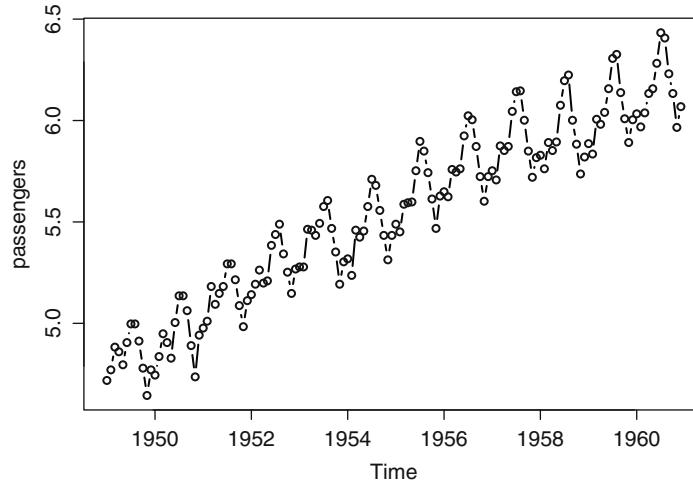


Fig. 13.4. Time series plot of the logarithms of the monthly total international airline passengers (in thousands).

$$(\Delta^d Y_t^{(\alpha)} - \mu) = \phi_1(\Delta^d Y_{t-1}^{(\alpha)} - \mu) + \cdots + \phi_p(\Delta^d Y_{t-p}^{(\alpha)} - \mu) + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}, \quad (13.5)$$

where $\epsilon_1, \epsilon_2, \dots$ is Gaussian white noise. Model (13.5) states that after a Box–Cox transformation, Y_t follows an ARIMA model with Gaussian noise that has a constant variance. The transformation parameter α is considered unknown and is estimated by maximum likelihood along with the AR and MA parameters and the noise variance. For notational simplicity, (13.5) uses a nonseasonal model, but a seasonal ARIMA model could just as easily have been used.

Example 13.2. Selecting a transformation for the housing starts

Figure 13.5 show the profile likelihood for α for the housing starts series (not the logarithms). The ARIMA model was $\text{ARIMA}\{(1, 1, 1) \times (1, 1, 1)_4\}$. The figure was created by the `BoxCox.Arima()` function in R's `FitAR` package. This function denotes the transformation parameter by λ . The MLE of α is 0.34 and the 95 % confidence interval is roughly from 0.15 to 0.55. Thus, the log transformation ($\alpha = 0$) is somewhat outside the confidence interval, but the square-root transformation is in the interval. Nonetheless, the log transformation worked satisfactorily in Example 13.1 and might be retained.

Without further analysis, it is not clear why $\alpha = 0.34$ achieves a better fit than the log transformation. Better fit could mean that the ARIMA model fits better, that the noise variability is more nearly constant, that the noise is closer to being Gaussian, or some combination of these effects.

It would be interesting to compare forecasts using the log and square-root transformations to see in what ways, if any, the square-root transformation outperforms the log transformation for forecasting. The forecasts would need to be back-transformed to the original scale in order for them to be comparable. One might use the final year as test data to see how well housing starts in that year are forecast. \square

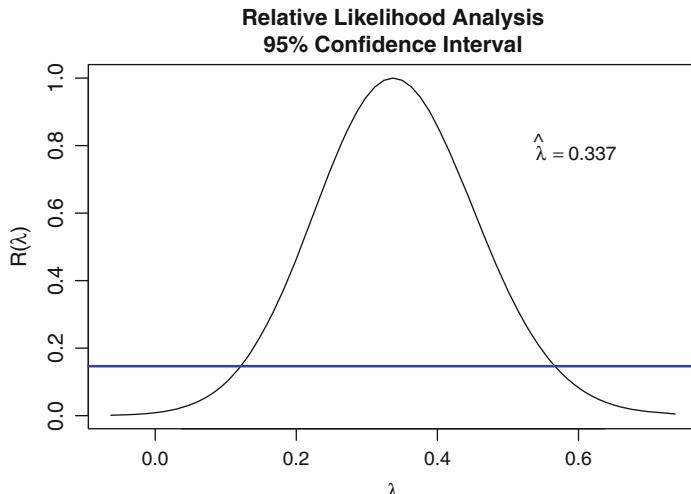


Fig. 13.5. Profile likelihood for α (called λ in the legend) in the housing start example. Values of λ with $R(\lambda)$ (the profile likelihood) above the horizontal line are in the 95 % confidence limit.

Data transformations can stabilize some types of variation in time series, but not all types. For example, in Fig. 12.2 the seasonal oscillations in the numbers of air passengers increase as the series itself increases, and we can see in Fig. 13.4 that a log transformation stabilizes these oscillations. In contrast, the S&P 500 returns in Fig. 4.1 exhibit periods of low and high volatility even though the returns maintain a mean near 0. Transformations cannot remove this type of volatility clustering. Instead, these changes of volatility could be modeled by a GARCH process; this topic is pursued in Chap. 14.

13.3 Time Series and Regression

In a multiple linear regression model (9.1) the errors ϵ_i are assumed to be mutually independent. However, if the data $\{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$ are time series, then it is likely that the errors are correlated, a problem we will call *residual correlation*; see Sect. 13.3.1.

Residual correlation causes standard errors and confidence intervals (which incorrectly assume uncorrelated noise) to be incorrect. In particular, the coverage probability of confidence intervals can be much lower than the nominal value. A solution to this problem is to adjust or correct the estimated covariance matrix of the coefficient estimates; see Sect. 13.3.2. An alternative solution is to model the noise as an ARMA process, assuming that the residuals are stationary; see Sect. 13.3.3.

13.3.1 Residual Correlation and Spurious Regressions

In the extreme case where the residuals are an integrated process, the least-squares estimator is inconsistent, meaning that it will not converge to the true parameter as the sample size converges to ∞ . If an $I(1)$ process is regressed on another $I(1)$ process and the two processes are independent (so that the regression coefficient is 0), it is quite possible to obtain a highly significant result, that is, to strongly reject the true null hypothesis that the regression coefficient is 0. This is called a *spurious regression*. The problem, of course, is that the test is based on the incorrect assumption of independent error. The residuals from the The problem of correlated noise can be detected by looking at the sample ACF of the residuals. Sometimes the presence of residual correlation is obvious. In other cases, it is not so clear and a statistical test is desirable. The Durbin–Watson test can be used to test the null hypothesis of no residual autocorrelation. More precisely, the null hypothesis of the Durbin–Watson test is that the first p autocorrelation coefficients are all 0, where p can be selected by the user. The p -value for a Durbin–Watson test is not trivial to compute, and different implementations use different computational methods. In the R function `durbinWatsonTest()` in the `car` package, p is called `max.lag` and has a default value of 1. The p -value is computed by `durbinWatsonTest()` using bootstrapping. The `lmtest` package of R has another function, `dwttest()`, that computes the Durbin–Watson test, but only with $p = 1$. The function `dwttest()` uses either a normal approximation (default) or an exact algorithm to calculate the p -value.

Example 13.3. Residual plots for weekly interest changes

Using the interest rate data from Chap. 9, Fig. 13.6 contains residual plots for the regression of `aaa_dif` on `cm10_dif` and `cm30_dif`. The normal plot in panel (a) shows heavy tails. A t -distribution was fit to the residuals, and the estimated degrees of freedom was 2.99, again indicating heavy tails. Panel (b) shows a QQ plot of the residuals and the quantiles of the fitted t -distribution with a 45° reference line. There is excellent agreement between the data and the t -distribution.

Panel (c) is a plot of the ACF of the residuals. There is some evidence of autocorrelation. The Durbin–Watson test was performed three times with

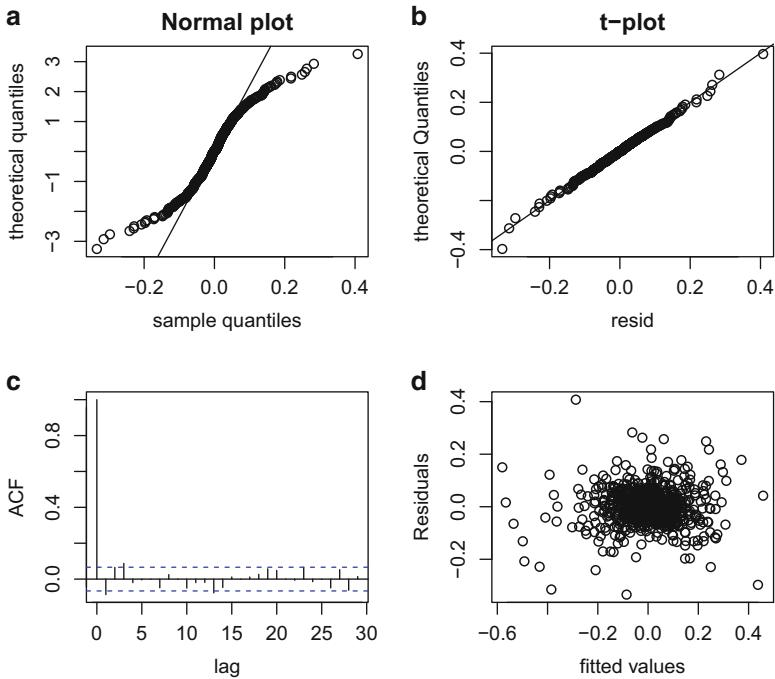


Fig. 13.6. Residual plots for the regression of `aaa_dif` on `cm10_dif` and `cm30_dif`.

R's `durbinWatsonTest()` using `max.lag = 1` and gave p -values of 0.006, 0.004, and 0.012. This shows the substantial random variation due to bootstrapping with the default of $B = 1000$ resamples. Using a larger number of resamples will compute the p -value with more accuracy. For example, when the number of resamples was increased to 10,000, three p -values were 0.0112, 0.0096, and 0.0106. Using `dwtest()`, the approximate p -value was 0.01089 and the exact p -value could not be computed. Despite some uncertainty about the p -value, it is clear that the p -value is small, so there is at least some residual autocorrelation.

To further investigate autocorrelation, ARMA models were fit to the residuals using the `auto.arima()` function in R to automatically select the order. Using BIC, the selected model is ARIMA(0,0,0), that is, white noise. Using AIC, the selected model is ARIMA(0,0,3) with estimates:

```
6 auto.arima(resid, ic="aic")

Series: resid
ARIMA(0,0,3) with zero mean

Coefficients:
      ma1      ma2      ma3
-0.0857  0.0770  0.0888
```

```
s.e. 0.0336 0.0338 0.0342

sigma^2 estimated as 0.004075: log likelihood=1172.54
AIC=-2337.09 AICc=-2337.04 BIC=-2317.97
```

Several of the coefficients are large relative to their standard errors. There is evidence of some autocorrelation, but not a great deal and the BIC-selected model does not have any autocorrelation. The sample size is 880, so there are enough data to detect small autocorrelations. The autocorrelation that was found seems of little practical significance and could perhaps be ignored; see Sect. 13.3.2 for further investigation. The plot of residuals versus fitted values in panel (d) shows no sign of heteroskedasticity. \square

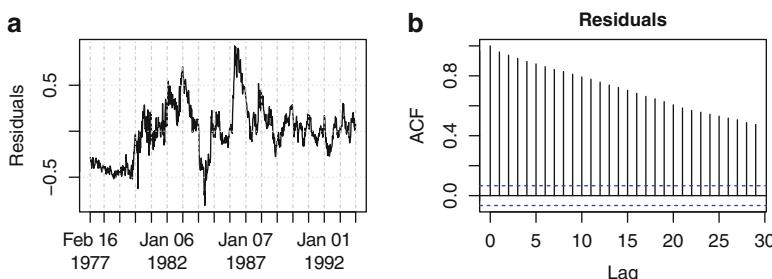


Fig. 13.7. Time series plot and ACF plot of residuals when `aaa` is regressed on `cm10` and `cm30`. The plots indicate that the residuals are nonstationary.

Example 13.4. Residual plots for weekly interest rates without differencing

The reader may have noticed that differenced time series have been used in the examples. There is a good reason for this. Many, if not most, financial time series are nonstationary or, at least, have very high and long-term autocorrelation. When one nonstationary series is regressed upon another, it happens frequently that the residuals are nonstationary. This is a substantial violation of the assumption of uncorrelated noise and can lead to serious problems. An estimator is said to be consistent if it converges to the true value of the parameter as the sample size increases to ∞ . The least-squares estimator is not consistent when the errors are an integrated process.

As an example, we regressed `aaa` on `cm10` and `cm30`. These are the weekly time series of AAA, 10-year Treasury, and 30-year Treasury interest rates, which, when differenced, gave us `aaa_dif`, `cm10_dif`, and `cm30_dif` used in previous examples. Figure 13.7 contains time series and ACF plots of the residuals. The residuals are very highly correlated and perhaps are nonstationary. Unit root tests provide more evidence that the residuals are nonstationary. The *p*-values of augmented Dickey–Fuller tests are on one side of 0.05 or the

other, depending on the order. With the default lag order in the `adf.test()` function from the `tseries` package in R, the p -value is 0.12, so one would not reject the null hypothesis of nonstationarity at level 0.05 or even level 0.1. The `kpsstest()` function does reject the null hypothesis of stationarity.

Let us compare the estimates from regression using the original series with the estimates from the differenced series. First, what should we expect when we make this comparison? Suppose that X_t and Y_t are time series following the regression model

$$Y_t = \alpha + \beta_0 t + \beta_1 X_i + \epsilon_t. \quad (13.6)$$

Note the linear time trend $\beta_0 t$. Then, upon differencing, we have

$$\Delta Y_t = \beta_0 + \beta_1 \Delta X_i + \Delta \epsilon_t, \quad (13.7)$$

so the original intercept α is removed, and the time trend's slope β_0 in (13.6) becomes an intercept in (13.7). The time trend could be omitted in (13.6) if the intercept in (13.7) is not significant, as happens in this example. The slope β_1 in (13.6) remains unchanged in (13.7). However, if ϵ_t is $I(1)$, then the regression of Y_t on X_t will not provide a consistent estimate of β_1 , but the regression of ΔY_t on ΔX_i will consistently estimate β_1 , so the estimates from the two regressions could be very different. This is what happens with this example.

The results from regression with the original series without the time trend are

```
Call:
lm(formula = aaa ~ cm10 + cm30)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.9803    0.0700 14.00 < 2e-16 ***
cm10        0.3183    0.0445  7.15 1.9e-12 ***
cm30        0.6504    0.0498 13.05 < 2e-16 ***
```

The results with the differenced series are

```
Call:
lm(formula = aaa_dif ~ cm10_dif + cm30_dif)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.38e-05  2.18e-03 -0.04      0.97
cm10_dif     3.60e-01  4.45e-02   8.09 2.0e-15 ***
cm30_dif     2.97e-01  4.98e-02   5.96 3.7e-09 ***
```

The estimated slopes for `cm10` and `cm10_dif`, 0.3183 and 0.360, are somewhat similar. However, the estimated slopes for `cm30` and `cm30_dif`, 0.650 and 0.297, are quite dissimilar relative to their standard errors. This is to

be expected if the estimators using the undifferenced series are not consistent; also, their standard errors are not valid because they are based on the assumption of uncorrelated noise. In the analysis with the differenced data, the p -value for the intercept is 0.97, so we can accept the null hypothesis that the intercept is zero; this justifies the omission of the time trend when using the undifferenced series. \square

Example 13.5. Simulated independent AR processes

To illustrate further the problems caused by regressing nonstationary series, or even stationary series with high correlation, we simulated two independent AR process, both of length 200 with $\phi = 0.99$.

```
7 set.seed(997711)
8 n = 200
9 x = arima.sim(list(order=c(1,0,0),ar=.99),n=n)
10 y = arima.sim(list(order=c(1,0,0),ar=.99),n=n)
11 fit1 = lm(y~x)
12 fit5 = lm(diff(y)~diff(x))
```

These processes are stationary but near the borderline of being nonstationary. After simulating these processes, one process was regressed on the other. We repeated this three more times. Since the processes are independent, the true slope is 0. In each case, the estimated slope was far from the true value of 0 and was statistically significant according to the (incorrect) p -value. The results are below.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.40	0.269	-31	1.9e-78
x	0.48	0.036	13	1.6e-29
<hr/>				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.96	0.328	18.2	4.9e-44
x	-0.43	0.088	-4.8	2.6e-06
<hr/>				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-5.154	0.213	-24.2	4.5e-61
x	0.095	0.031	3.1	2.3e-03
<hr/>				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.51	0.312	-1.6	1.1e-01
x	-0.53	0.079	-6.7	2.3e-10

Notice how the estimated intercepts and slope randomly vary between the four simulations. The standard errors and p -values are based on the invalid assumption of independent errors and are erroneous and very misleading, a problem that is called *spurious regression*. Fortunately, the violation of the independence assumption would be easy to detect by plotting the residuals.

We also regressed the differenced series and obtained completely different results:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.082	0.069	1.18	0.24
diff(x)	-0.023	0.068	-0.34	0.73

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.027	0.064	-0.41	0.68
diff(x)	-0.021	0.063	-0.33	0.74

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.015	0.071	-0.21	0.83
diff(x)	-0.022	0.076	-0.29	0.77

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.025	0.077	-0.32	0.75
diff(x)	0.022	0.078	0.28	0.78

Notice that now the estimated slopes are all near the true value of 0. All the p-values are large and lead one to the correct conclusion that the true slope is 0.

When the noise process is stationary, an alternative to differencing is to use an ARMA model for the noise process; see Sect. 13.3.3. \square

13.3.2 Heteroscedasticity and Autocorrelation Consistent (HAC) Standard Errors

We now consider the effect of correlated noise and heteroskedasticity on standard errors and confidence intervals in multiple linear regression models. If $\text{COV}(\boldsymbol{\epsilon}) \neq \sigma_\epsilon^2 \mathbf{I}$ but rather $\text{COV}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}_\epsilon$ for some matrix $\boldsymbol{\Sigma}_\epsilon$, then

$$\begin{aligned}\text{COV}(\hat{\boldsymbol{\beta}} | \mathbf{x}_1, \dots, \mathbf{x}_n) &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \text{COV}(\mathbf{Y} | \mathbf{x}_1, \dots, \mathbf{x}_n) \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \\ &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\Sigma}_\epsilon \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1}. \end{aligned} \quad (13.8)$$

This result lets us see the effect of correlation or nonconstant variance among $\epsilon_1, \dots, \epsilon_n$.

Example 13.6. Regression with AR(1) errors

Suppose that $\epsilon_1, \dots, \epsilon_n$ is a stationary AR(1) process so that $\epsilon_t = \phi \epsilon_{t-1} + u_t$, where $|\phi| < 1$ and u_1, u_2, \dots is weak WN($0, \sigma_u^2$). Then

$$\boldsymbol{\Sigma}_\epsilon = \sigma_\epsilon^2 \begin{pmatrix} 1 & \phi & \phi^2 & \cdots & \phi^{n-1} \\ \phi & 1 & \phi & \cdots & \phi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{n-1} & \phi^{n-2} & \phi^{n-3} & \cdots & 1 \end{pmatrix}. \quad (13.9)$$

As an example, suppose that $n = 21$, X_1, \dots, X_n are equally spaced between -10 and 10 , and $\sigma_\epsilon^2 = 1$. Substituting (13.9) into (13.8) gives the covariance matrix of the estimator $(\hat{\beta}_0, \hat{\beta}_1)$, and taking the square roots of the diagonal elements gives the standard errors. This was done with $\phi = -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75$.

Figure 13.8 plots the ratios of standard errors for the independent case ($\phi = 0$) to the standard errors for the true value of ϕ . These ratios are the factors by which the standard errors are miscalculated if we assume that $\phi = 0$, but it is not. Notice that negative values of ϕ result in a conservative (too large) standard error, but positive values of ϕ give a standard error that is too small. In the case of $\phi = 0.75$, assuming independence gives standard errors that are only about half as large as they should be. As discussed in Sect. 13.3.3, this problem can be fixed by assuming (correctly) that the noise process is AR(1). \square

As discussed in Sect. 13.3.1, if the errors in a regression model are an integrated process, such as a random walk, the least-squares estimator is inconsistent. However, when the dependence between the errors is not too strong, there are mild conditions under which the least-squares estimator is consistent, meaning that it will converge to the true parameter as the sample size converges to ∞ . For the latter case there are methods available to estimate consistent standard errors for the coefficient estimates. Two simple and widely used approaches are the heteroskedasticity consistent (HC) and heteroskedasticity and autocorrelation consistent (HAC) estimators.

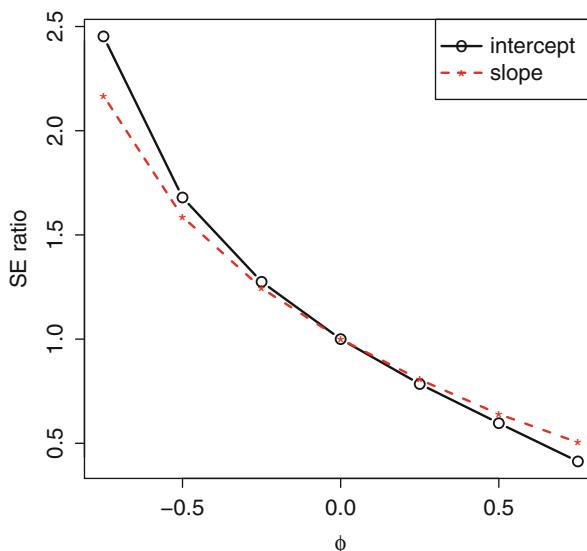


Fig. 13.8. Factor by which the standard error is changed when ϕ deviates from 0 for intercept (solid) and slope (dashed).

Let $\hat{\epsilon}_i$ denote the OLS residuals, let $\widehat{\Sigma}_{\hat{\epsilon}} = \text{diag}\{\hat{\epsilon}_1^2, \dots, \hat{\epsilon}_n^2\}$ denote a diagonal matrix of the squared residuals, and let $\widehat{\mathbf{C}}_{HC} = \mathbf{X}^\top \widehat{\Sigma}_{\hat{\epsilon}} \mathbf{X}$. Then a heteroskedasticity consistent (HC) estimator (see White, 1980) of the covariance matrix for the coefficient estimates is

$$\widehat{\text{COV}}_{HC}(\hat{\beta} | \mathbf{x}_1, \dots, \mathbf{x}_n) = (\mathbf{X}^\top \mathbf{X})^{-1} \widehat{\mathbf{C}}_{HC} (\mathbf{X}^\top \mathbf{X})^{-1}. \quad (13.10)$$

The corresponding HC standard errors are defined as the square roots of the diagonal entries of (13.10).

A heteroskedasticity and autocorrelation consistent (HAC) estimator (see Newey and West, 1987) of the covariance matrix for the coefficient estimates is similarly defined as

$$\widehat{\text{COV}}_{HAC}(\hat{\beta} | \mathbf{x}_1, \dots, \mathbf{x}_n) = (\mathbf{X}^\top \mathbf{X})^{-1} \widehat{\mathbf{C}}_{HAC} (\mathbf{X}^\top \mathbf{X})^{-1}, \quad (13.11)$$

in which

$$\widehat{\mathbf{C}}_{HAC} = \widehat{\mathbf{C}}_{HC} + \sum_{\ell=1}^L w_\ell \sum_{i=\ell+1}^n \left(\mathbf{X}_i \hat{\epsilon}_i \hat{\epsilon}_{i-\ell} \mathbf{X}_{i-\ell}^\top + \mathbf{X}_{i-\ell} \hat{\epsilon}_{i-\ell} \hat{\epsilon}_i \mathbf{X}_i^\top \right), \quad (13.12)$$

where $w_\ell = 1 - \ell/(L+1)$ denotes the Bartlett weight function, although other weight functions w_ℓ can also be used. The corresponding HAC standard errors are defined as the square root of the diagonal entries of (13.11).

Example 13.7. HC and HAC estimates for regression of weekly interest changes

In Sect. 13.3.1 a regression of `aaa_dif` on `cm10_dif` and `cm30_dif` produced residuals that exhibited minor autocorrelation; AIC suggested an MA(3) model for the residuals while BIC selected ARIMA(0,0,0), i.e., white noise. We now consider whether ignoring the small autocorrelations has a practical impact on inference. The previous regression results are obtained from the following R commands.

```
13 dat = read.table(file="WeekInt.txt", header=T)
14 attach(dat)
15 cm10_dif = diff(cm10)
16 aaa_dif = diff(aaa)
17 cm30_dif = diff(cm30)
18 fit = lm(aaa_dif ~ cm10_dif + cm30_dif)
19 round(summary(fit)$coef, 4)
```

The HC and HAC covariance matrix estimates can be computed using the `NeweyWest()` function from the R package `sandwich`. The first argument is a fitted model object, in this case `fit`. In both cases we set `prewhite = F`. For the HAC estimate, the argument `lag` corresponds to the maximal lag L used in the Bartlett weight function above. If no value is specified, one is selected automatically via the `bwNeweyWest()` function (see the help file for more information). For the HC estimate we specify `lag = 0`. The HC estimate and HAC estimate with $L = 3$ are shown below.

```

20 library(sandwich)
21 options(digits=2)
22 NeweyWest(fit, lag = 0, prewhite = F)

  (Intercept) cm10_dif cm30_dif
(Intercept) 4.7e-06 7.3e-06 -1.1e-05
cm10_dif    7.3e-06 6.3e-03 -6.2e-03
cm30_dif   -1.1e-05 -6.2e-03  6.7e-03

23 NeweyWest(fit, lag = 3, prewhite = F)

  (Intercept) cm10_dif cm30_dif
(Intercept) 4.6e-06 -0.00003 2.6e-05
cm10_dif   -3.0e-05  0.00666 -6.6e-03
cm30_dif    2.6e-05 -0.00662 7.0e-03

```

The OLS regression results, as well as the HC and HAC estimated standard errors, and their corresponding t values are summarized in Table 13.1. Recall that the HC and HAC standard error estimates are computed as the square roots of the diagonal entries of the covariance matrix estimates.

```

24 sqrt(diag(NeweyWest(fit, lag = 0, prewhite = F)))
25 sqrt(diag(NeweyWest(fit, lag = 3, prewhite = F)))

```

The corresponding t values are the OLS coefficient estimates divided by their standard error estimates.

```

26 coef(fit)/sqrt(diag(NeweyWest(fit, lag = 0, prewhite = F)))
27 coef(fit)/sqrt(diag(NeweyWest(fit, lag = 3, prewhite = F)))

```

Table 13.1. Regression estimates of `aaa_dif` on `cm10_dif` and `cm30_dif`: the OLS estimates, estimated standard errors, and t values are shown on the left; the estimated HC standard errors and corresponding t values are shown in the middle; and the estimated HAC standard errors with $L = 3$ and corresponding t values are shown on the right.

Coefficient	OLS			HC			HAC _{L=3}	
	Estimate	Std. Err.	t value	Std. Err.	t value	Std. Err.	t value	
(Intercept)	-0.0001	0.0022	-0.043	0.0022	-0.043	0.0021	-0.044	
cm10_dif	0.3602	0.0445	8.091	0.0791	4.553	0.0816	4.415	
cm30_dif	0.2968	0.0498	5.956	0.0816	3.637	0.0836	3.551	

From Table 13.1 we see that the HC and HAC estimates produced similar results. The estimated standard error for the intercept are stable, while the estimated HC and HAC standard errors for the `cm10_dif` and `cm30_dif` coefficients are about twice as large as the OLS estimates of the standard errors, and as a result, the corresponding t values are about half as large in magnitude. In this case, however, the `cm10_dif` and `cm30_dif` coefficients remain statistically significant, with both estimates over three standard errors above zero. The minor serial correlation (and heteroskedasticity) in the OLS residuals does not appear to have a practical impact on inference in this example. \square

13.3.3 Linear Regression with ARMA Errors

When residual analysis shows that the residuals are correlated, then one of the key assumptions of the linear model does not hold, and tests and confidence intervals based on this assumption are invalid and cannot be trusted. Fortunately, there is a solution to this problem: replace the assumption of independent noise by the weaker assumption that the noise process is stationary but possibly correlated. One could, for example, assume that the noise is an ARMA process. This is the strategy we will discuss in this section; this approach is referred to as an ARMAX model, in which the X indicates the inclusion of exogenous regression variables.

The linear regression model with ARMA errors combines the linear regression model (9.1) and the ARMA model (12.26) for the noise, so that

$$Y_t = \beta_0 + \beta_1 X_{t,1} + \cdots + \beta_p X_{t,p} + \epsilon_t, \quad (13.13)$$

where

$$(1 - \phi_1 B - \cdots - \phi_p B^p) \epsilon_t = (1 + \theta_1 B + \cdots + \theta_q B^q) u_t, \quad (13.14)$$

and u_1, \dots, u_n is white noise.

Example 13.8. Demand for ice cream

This example uses the data set `Icecream` in R's `Ecdat` package. The data are four-weekly observations from March 18, 1951, to July 11, 1953 on four variables, `cons` = U.S. consumption of ice cream per head in pints; `income` = average family income per week (in U.S. Dollars); `price` = price of ice cream (per pint); and `temp` = average temperature (in Fahrenheit). There is a total of 30 observations. Since there are 13 four-week periods per year, there are slightly over two years of data.

First, a linear model was fit with `cons` as the response and `income`, `price`, and `temp` as the predictor variables. One can see that `income` and `temp` are significant, especially `temp` (not surprisingly).

```
Call:
lm(formula = cons ~ income + price + temp, data = Icecream)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.06530 -0.01187  0.00274  0.01595  0.07899 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.197315  0.270216   0.73   0.472    
income      0.003308  0.001171   2.82   0.009 **  
price       -1.044414  0.834357  -1.25   0.222    

```