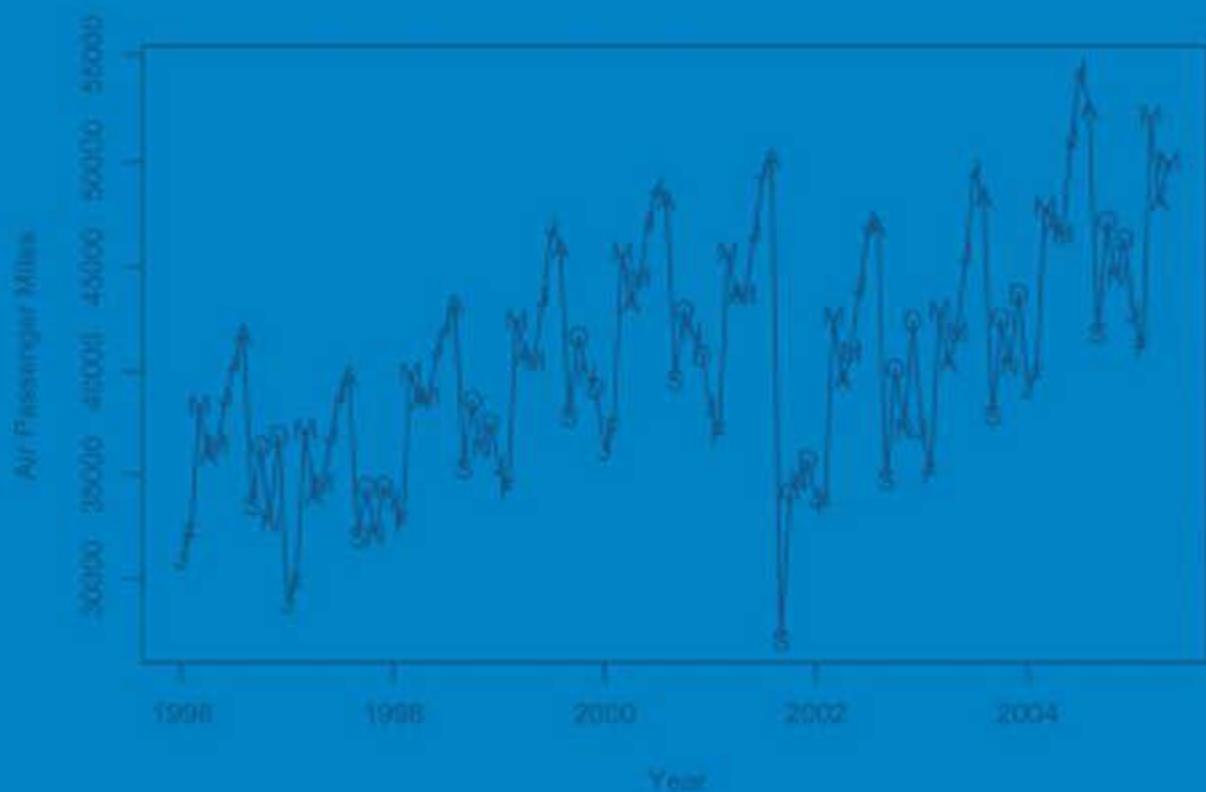


Springer Texts in Statistics

Jonathan D. Cryer
Kung-Sik Chan

Time Series Analysis

With Applications in R



Second Edition

 Springer

Statistics Texts in Statistics

Series Editors:

G. Casella

S. Fienberg

I. Olkin

Springer Texts in Statistics

- Athreya/Lahiri*: Measure Theory and Probability Theory
Bilodeau/Brenner: Theory of Multivariate Statistics
Brockwell/Davis: An Introduction to Time Series and Forecasting
Carmona: Statistical Analysis of Financial Data in S-PLUS
Chow/Teicher: Probability Theory: Independence, Interchangeability, Martingales, 3rd ed.
Christensen: Advanced Linear Modeling: Multivariate, Time Series, and Spatial Data; Nonparametric Regression and Response Surface Maximization, 2nd ed.
Christensen: Log-Linear Models and Logistic Regression, 2nd ed.
Christensen: Plane Answers to Complex Questions: The Theory of Linear Models, 2nd ed.
Cryer/Chan: Time Series Analysis, Second Edition
Davis: Statistical Methods for the Analysis of Repeated Measurements
Dean/Voss: Design and Analysis of Experiments
Dekking/Kraaijkamp/Lopuhaä/Meester: A Modern Introduction to Probability and Statistics
Durrett: Essential of Stochastic Processes
Edwards: Introduction to Graphical Modeling, 2nd ed.
Everitt: An R and S-PLUS Companion to Multivariate Analysis
Gentle: Matrix Algebra: Theory, Computations, and Applications in Statistics
Ghosh/Delampady/Samanta: An Introduction to Bayesian Analysis
Gut: Probability: A Graduate Course
Heiberger/Holland: Statistical Analysis and Data Display; An Intermediate Course with Examples in S-PLUS, R, and SAS
Jobson: Applied Multivariate Data Analysis, Volume I: Regression and Experimental Design
Jobson: Applied Multivariate Data Analysis, Volume II: Categorical and Multivariate Methods
Karr: Probability
Kulkarni: Modeling, Analysis, Design, and Control of Stochastic Systems
Lange: Applied Probability
Lange: Optimization
Lehmann: Elements of Large Sample Theory
Lehmann/Romano: Testing Statistical Hypotheses, 3rd ed.
Lehmann/Casella: Theory of Point Estimation, 2nd ed.
Longford: Studying Human Populations: An Advanced Course in Statistics
Marin/Robert: Bayesian Core: A Practical Approach to Computational Bayesian Statistics
Nolan/Speed: Stat Labs: Mathematical Statistics Through Applications
Pitman: Probability
Rawlings/Pantula/Dickey: Applied Regression Analysis
Robert: The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation, 2nd ed.
Robert/Casella: Monte Carlo Statistical Methods, 2nd ed.
Rose/Smith: Mathematical Statistics with *Mathematica*
Ruppert: Statistics and Finance: An Introduction
Sen/Srivastava: Regression Analysis: Theory, Methods, and Applications.
Shao: Mathematical Statistics, 2nd ed.
Shorack: Probability for Statisticians
Shumway/Stoffer: Time Series Analysis and Its Applications, 2nd ed.
Simonoff: Analyzing Categorical Data
Terrell: Mathematical Statistics: A Unified Introduction
Timm: Applied Multivariate Analysis
Toutenberg: Statistical Analysis of Designed Experiments, 2nd ed.
Wasserman: All of Nonparametric Statistics
Wasserman: All of Statistics: A Concise Course in Statistical Inference
Weiss: Modeling Longitudinal Data
Whittle: Probability via Expectation, 4th ed.

Jonathan D. Cryer • Kung-Sik Chan

Time Series Analysis

With Applications in R

Second Edition



Jonathan D. Cryer
Department of Statistics & Actuarial Science
University of Iowa
Iowa City, Iowa 52242
USA
jon-cryer@uiowa.edu

Kung-Sik Chan
Department of Statistics & Actuarial Science
University of Iowa
Iowa City, Iowa 52242
USA
kung-sik-chan@uiowa.edu

Series Editors:

George Casella
Department of Statistics
University of Florida
Gainesville, FL 32611-8545
USA

Stephen Fienberg
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213-3890
USA

Ingram Okin
Department of Statistics
Stanford University
Stanford, CA 94305
USA

ISBN: 978-0-387-75958-6

e-ISBN: 978-0-387-75959-3

Library of Congress Control Number: 2008923058

© 2008 Springer Science+Business Media, LLC

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

To our families

PREFACE

The theory and practice of time series analysis have developed rapidly since the appearance in 1970 of the seminal work of George E. P. Box and Gwilym M. Jenkins, *Time Series Analysis: Forecasting and Control*, now available in its third edition (1994) with co-author Gregory C. Reinsel. Many books on time series have appeared since then, but some of them give too little practical application, while others give too little theoretical background. This book attempts to present both application, and theory at a level accessible to a wide variety of students and practitioners. Our approach is to mix application and theory throughout the book as they are naturally needed.

The book was developed for a one-semester course usually attended by students in statistics, economics, business, engineering, and quantitative social sciences. Basic applied statistics through multiple linear regression is assumed. Calculus is assumed only to the extent of minimizing sums of squares, but a calculus-based introduction to statistics is necessary for a thorough understanding of some of the theory. However, required facts concerning expectation, variance, covariance, and correlation are reviewed in appendices. Also, conditional expectation properties and minimum mean square error prediction are developed in appendices. Actual time series data drawn from various disciplines are used throughout the book to illustrate the methodology. The book contains additional topics of a more advanced nature that can be selected for inclusion in a course if the instructor so chooses.

All of the plots and numerical output displayed in the book have been produced with the R software, which is available from the R Project for Statistical Computing at www.r-project.org. Some of the numerical output has been edited for additional clarity or for simplicity. R is available as free software under the terms of the Free Software Foundation's GNU General Public License in source code form. It runs on a wide variety of UNIX platforms and similar systems, Windows, and MacOS.

R is a language and environment for statistical computing and graphics, provides a wide variety of statistical (e.g., time-series analysis, linear and nonlinear modeling, classical statistical tests) and graphical techniques, and is highly extensible. The extensive appendix An Introduction to R, provides an introduction to the R software specially designed to go with this book. One of the authors (KSC) has produced a large number of new or enhanced R functions specifically tailored to the methods described in this book. They are listed on page 468 and are available in the package named TSA on the R Project's Website at www.r-project.org. We have also constructed R command script files for each chapter. These are available for download at www.stat.uiowa.edu/~kchan/TSA.htm. We also show the required R code beneath nearly every table and graphical display in the book. The datasets required for the exercises are named in each exercise by an appropriate filename; for example, larain for the Los Angeles rainfall data. However, if you are using the TSA package, the datasets are part of the package and may be accessed through the R command `data(larain)`, for example.

All of the datasets are also available at the textbook website as ACSCII files with variable names in the first row. We believe that many of the plots and calculations

described in the book could also be obtained with other software, such as SAS[®], Splus[®], Statgraphics[®], SCA[®], EViews[®], RATS[®], Ox[®], and others.

This book is a second edition of the book *Time Series Analysis* by Jonathan Cryer, published in 1986 by PWS-Kent Publishing (Duxbury Press). This new edition contains nearly all of the well-received original in addition to considerable new material, numerous new datasets, and new exercises. Some of the new topics that are integrated with the original include unit root tests, extended autocorrelation functions, subset ARIMA models, and bootstrapping. Completely new chapters cover the topics of time series regression models, time series models of heteroscedasticity, spectral analysis, and threshold models. Although the level of difficulty in these new chapters is somewhat higher than in the more basic material, we believe that the discussion is presented in a way that will make the material accessible and quite useful to a broad audience of users. Chapter 15, Threshold Models, is placed last since it is the only chapter that deals with nonlinear time series models. It could be covered earlier, say after Chapter 12. Also, Chapters 13 and 14 on spectral analysis could be covered after Chapter 10.

We would like to thank John Kimmel, Executive Editor, Statistics, at Springer, for his continuing interest and guidance during the long preparation of the manuscript. Professor Howell Tong of the London School of Economics, Professor Henghsiu Tsai of Academica Sinica, Taipei, Professor Noelle Samia of Northwestern University, Professor W. K. Li and Professor Kai W. Ng, both of the University of Hong Kong, and Professor Nils Christian Stenseth of the University of Oslo kindly read parts of the manuscript, and Professor Jun Yan used a preliminary version of the text for a class at the University of Iowa. Their constructive comments are greatly appreciated. We would like to thank Samuel Hao who helped with the exercise solutions and read the appendix: An Introduction to R. We would also like to thank several anonymous reviewers who read the manuscript at various stages. Their reviews led to a much improved book. Finally, one of the authors (JDC) would like to thank Dan, Marian, and Gene for providing such a great place, *Casa de Artes*, Club Santiago, Mexico, for working on the first draft of much of this new edition.

Iowa City, Iowa
January 2008

Jonathan D. Cryer
Kung-Sik Chan

CONTENTS

CHAPTER 1 INTRODUCTION	1
1.1 Examples of Time Series	1
1.2 A Model-Building Strategy	8
1.3 Time Series Plots in History	8
1.4 An Overview of the Book	9
Exercises	10
CHAPTER 2 FUNDAMENTAL CONCEPTS	11
2.1 Time Series and Stochastic Processes	11
2.2 Means, Variances, and Covariances	11
2.3 Stationarity	16
2.4 Summary	19
Exercises	19
Appendix A: Expectation, Variance, Covariance, and Correlation .	24
CHAPTER 3 TRENDS	27
3.1 Deterministic Versus Stochastic Trends	27
3.2 Estimation of a Constant Mean	28
3.3 Regression Methods.	30
3.4 Reliability and Efficiency of Regression Estimates	36
3.5 Interpreting Regression Output	40
3.6 Residual Analysis	42
3.7 Summary	50
Exercises	50
CHAPTER 4 MODELS FOR STATIONARY TIME SERIES	55
4.1 General Linear Processes	55
4.2 Moving Average Processes	57
4.3 Autoregressive Processes	66
4.4 The Mixed Autoregressive Moving Average Model.	77
4.5 Invertibility.	79
4.6 Summary	80
Exercises	81
Appendix B: The Stationarity Region for an AR(2) Process	84
Appendix C: The Autocorrelation Function for ARMA(p,q).	85

CHAPTER 5 MODELS FOR NONSTATIONARY TIME SERIES	.87
5.1 Stationarity Through Differencing	88
5.2 ARIMA Models.	92
5.3 Constant Terms in ARIMA Models.	97
5.4 Other Transformations	98
5.5 Summary	102
Exercises.	103
Appendix D: The Backshift Operator.	106
CHAPTER 6 MODEL SPECIFICATION	109
6.1 Properties of the Sample Autocorrelation Function	109
6.2 The Partial and Extended Autocorrelation Functions	112
6.3 Specification of Some Simulated Time Series.	117
6.4 Nonstationarity.	125
6.5 Other Specification Methods	130
6.6 Specification of Some Actual Time Series.	133
6.7 Summary	141
Exercises.	141
CHAPTER 7 PARAMETER ESTIMATION	149
7.1 The Method of Moments	149
7.2 Least Squares Estimation	154
7.3 Maximum Likelihood and Unconditional Least Squares	158
7.4 Properties of the Estimates	160
7.5 Illustrations of Parameter Estimation	163
7.6 Bootstrapping ARIMA Models	167
7.7 Summary	170
Exercises.	170
CHAPTER 8 MODEL DIAGNOSTICS	175
8.1 Residual Analysis	175
8.2 Overfitting and Parameter Redundancy.	185
8.3 Summary	188
Exercises.	188

CHAPTER 9 FORECASTING	191
9.1 Minimum Mean Square Error Forecasting	191
9.2 Deterministic Trends	191
9.3 ARIMA Forecasting	193
9.4 Prediction Limits	203
9.5 Forecasting Illustrations	204
9.6 Updating ARIMA Forecasts	207
9.7 Forecast Weights and Exponentially Weighted Moving Averages	207
9.8 Forecasting Transformed Series	209
9.9 Summary of Forecasting with Certain ARIMA Models	211
9.10 Summary	213
Exercises	213
Appendix E: Conditional Expectation	218
Appendix F: Minimum Mean Square Error Prediction	218
Appendix G: The Truncated Linear Process	221
Appendix H: State Space Models	222
CHAPTER 10 SEASONAL MODELS	227
10.1 Seasonal ARIMA Models	228
10.2 Multiplicative Seasonal ARMA Models	230
10.3 Nonstationary Seasonal ARIMA Models	233
10.4 Model Specification, Fitting, and Checking	234
10.5 Forecasting Seasonal Models	241
10.6 Summary	246
Exercises	246
CHAPTER 11 TIME SERIES REGRESSION MODELS	249
11.1 Intervention Analysis	249
11.2 Outliers	257
11.3 Spurious Correlation	260
11.4 Prewhitening and Stochastic Regression	265
11.5 Summary	273
Exercises	274

CHAPTER 12 TIME SERIES MODELS OF HETEROSCEDASTICITY.....	277
12.1 Some Common Features of Financial Time Series.....	278
12.2 The ARCH(1) Model	285
12.3 GARCH Models	289
12.4 Maximum Likelihood Estimation	298
12.5 Model Diagnostics	301
12.6 Conditions for the Nonnegativity of the Conditional Variances	307
12.7 Some Extensions of the GARCH Model	310
12.8 Another Example: The Daily USD/HKD Exchange Rates ..	311
12.9 Summary	315
Exercises.....	316
Appendix I: Formulas for the Generalized Portmanteau Tests ..	318
CHAPTER 13 INTRODUCTION TO SPECTRAL ANALYSIS.....	319
13.1 Introduction	319
13.2 The Periodogram	322
13.3 The Spectral Representation and Spectral Distribution.....	327
13.4 The Spectral Density	330
13.5 Spectral Densities for ARMA Processes	332
13.6 Sampling Properties of the Sample Spectral Density	340
13.7 Summary	346
Exercises.....	346
Appendix J: Orthogonality of Cosine and Sine Sequences ..	349
CHAPTER 14 ESTIMATING THE SPECTRUM	351
14.1 Smoothing the Spectral Density	351
14.2 Bias and Variance	354
14.3 Bandwidth	355
14.4 Confidence Intervals for the Spectrum	356
14.5 Leakage and Tapering.....	358
14.6 Autoregressive Spectrum Estimation.....	363
14.7 Examples with Simulated Data	364
14.8 Examples with Actual Data	370
14.9 Other Methods of Spectral Estimation.....	376
14.10 Summary	378
Exercises.....	378
Appendix K: Tapering and the Dirichlet Kernel	381

CHAPTER 15 THRESHOLD MODELS	383
15.1 Graphically Exploring Nonlinearity	384
15.2 Tests for Nonlinearity	390
15.3 Polynomial Models Are Generally Explosive	393
15.4 First-Order Threshold Autoregressive Models	395
15.5 Threshold Models	399
15.6 Testing for Threshold Nonlinearity	400
15.7 Estimation of a TAR Model	402
15.8 Model Diagnostics	411
15.9 Prediction	415
15.10 Summary	420
Exercises	420
Appendix L: The Generalized Portmanteau Test for TAR	421
APPENDIX: AN INTRODUCTION TO R	423
Introduction	423
Chapter 1 R Commands	429
Chapter 2 R Commands	433
Chapter 3 R Commands	433
Chapter 4 R Commands	438
Chapter 5 R Commands	439
Chapter 6 R Commands	441
Chapter 7 R Commands	442
Chapter 8 R Commands	446
Chapter 9 R Commands	447
Chapter 10 R Commands	450
Chapter 11 R Commands	451
Chapter 12 R Commands	457
Chapter 13 R Commands	460
Chapter 14 R Commands	461
Chapter 15 R Commands	462
New or Enhanced Functions in the TSA Library	468
DATASET INFORMATION	471
BIBLIOGRAPHY	477
INDEX	487

CHAPTER 1

INTRODUCTION

Data obtained from observations collected sequentially over time are extremely common. In business, we observe weekly interest rates, daily closing stock prices, monthly price indices, yearly sales figures, and so forth. In meteorology, we observe daily high and low temperatures, annual precipitation and drought indices, and hourly wind speeds. In agriculture, we record annual figures for crop and livestock production, soil erosion, and export sales. In the biological sciences, we observe the electrical activity of the heart at millisecond intervals. In ecology, we record the abundance of an animal species. The list of areas in which time series are studied is virtually endless. The purpose of time series analysis is generally twofold: to understand or model the stochastic mechanism that gives rise to an observed series and to predict or forecast the future values of a series based on the history of that series and, possibly, other related series or factors.

This chapter will introduce a variety of examples of time series from diverse areas of application. A somewhat unique feature of time series and their models is that we usually cannot assume that the observations arise independently from a common population (or from populations with different means, for example). Studying models that incorporate dependence is the key concept in time series analysis.

1.1 Examples of Time Series

In this section, we introduce a number of examples that will be pursued in later chapters.

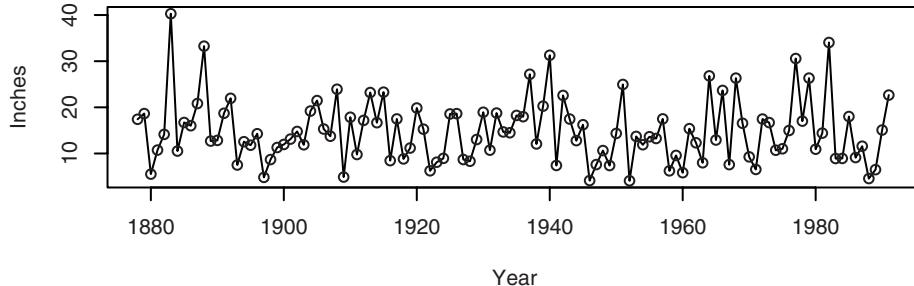
Annual Rainfall in Los Angeles

Exhibit 1.1 displays a time series plot of the annual rainfall amounts recorded in Los Angeles, California, over more than 100 years. The plot shows considerable variation in rainfall amount over the years—some years are low, some high, and many are in-between in value. The year 1883 was an exceptionally wet year for Los Angeles, while 1983 was quite dry. For analysis and modeling purposes we are interested in whether or not consecutive years are related in some way. If so, we might be able to use one year's rainfall value to help forecast next year's rainfall amount. One graphical way to investigate that question is to pair up consecutive rainfall values and plot the resulting scatterplot of pairs.

Exhibit 1.2 shows such a scatterplot for rainfall. For example, the point plotted near the lower right-hand corner shows that the year of extremely high rainfall, 40 inches in 1883, was followed by a middle of the road amount (about 12 inches) in 1884. The point

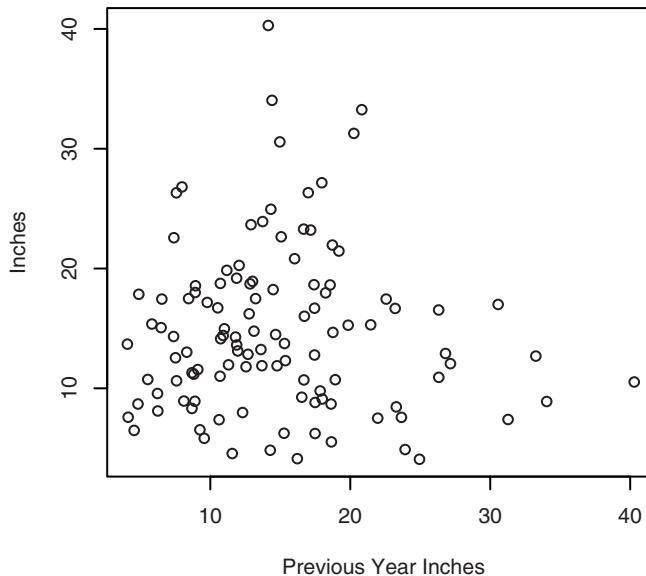
near the top of the display shows that the 40 inch year was preceded by a much more typical year of about 15 inches.

Exhibit 1.1 Time Series Plot of Los Angeles Annual Rainfall



```
> library(TSA)
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(larain); plot(larain, ylab='Inches', xlab='Year', type='o')
```

Exhibit 1.2 Scatterplot of LA Rainfall versus Last Year's LA Rainfall



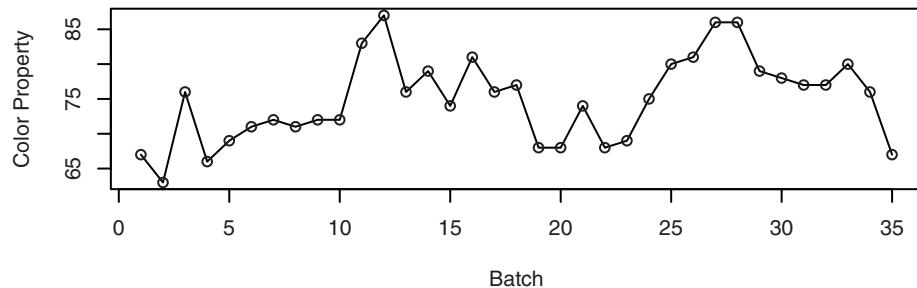
```
> win.graph(width=3, height=3, pointsize=8)
> plot(y=larain, x=zlag(larain), ylab='Inches',
       xlab='Previous Year Inches')
```

The main impression that we obtain from this plot is that there is little if any information about this year's rainfall amount from last year's amount. The plot shows no "trends" and no general tendencies. There is little correlation between last year's rainfall amount and this year's amount. From a modeling or forecasting point of view, this is not a very interesting time series!

An Industrial Chemical Process

As a second example, we consider a time series from an industrial chemical process. The variable measured here is a color property from consecutive batches in the process. Exhibit 1.3 shows a time series plot of these color values. Here values that are neighbors in time tend to be similar in size. It seems that neighbors are related to one another.

Exhibit 1.3 Time Series Plot of Color Property from a Chemical Process

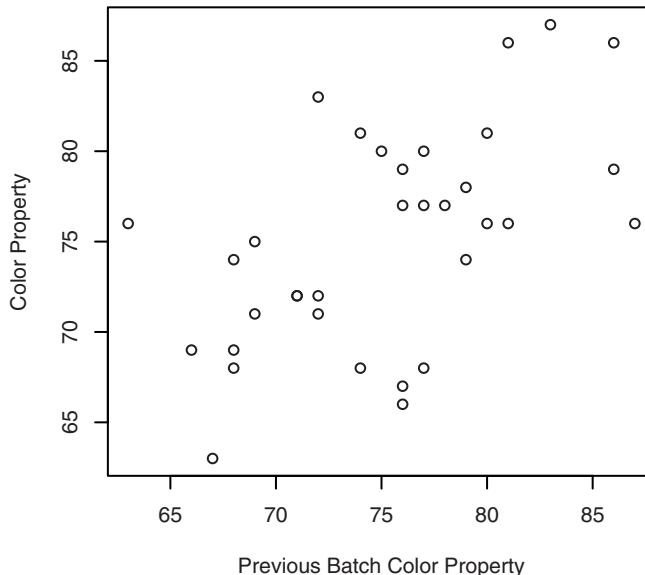


```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(color)
> plot(color, ylab='Color Property', xlab='Batch', type='o')
```

This can be seen better by constructing the scatterplot of neighboring pairs as we did with the first example.

Exhibit 1.4 displays the scatterplot of the neighboring pairs of color values. We see a slight upward trend in this plot—low values tend to be followed in the next batch by low values, middle-sized values tend to be followed by middle-sized values, and high values tend to be followed by high values. The trend is apparent but is not terribly strong. For example, the correlation in this scatterplot is about 0.6.

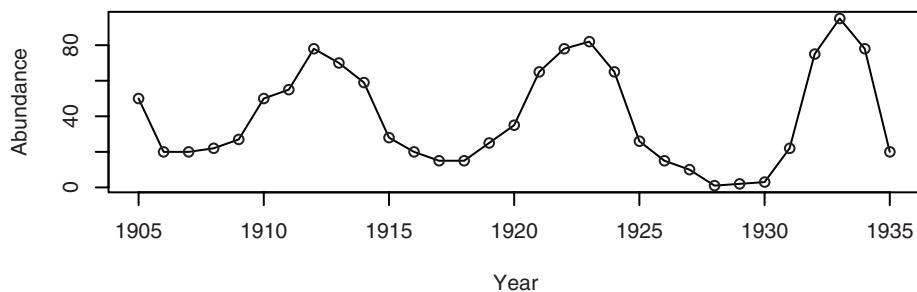
Exhibit 1.4 Scatterplot of Color Value versus Previous Color Value



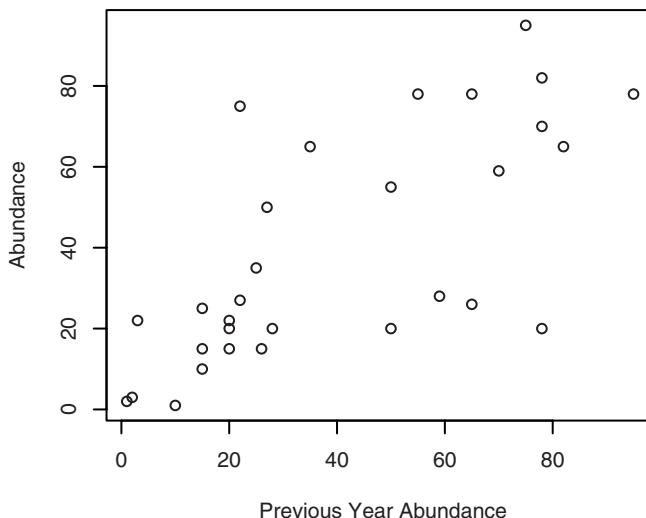
```
> win.graph(width=3,height=3,pointsizes=8)
> plot(y=color,x=zlag(color),ylab='Color Property',
       xlab='Previous Batch Color Property')
```

Annual Abundance of Canadian Hare

Our third example concerns the annual abundance of Canadian hare. Exhibit 1.5 gives the time series plot of this abundance over about 30 years. Neighboring values here are very closely related. Large changes in abundance do not occur from one year to the next. This neighboring correlation is seen clearly in Exhibit 1.6 where we have plotted abundance versus the previous year's abundance. As in the previous example, we see an upward trend in the plot—low values tend to be followed by low values in the next year, middle-sized values by middle-sized values, and high values by high values.

Exhibit 1.5 Abundance of Canadian Hare

```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(hare); plot(hare, ylab='Abundance', xlab='Year', type='o')
```

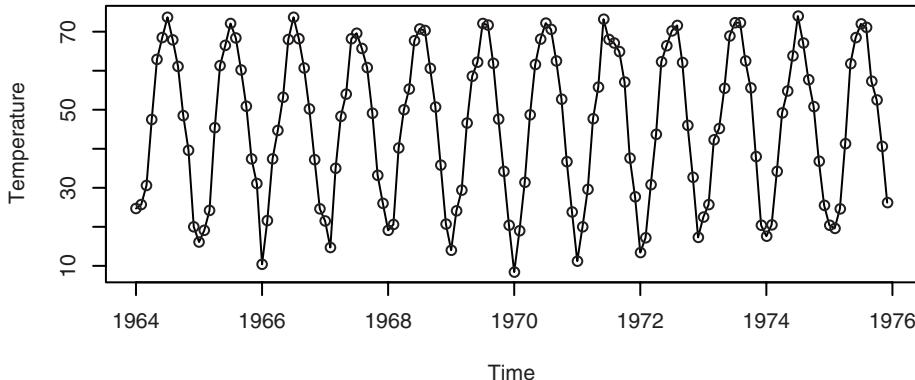
Exhibit 1.6 Hare Abundance versus Previous Year's Hare Abundance

```
> win.graph(width=3, height=3, pointsize=8)
> plot(y=hare, x=zlag(hare), ylab='Abundance',
       xlab='Previous Year Abundance')
```

Monthly Average Temperatures in Dubuque, Iowa

The average monthly temperatures (in degrees Fahrenheit) over a number of years recorded in Dubuque, Iowa, are shown in Exhibit 1.7.

Exhibit 1.7 Average Monthly Temperatures, Dubuque, Iowa



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(tempdub); plot(tempdub, ylab='Temperature', type='o')
```

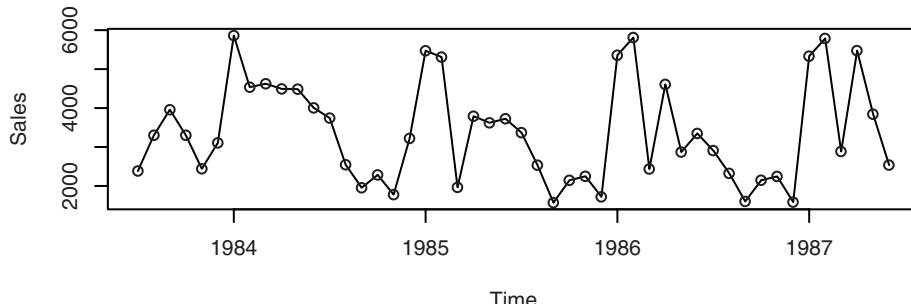
This time series displays a very regular pattern called **seasonality**. Seasonality for monthly values occurs when observations twelve months apart are related in some manner or another. All Januaries and Februarys are quite cold but they are similar in value and different from the temperatures of the warmer months of June, July, and August, for example. There is still variation among the January values and variation among the June values. Models for such series must accommodate this variation while preserving the similarities. Here the reason for the seasonality is well understood—the Northern Hemisphere's changing inclination toward the sun.

Monthly Oil Filter Sales

Our last example for this chapter concerns the monthly sales to dealers of a specialty oil filter for construction equipment manufactured by John Deere. When these data were first presented to one of the authors, the manager said, “There is no reason to believe that these sales are seasonal.” Seasonality would be present if January values tended to be related to other January values, February values tended to be related to other February values, and so forth. The time series plot shown in Exhibit 1.8 is not designed to display seasonality especially well. Exhibit 1.9 gives the same plot but amended to use meaningful plotting symbols. In this plot, all January values are plotted with the character J, all Februarys with F, all Marches with M, and so forth.[†] With these plotting symbols, it is much easier to see that sales for the winter months of January and February all tend to be high, while sales in September, October, November, and December are gener-

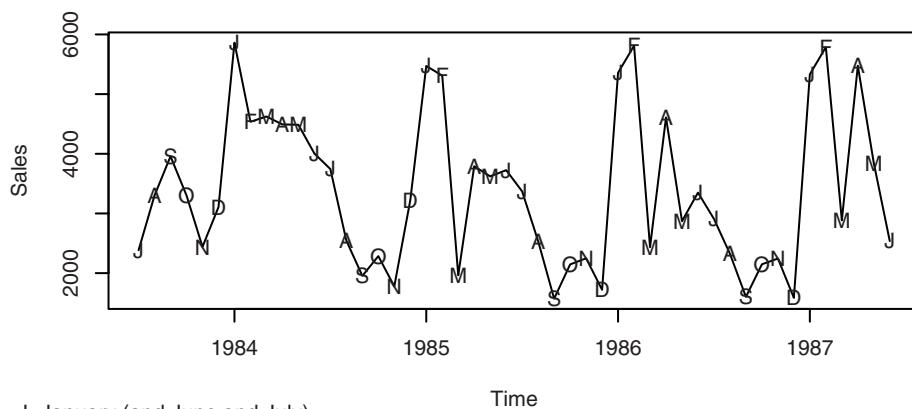
ally quite low. The seasonality in the data is much easier to see from this modified time series plot.

Exhibit 1.8 Monthly Oil Filter Sales



```
> data(oilfilters); plot(oilfilters,type='o',ylab='Sales')
```

Exhibit 1.9 Monthly Oil Filter Sales with Special Plotting Symbols



J=January (and June and July),

F=February, M=March (and May), and so forth

```
> plot(oilfilters,type='l',ylab='Sales')
> points(y=oilfilters,x=time(oilfilters),
  pch=as.vector(season(oilfilters)))
```

[†] In reading the plot, you will still have to distinguish between Januaries, Junes, and Julys, between Marches and Mays, and Aprils and Augusts, but this is easily done by looking at neighboring plotting characters.

In general, our goal is to emphasize plotting methods that are appropriate and useful for finding patterns that will lead to suitable models for our time series data. In later chapters, we will consider several different ways to incorporate seasonality into time series models.

1.2 A Model-Building Strategy

Finding appropriate models for time series is a nontrivial task. We will develop a multi-step model-building strategy espoused so well by Box and Jenkins (1976). There are three main steps in the process, each of which may be used several times:

1. **model specification** (or identification)
2. **model fitting**, and
3. **model diagnostics**

In model specification (or identification), the classes of time series models are selected that may be appropriate for a given observed series. In this step we look at the time plot of the series, compute many different statistics from the data, and also apply any knowledge of the subject matter in which the data arise, such as biology, business, or ecology. It should be emphasized that the model chosen at this point is *tentative* and subject to revision later on in the analysis.

In choosing a model, we shall attempt to adhere to the **principle of parsimony**; that is, the model used should require the smallest number of parameters that will adequately represent the time series. Albert Einstein is quoted in Parzen (1982, p. 68) as remarking that “everything should be made as simple as possible but not simpler.”

The model will inevitably involve one or more parameters whose values must be estimated from the observed series. Model fitting consists of finding the best possible estimates of those unknown parameters within a given model. We shall consider criteria such as least squares and maximum likelihood for estimation.

Model diagnostics is concerned with assessing the quality of the model that we have specified and estimated. How well does the model fit the data? Are the assumptions of the model reasonably well satisfied? If no inadequacies are found, the modeling may be assumed to be complete, and the model may be used, for example, to forecast future values. Otherwise, we choose another model in the light of the inadequacies found; that is, we return to the model specification step. In this way, we cycle through the three steps until, ideally, an acceptable model is found.

Because the computations required for each step in model building are intensive, we shall rely on readily available statistical software to carry out the calculations and do the plotting.

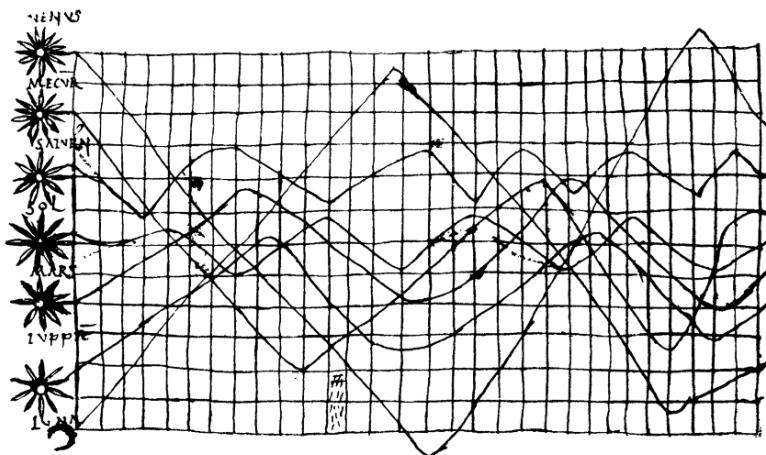
1.3 Time Series Plots in History

According to Tufte (1983, p. 28), “The time-series plot is the most frequently used form of graphic design. With one dimension marching along to the regular rhythm of sec-

onds, minutes, hours, days, weeks, months, years, or millennia, the natural ordering of the time scale gives this design a strength and efficiency of interpretation found in no other graphic arrangement.”

Exhibit 1.10 reproduces what appears to be the oldest known example of a time series plot, dating from the tenth (or possibly eleventh) century and showing the inclinations of the planetary orbits.[†] Commenting on this artifact, Tufte says “It appears as a mysterious and isolated wonder in the history of data graphics, since the next extant graphic of a plotted time-series shows up some 800 years later.”

Exhibit 1.10 A Tenth-Century Time Series Plot



1.4 An Overview of the Book

Chapter 2 develops the basic ideas of mean, covariance, and correlation functions and ends with the important concept of stationarity. Chapter 3 discusses trend analysis and investigates how to estimate and check common deterministic trend models, such as those for linear time trends and seasonal means.

Chapter 4 begins the development of parametric models for stationary time series, namely the so-called autoregressive moving average (ARMA) models (also known as Box-Jenkins models). These models are then generalized in Chapter 5 to encompass certain types of stochastic nonstationary cases—the ARIMA models.

Chapters 6, 7, and 8 form the heart of the model-building strategy for ARIMA modeling. Techniques are presented for tentatively specifying models (Chapter 6), efficiently estimating the model parameters using least squares and maximum likelihood (Chapter 7), and determining how well the models fit the data (Chapter 8).

Chapter 9 thoroughly develops the theory and methods of minimum mean square error forecasting for ARIMA models. Chapter 10 extends the ideas of Chapters 4

[†] From Tufte (1983, p. 28).

through 9 to stochastic seasonal models. The remaining chapters cover selected topics and are of a somewhat more advanced nature.

EXERCISES

- 1.1** Use software to produce the time series plot shown in Exhibit 1.2, on page 2. The data are in the file named `larain`.[†]
- 1.2** Produce the time series plot displayed in Exhibit 1.3, on page 3. The data file is named `color`.
- 1.3** Simulate a completely random process of length 48 with independent, normal values. Plot the time series plot. Does it look “random”? Repeat this exercise several times with a new simulation each time.
- 1.4** Simulate a completely random process of length 48 with independent, chi-square distributed values, each with 2 degrees of freedom. Display the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.
- 1.5** Simulate a completely random process of length 48 with independent, *t*-distributed values each with 5 degrees of freedom. Construct the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.
- 1.6** Construct a time series plot with monthly plotting symbols for the Dubuque temperature series as in Exhibit 1.9, on page 7. The data are in the file named `tempdub`.

[†]If you have installed the R package `TSA`, available for download at www.r-project.org, the `larain` data are accessed by the R command: `data(larain)`. An ASCII file of the data is also available on the book Website at www.stat.uiowa.edu/~kchan/TSA.htm.

CHAPTER 2

FUNDAMENTAL CONCEPTS

This chapter describes the fundamental concepts in the theory of time series models. In particular, we introduce the concepts of stochastic processes, mean and covariance functions, stationary processes, and autocorrelation functions.

2.1 Time Series and Stochastic Processes

The sequence of random variables $\{Y_t: t = 0, \pm 1, \pm 2, \pm 3, \dots\}$ is called a **stochastic process** and serves as a model for an observed time series. It is known that the complete probabilistic structure of such a process is determined by the set of distributions of all finite collections of the Y 's. Fortunately, we will not have to deal explicitly with these multivariate distributions. Much of the information in these joint distributions can be described in terms of means, variances, and covariances. Consequently, we concentrate our efforts on these first and second moments. (If the joint distributions of the Y 's are multivariate *normal* distributions, then the first and second moments completely determine all the joint distributions.)

2.2 Means, Variances, and Covariances

For a stochastic process $\{Y_t: t = 0, \pm 1, \pm 2, \pm 3, \dots\}$, the **mean function** is defined by

$$\mu_t = E(Y_t) \quad \text{for } t = 0, \pm 1, \pm 2, \dots \quad (2.2.1)$$

That is, μ_t is just the expected value of the process at time t . In general, μ_t can be different at each time point t .

The **autocovariance function**, $\gamma_{t,s}$, is defined as

$$\gamma_{t,s} = \text{Cov}(Y_t, Y_s) \quad \text{for } t, s = 0, \pm 1, \pm 2, \dots \quad (2.2.2)$$

where $\text{Cov}(Y_t, Y_s) = E[(Y_t - \mu_t)(Y_s - \mu_s)] = E(Y_t Y_s) - \mu_t \mu_s$.

The **autocorrelation function**, $\rho_{t,s}$, is given by

$$\rho_{t,s} = \text{Corr}(Y_t, Y_s) \quad \text{for } t, s = 0, \pm 1, \pm 2, \dots \quad (2.2.3)$$

where

$$\text{Corr}(Y_t, Y_s) = \frac{\text{Cov}(Y_t, Y_s)}{\sqrt{\text{Var}(Y_t)\text{Var}(Y_s)}} = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}} \quad (2.2.4)$$

We review the basic properties of expectation, variance, covariance, and correlation in Appendix A on page 24.

Recall that both covariance and correlation are measures of the (linear) dependence between random variables but that the unitless correlation is somewhat easier to interpret. The following important properties follow from known results and our definitions:

$$\left. \begin{array}{l} \gamma_{t,t} = \text{Var}(Y_t) \quad \rho_{t,t} = 1 \\ \gamma_{t,s} = \gamma_{s,t} \quad \rho_{t,s} = \rho_{s,t} \\ |\gamma_{t,s}| \leq \sqrt{\gamma_{t,t}\gamma_{s,s}} \quad |\rho_{t,s}| \leq 1 \end{array} \right\} \quad (2.2.5)$$

Values of $\rho_{t,s}$ near ± 1 indicate strong (linear) dependence, whereas values near zero indicate weak (linear) dependence. If $\rho_{t,s} = 0$, we say that Y_t and Y_s are *uncorrelated*.

To investigate the covariance properties of various time series models, the following result will be used repeatedly: If c_1, c_2, \dots, c_m and d_1, d_2, \dots, d_n are constants and t_1, t_2, \dots, t_m and s_1, s_2, \dots, s_n are time points, then

$$\text{Cov} \left[\sum_{i=1}^m c_i Y_{t_i}, \sum_{j=1}^n d_j Y_{s_j} \right] = \sum_{i=1}^m \sum_{j=1}^n c_i d_j \text{Cov}(Y_{t_i}, Y_{s_j}) \quad (2.2.6)$$

The proof of Equation (2.2.6), though tedious, is a straightforward application of the linear properties of expectation. As a special case, we obtain the well-known result

$$\text{Var} \left[\sum_{i=1}^n c_i Y_{t_i} \right] = \sum_{i=1}^n c_i^2 \text{Var}(Y_{t_i}) + 2 \sum_{i=2}^n \sum_{j=1}^{i-1} c_i c_j \text{Cov}(Y_{t_i}, Y_{t_j}) \quad (2.2.7)$$

The Random Walk

Let e_1, e_2, \dots be a sequence of independent, identically distributed random variables each with zero mean and variance σ_e^2 . The observed time series, $\{Y_t: t = 1, 2, \dots\}$, is constructed as follows:

$$\left. \begin{array}{l} Y_1 = e_1 \\ Y_2 = e_1 + e_2 \\ \vdots \\ Y_t = e_1 + e_2 + \dots + e_t \end{array} \right\} \quad (2.2.8)$$

Alternatively, we can write

$$Y_t = Y_{t-1} + e_t \quad (2.2.9)$$

with “initial condition” $Y_1 = e_1$. If the e ’s are interpreted as the sizes of the “steps” taken (forward or backward) along a number line, then Y_t is the position of the “random walker” at time t . From Equation (2.2.8), we obtain the mean function

$$\begin{aligned}\mu_t &= E(Y_t) = E(e_1 + e_2 + \cdots + e_t) = E(e_1) + E(e_2) + \cdots + E(e_t) \\ &= 0 + 0 + \cdots + 0\end{aligned}$$

so that

$$\mu_t = 0 \quad \text{for all } t \quad (2.2.10)$$

We also have

$$\begin{aligned}Var(Y_t) &= Var(e_1 + e_2 + \cdots + e_t) = Var(e_1) + Var(e_2) + \cdots + Var(e_t) \\ &= \sigma_e^2 + \sigma_e^2 + \cdots + \sigma_e^2\end{aligned}$$

so that

$$Var(Y_t) = t\sigma_e^2 \quad (2.2.11)$$

Notice that the process variance increases linearly with time.

To investigate the covariance function, suppose that $1 \leq t \leq s$. Then we have

$$\gamma_{t,s} = Cov(Y_t, Y_s) = Cov(e_1 + e_2 + \cdots + e_t, e_1 + e_2 + \cdots + e_t + e_{t+1} + \cdots + e_s)$$

From Equation (2.2.6), we have

$$\gamma_{t,s} = \sum_{i=1}^s \sum_{j=1}^t Cov(e_i, e_j)$$

However, these covariances are zero unless $i = j$, in which case they equal $Var(e_i) = \sigma_e^2$. There are exactly t of these so that $\gamma_{t,s} = t\sigma_e^2$.

Since $\gamma_{t,s} = \gamma_{s,t}$, this specifies the autocovariance function for all time points t and s and we can write

$$\gamma_{t,s} = t\sigma_e^2 \quad \text{for } 1 \leq t \leq s \quad (2.2.12)$$

The autocorrelation function for the random walk is now easily obtained as

$$\rho_{t,s} = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}} = \frac{\sqrt{t}}{\sqrt{s}} \quad \text{for } 1 \leq t \leq s \quad (2.2.13)$$

The following numerical values help us understand the behavior of the random walk.

$$\begin{aligned}\rho_{1,2} &= \sqrt{\frac{1}{2}} = 0.707 & \rho_{8,9} &= \sqrt{\frac{8}{9}} = 0.943 \\ \rho_{24,25} &= \sqrt{\frac{24}{25}} = 0.980 & \rho_{1,25} &= \sqrt{\frac{1}{25}} = 0.200\end{aligned}$$

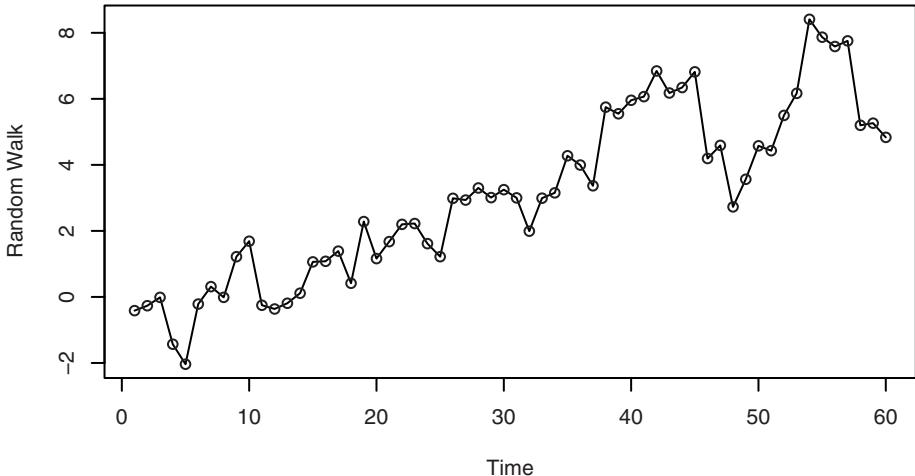
The values of Y at neighboring time points are more and more strongly and positively correlated as time goes by. On the other hand, the values of Y at distant time points are less and less correlated.

A simulated random walk is shown in Exhibit 2.1 where the e 's were selected from a standard normal distribution. Note that even though the theoretical mean function is

zero for all time points, the fact that the variance increases over time and that the correlation between process values nearby in time is nearly 1 indicate that we should expect long excursions of the process away from the mean level of zero.

The simple random walk process provides a good model (at least to a first approximation) for phenomena as diverse as the movement of common stock price, and the position of small particles suspended in a fluid—so-called Brownian motion.

Exhibit 2.1 Time Series Plot of a Random Walk



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(rwalk) # rwalk contains a simulated random walk
> plot(rwalk, type='o', ylab='Random Walk')
```

A Moving Average

As a second example, suppose that $\{Y_t\}$ is constructed as

$$Y_t = \frac{e_t + e_{t-1}}{2} \quad (2.2.14)$$

where (as always throughout this book) the e 's are assumed to be independent and identically distributed with zero mean and variance σ_e^2 . Here

$$\begin{aligned} \mu_t &= E(Y_t) = E\left\{\frac{e_t + e_{t-1}}{2}\right\} = \frac{E(e_t) + E(e_{t-1})}{2} \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} \text{Var}(Y_t) &= \text{Var}\left\{\frac{e_t + e_{t-1}}{2}\right\} = \frac{\text{Var}(e_t) + \text{Var}(e_{t-1})}{4} \\ &= 0.5\sigma_e^2 \end{aligned}$$

Also

$$\begin{aligned} \text{Cov}(Y_t, Y_{t-1}) &= \text{Cov}\left\{\frac{e_t + e_{t-1}}{2}, \frac{e_{t-1} + e_{t-2}}{2}\right\} \\ &= \frac{\text{Cov}(e_t, e_{t-1}) + \text{Cov}(e_t, e_{t-2}) + \text{Cov}(e_{t-1}, e_{t-1})}{4} \\ &\quad + \frac{\text{Cov}(e_{t-1}, e_{t-2})}{4} \\ &= \frac{\text{Cov}(e_{t-1}, e_{t-1})}{4} \quad (\text{as all the other covariances are zero}) \\ &= 0.25\sigma_e^2 \end{aligned}$$

or

$$\gamma_{t,t-1} = 0.25\sigma_e^2 \quad \text{for all } t \quad (2.2.15)$$

Furthermore,

$$\begin{aligned} \text{Cov}(Y_t, Y_{t-2}) &= \text{Cov}\left\{\frac{e_t + e_{t-1}}{2}, \frac{e_{t-2} + e_{t-3}}{2}\right\} \\ &= 0 \quad \text{since the } e's \text{ are independent.} \end{aligned}$$

Similarly, $\text{Cov}(Y_t, Y_{t-k}) = 0$ for $k > 1$, so we may write

$$\gamma_{t,s} = \begin{cases} 0.5\sigma_e^2 & \text{for } |t-s| = 0 \\ 0.25\sigma_e^2 & \text{for } |t-s| = 1 \\ 0 & \text{for } |t-s| > 1 \end{cases}$$

For the autocorrelation function, we have

$$\rho_{t,s} = \begin{cases} 1 & \text{for } |t-s| = 0 \\ 0.5 & \text{for } |t-s| = 1 \\ 0 & \text{for } |t-s| > 1 \end{cases} \quad (2.2.16)$$

since $0.25\sigma_e^2/0.5\sigma_e^2 = 0.5$.

Notice that $\rho_{2,1} = \rho_{3,2} = \rho_{4,3} = \rho_{9,8} = 0.5$. Values of Y precisely one time unit apart have exactly the same correlation no matter where they occur in time. Furthermore, $\rho_{3,1} = \rho_{4,2} = \rho_{t,t-2}$ and, more generally, $\rho_{t,t-k}$ is the same for all values of t . This leads us to the important concept of stationarity.

2.3 Stationarity

To make statistical inferences about the structure of a stochastic process on the basis of an observed record of that process, we must usually make some simplifying (and presumably reasonable) assumptions about that structure. The most important such assumption is that of **stationarity**. The basic idea of stationarity is that the probability laws that govern the behavior of the process do not change over time. In a sense, the process is in statistical equilibrium. Specifically, a process $\{Y_t\}$ is said to be **strictly stationary** if the joint distribution of $Y_{t_1}, Y_{t_2}, \dots, Y_{t_n}$ is the same as the joint distribution of $Y_{t_1-k}, Y_{t_2-k}, \dots, Y_{t_n-k}$ for all choices of time points t_1, t_2, \dots, t_n and all choices of time lag k .

Thus, when $n = 1$ the (univariate) distribution of Y_t is the same as that of Y_{t-k} for all t and k ; in other words, the Y 's are (marginally) identically distributed. It then follows that $E(Y_t) = E(Y_{t-k})$ for all t and k so that the mean function is constant for all time. Additionally, $\text{Var}(Y_t) = \text{Var}(Y_{t-k})$ for all t and k so that the variance is also constant over time.

Setting $n = 2$ in the stationarity definition we see that the bivariate distribution of Y_t and Y_s must be the same as that of Y_{t-k} and Y_{s-k} from which it follows that $\text{Cov}(Y_t, Y_s) = \text{Cov}(Y_{t-k}, Y_{s-k})$ for all t, s , and k . Putting $k = s$ and then $k = t$, we obtain

$$\begin{aligned}\gamma_{t,s} &= \text{Cov}(Y_{t-s}, Y_0) \\ &= \text{Cov}(Y_0, Y_{s-t}) \\ &= \text{Cov}(Y_0, Y_{|t-s|}) \\ &= \gamma_{0,|t-s|}\end{aligned}$$

That is, the covariance between Y_t and Y_s depends on time only through the time difference $|t-s|$ and not otherwise on the actual times t and s . Thus, for a stationary process, we can simplify our notation and write

$$\gamma_k = \text{Cov}(Y_t, Y_{t-k}) \quad \text{and} \quad \rho_k = \text{Corr}(Y_t, Y_{t-k}) \quad (2.3.1)$$

Note also that

$$\rho_k = \frac{\gamma_k}{\gamma_0}$$

The general properties given in Equation (2.2.5) now become

$$\left. \begin{array}{ll} \gamma_0 = \text{Var}(Y_t) & \rho_0 = 1 \\ \gamma_k = \gamma_{-k} & \rho_k = \rho_{-k} \\ |\gamma_k| \leq \gamma_0 & |\rho_k| \leq 1 \end{array} \right\} \quad (2.3.2)$$

If a process is strictly stationary and has finite variance, then the covariance function must depend only on the time lag.

A definition that is similar to that of strict stationarity but is mathematically weaker

is the following: A stochastic process $\{Y_t\}$ is said to be **weakly** (or **second-order**) **stationary** if

1. The mean function is constant over time, and
2. $\gamma_{t,t-k} = \gamma_{0,k}$ for all time t and lag k

In this book the term stationary when used alone will always refer to this weaker form of stationarity. However, if the joint distributions for the process are all multivariate normal distributions, it can be shown that the two definitions coincide. For stationary processes, we usually only consider $k \geq 0$.

White Noise

A very important example of a stationary process is the so-called **white noise** process, which is defined as a sequence of independent, identically distributed random variables $\{e_t\}$. Its importance stems not from the fact that it is an interesting model itself but from the fact that many useful processes can be constructed from white noise. The fact that $\{e_t\}$ is strictly stationary is easy to see since

$$\begin{aligned} & Pr(e_{t_1} \leq x_1, e_{t_2} \leq x_2, \dots, e_{t_n} \leq x_n) \\ &= Pr(e_{t_1} \leq x_1)Pr(e_{t_2} \leq x_2) \cdots Pr(e_{t_n} \leq x_n) \quad (\text{by independence}) \\ &= Pr(e_{t_1-k} \leq x_1)Pr(e_{t_2-k} \leq x_2) \cdots Pr(e_{t_n-k} \leq x_n) \\ & \hspace{10em} (\text{identical distributions}) \\ &= Pr(e_{t_1-k} \leq x_1, e_{t_2-k} \leq x_2, \dots, e_{t_n-k} \leq x_n) \quad (\text{by independence}) \end{aligned}$$

as required. Also, $\mu_t = E(e_t)$ is constant and

$$\gamma_k = \begin{cases} Var(e_t) & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases}$$

Alternatively, we can write

$$\rho_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \quad (2.3.3)$$

The term white noise arises from the fact that a frequency analysis of the model shows that, in analogy with white light, all frequencies enter equally. We usually assume that the white noise process has mean zero and denote $Var(e_t)$ by σ_e^2 .

The moving average example, on page 14, where $Y_t = (e_t + e_{t-1})/2$, is another example of a stationary process constructed from white noise. In our new notation, we have for the moving average process that

$$\rho_k = \begin{cases} 1 & \text{for } k = 0 \\ 0.5 & \text{for } |k| = 1 \\ 0 & \text{for } |k| \geq 2 \end{cases}$$

Random Cosine Wave

As a somewhat different example,[†] consider the process defined as follows:

$$Y_t = \cos\left[2\pi\left(\frac{t}{12} + \Phi\right)\right] \quad \text{for } t = 0, \pm 1, \pm 2, \dots$$

where Φ is selected (once) from a uniform distribution on the interval from 0 to 1. A sample from such a process will appear highly deterministic since Y_t will repeat itself identically every 12 time units and look like a perfect (discrete time) cosine curve. However, its maximum will not occur at $t = 0$ but will be determined by the random phase Φ . The phase Φ can be interpreted as the fraction of a complete cycle completed by time $t = 0$. Still, the statistical properties of this process can be computed as follows:

$$\begin{aligned} E(Y_t) &= E\left\{\cos\left[2\pi\left(\frac{t}{12} + \Phi\right)\right]\right\} \\ &= \int_0^1 \cos\left[2\pi\left(\frac{t}{12} + \phi\right)\right] d\phi \\ &= \frac{1}{2\pi} \sin\left[2\pi\left(\frac{t}{12} + \phi\right)\right] \Big|_{\phi=0}^1 \\ &= \frac{1}{2\pi} \left[\sin\left(2\pi \frac{t}{12} + 2\pi\right) - \sin\left(2\pi \frac{t}{12}\right) \right] \end{aligned}$$

But this is zero since the sines must agree. So $\mu_t = 0$ for all t .

Also

$$\begin{aligned} \gamma_{t,s} &= E\left\{\cos\left[2\pi\left(\frac{t}{12} + \Phi\right)\right] \cos\left[2\pi\left(\frac{s}{12} + \Phi\right)\right]\right\} \\ &= \int_0^1 \cos\left[2\pi\left(\frac{t}{12} + \phi\right)\right] \cos\left[2\pi\left(\frac{s}{12} + \phi\right)\right] d\phi \\ &= \frac{1}{2} \int_0^1 \left\{ \cos\left[2\pi\left(\frac{t-s}{12}\right)\right] + \cos\left[2\pi\left(\frac{t+s}{12} + 2\phi\right)\right] \right\} d\phi \\ &= \frac{1}{2} \left\{ \cos\left[2\pi\left(\frac{t-s}{12}\right)\right] + \frac{1}{4\pi} \sin\left[2\pi\left(\frac{t+s}{12} + 2\phi\right)\right] \Big|_{\phi=0}^1 \right\} \\ &= \frac{1}{2} \cos\left[2\pi\left(\frac{|t-s|}{12}\right)\right] \end{aligned}$$

[†] This example contains optional material that is not needed in order to understand most of the remainder of this book. It will be used in Chapter 13, Introduction to Spectral Analysis.

So the process is stationary with autocorrelation function

$$\rho_k = \cos\left(2\pi\frac{k}{12}\right) \quad \text{for } k = 0, \pm 1, \pm 2, \dots \quad (2.3.4)$$

This example suggests that it will be difficult to assess whether or not stationarity is a reasonable assumption for a given time series on the basis of the time sequence plot of the observed data.

The random walk of page 12, where $Y_t = e_1 + e_2 + \cdots + e_t$, is also constructed from white noise but is *not* stationary. For example, the variance function, $\text{Var}(Y_t) = t\sigma_e^2$, is *not* constant; furthermore, the covariance function $\gamma_{t,s} = t\sigma_e^2$ for $0 \leq t \leq s$ does not depend only on time lag. However, suppose that instead of analyzing $\{Y_t\}$ directly, we consider the differences of successive Y -values, denoted ∇Y_t . Then $\nabla Y_t = Y_t - Y_{t-1} = e_t$, so the *differenced series*, $\{\nabla Y_t\}$, is stationary. This represents a simple example of a technique found to be extremely useful in many applications. Clearly, many real time series cannot be reasonably modeled by stationary processes since they are not in statistical equilibrium but are evolving over time. However, we can frequently transform non-stationary series into stationary series by simple techniques such as differencing. Such techniques will be vigorously pursued in the remaining chapters.

2.4 Summary

In this chapter we have introduced the basic concepts of stochastic processes that serve as models for time series. In particular, you should now be familiar with the important concepts of mean functions, autocovariance functions, and autocorrelation functions. We illustrated these concepts with the basic processes: the random walk, white noise, a simple moving average, and a random cosine wave. Finally, the fundamental concept of stationarity introduced here will be used throughout the book.

EXERCISES

- 2.1** Suppose $E(X) = 2$, $\text{Var}(X) = 9$, $E(Y) = 0$, $\text{Var}(Y) = 4$, and $\text{Corr}(X, Y) = 0.25$. Find:
 - (a) $\text{Var}(X + Y)$.
 - (b) $\text{Cov}(X, X + Y)$.
 - (c) $\text{Corr}(X + Y, X - Y)$.
- 2.2** If X and Y are dependent but $\text{Var}(X) = \text{Var}(Y)$, find $\text{Cov}(X + Y, X - Y)$.
- 2.3** Let X have a distribution with mean μ and variance σ^2 , and let $Y_t = X$ for all t .
 - (a) Show that $\{Y_t\}$ is strictly and weakly stationary.
 - (b) Find the autocovariance function for $\{Y_t\}$.
 - (c) Sketch a “typical” time plot of Y_t .

- 2.4** Let $\{e_t\}$ be a zero mean white noise process. Suppose that the observed process is $Y_t = e_t + \theta e_{t-1}$, where θ is either 3 or 1/3.
- Find the autocorrelation function for $\{Y_t\}$ both when $\theta = 3$ and when $\theta = 1/3$.
 - You should have discovered that the time series is stationary regardless of the value of θ and that the autocorrelation functions are the same for $\theta = 3$ and $\theta = 1/3$. For simplicity, suppose that the process mean is known to be zero and the variance of Y_t is known to be 1. You observe the series $\{Y_t\}$ for $t = 1, 2, \dots, n$ and suppose that you can produce good estimates of the autocorrelations ρ_k . Do you think that you could determine which value of θ is correct (3 or 1/3) based on the estimate of ρ_k ? Why or why not?
- 2.5** Suppose $Y_t = 5 + 2t + X_t$, where $\{X_t\}$ is a zero-mean stationary series with autocovariance function γ_k .
- Find the mean function for $\{Y_t\}$.
 - Find the autocovariance function for $\{Y_t\}$.
 - Is $\{Y_t\}$ stationary? Why or why not?
- 2.6** Let $\{X_t\}$ be a stationary time series, and define $Y_t = \begin{cases} X_t & \text{for } t \text{ odd} \\ X_t + 3 & \text{for } t \text{ even.} \end{cases}$
- Show that $Cov(Y_t, Y_{t-k})$ is free of t for all lags k .
 - Is $\{Y_t\}$ stationary?
- 2.7** Suppose that $\{Y_t\}$ is stationary with autocovariance function γ_k .
- Show that $W_t = \nabla Y_t = Y_t - Y_{t-1}$ is stationary by finding the mean and autocovariance function for $\{W_t\}$.
 - Show that $U_t = \nabla^2 Y_t = \nabla[Y_t - Y_{t-1}] = Y_t - 2Y_{t-1} + Y_{t-2}$ is stationary. (You need not find the mean and autocovariance function for $\{U_t\}$.)
- 2.8** Suppose that $\{Y_t\}$ is stationary with autocovariance function γ_k . Show that for any fixed positive integer n and any constants c_1, c_2, \dots, c_n , the process $\{W_t\}$ defined by $W_t = c_1 Y_t + c_2 Y_{t-1} + \dots + c_n Y_{t-n+1}$ is stationary. (Note that Exercise 2.7 is a special case of this result.)
- 2.9** Suppose $Y_t = \beta_0 + \beta_1 t + X_t$, where $\{X_t\}$ is a zero-mean stationary series with autocovariance function γ_k and β_0 and β_1 are constants.
- Show that $\{Y_t\}$ is not stationary but that $W_t = \nabla Y_t = Y_t - Y_{t-1}$ is stationary.
 - In general, show that if $Y_t = \mu_t + X_t$, where $\{X_t\}$ is a zero-mean stationary series and μ_t is a polynomial in t of degree d , then $\nabla^m Y_t = \nabla(\nabla^{m-1} Y_t)$ is stationary for $m \geq d$ and nonstationary for $0 \leq m < d$.
- 2.10** Let $\{X_t\}$ be a zero-mean, unit-variance stationary process with autocorrelation function ρ_k . Suppose that μ_t is a nonconstant function and that σ_t is a positive-valued nonconstant function. The observed series is formed as $Y_t = \mu_t + \sigma_t X_t$.
- Find the mean and covariance function for the $\{Y_t\}$ process.
 - Show that the autocorrelation function for the $\{Y_t\}$ process depends only on the time lag. Is the $\{Y_t\}$ process stationary?
 - Is it possible to have a time series with a constant mean and with $Corr(Y_t, Y_{t-k})$ free of t but with $\{Y_t\}$ not stationary?

- 2.11** Suppose $Cov(X_t, X_{t-k}) = \gamma_k$ is free of t but that $E(X_t) = 3t$.
- Is $\{X_t\}$ stationary?
 - Let $Y_t = 7 - 3t + X_t$. Is $\{Y_t\}$ stationary?
- 2.12** Suppose that $Y_t = e_t - e_{t-12}$. Show that $\{Y_t\}$ is stationary and that, for $k > 0$, its autocorrelation function is nonzero only for lag $k = 12$.
- 2.13** Let $Y_t = e_t - \theta(e_{t-1})^2$. For this exercise, assume that the white noise series is normally distributed.
- Find the autocorrelation function for $\{Y_t\}$.
 - Is $\{Y_t\}$ stationary?
- 2.14** Evaluate the mean and covariance function for each of the following processes. In each case, determine whether or not the process is stationary.
- $Y_t = \theta_0 + t e_t$.
 - $W_t = \nabla Y_t$, where Y_t is as given in part (a).
 - $Y_t = e_t e_{t-1}$. (You may assume that $\{e_t\}$ is normal white noise.)
- 2.15** Suppose that X is a random variable with zero mean. Define a time series by $Y_t = (-1)^t X$.
- Find the mean function for $\{Y_t\}$.
 - Find the covariance function for $\{Y_t\}$.
 - Is $\{Y_t\}$ stationary?
- 2.16** Suppose $Y_t = A + X_t$, where $\{X_t\}$ is stationary and A is random but independent of $\{X_t\}$. Find the mean and covariance function for $\{Y_t\}$ in terms of the mean and autocovariance function for $\{X_t\}$ and the mean and variance of A .
- 2.17** Let $\{Y_t\}$ be stationary with autocovariance function γ_k . Let $\bar{Y} = \frac{1}{n} \sum_{t=1}^n Y_t$. Show that

$$\begin{aligned} Var(\bar{Y}) &= \frac{\gamma_0}{n} + \frac{2}{n} \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \gamma_k \\ &= \frac{1}{n} \sum_{k=-n+1}^{n-1} \left(1 - \frac{|k|}{n}\right) \gamma_k \end{aligned}$$

- 2.18** Let $\{Y_t\}$ be stationary with autocovariance function γ_k . Define the sample variance as $S^2 = \frac{1}{n-1} \sum_{t=1}^n (Y_t - \bar{Y})^2$.
- First show that $\sum_{t=1}^n (Y_t - \mu)^2 = \sum_{t=1}^n (Y_t - \bar{Y})^2 + n(\bar{Y} - \mu)^2$.
 - Use part (a) to show that
 - $E(S^2) = \frac{n}{n-1} \gamma_0 - \frac{n}{n-1} Var(\bar{Y}) = \gamma_0 - \frac{2}{n-1} \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \gamma_k$.
- (Use the results of Exercise 2.17 for the last expression.)
- (d)** If $\{Y_t\}$ is a white noise process with variance γ_0 , show that $E(S^2) = \gamma_0$.

- 2.19** Let $Y_1 = \theta_0 + e_1$, and then for $t > 1$ define Y_t recursively by $Y_t = \theta_0 + Y_{t-1} + e_t$. Here θ_0 is a constant. The process $\{Y_t\}$ is called a **random walk with drift**.
- Show that Y_t may be rewritten as $Y_t = t\theta_0 + e_t + e_{t-1} + \cdots + e_1$.
 - Find the mean function for Y_t .
 - Find the autocovariance function for Y_t .
- 2.20** Consider the standard random walk model where $Y_t = Y_{t-1} + e_t$ with $Y_1 = e_1$.
- Use the representation of Y_t above to show that $\mu_t = \mu_{t-1}$ for $t > 1$ with initial condition $\mu_1 = E(e_1) = 0$. Hence show that $\mu_t = 0$ for all t .
 - Similarly, show that $Var(Y_t) = Var(Y_{t-1}) + \sigma_e^2$ for $t > 1$ with $Var(Y_1) = \sigma_e^2$ and hence $Var(Y_t) = t\sigma_e^2$.
 - For $0 \leq t \leq s$, use $Y_s = Y_t + e_{t+1} + e_{t+2} + \cdots + e_s$ to show that $Cov(Y_t, Y_s) = Var(Y_t)$ and, hence, that $Cov(Y_t, Y_s) = \min(t, s)\sigma_e^2$.
- 2.21** For a random walk with random starting value, let $Y_t = Y_0 + e_t + e_{t-1} + \cdots + e_1$ for $t > 0$, where Y_0 has a distribution with mean μ_0 and variance σ_0^2 . Suppose further that Y_0, e_1, \dots, e_t are independent.
- Show that $E(Y_t) = \mu_0$ for all t .
 - Show that $Var(Y_t) = t\sigma_e^2 + \sigma_0^2$.
 - Show that $Cov(Y_t, Y_s) = \min(t, s)\sigma_e^2 + \sigma_0^2$.
 - Show that $Corr(Y_t, Y_s) = \frac{t\sigma_e^2 + \sigma_0^2}{s\sigma_e^2 + \sigma_0^2}$ for $0 \leq t \leq s$.
- 2.22** Let $\{e_t\}$ be a zero-mean white noise process, and let c be a constant with $|c| < 1$. Define Y_t recursively by $Y_t = cY_{t-1} + e_t$ with $Y_1 = e_1$.
- Show that $E(Y_t) = 0$.
 - Show that $Var(Y_t) = \sigma_e^2(1 + c^2 + c^4 + \cdots + c^{2t-2})$. Is $\{Y_t\}$ stationary?
 - Show that

$$Corr(Y_t, Y_{t-1}) = c \sqrt{\frac{Var(Y_{t-1})}{Var(Y_t)}} \text{ and, in general,}$$

$$Corr(Y_t, Y_{t-k}) = c^k \sqrt{\frac{Var(Y_{t-k})}{Var(Y_t)}} \text{ for } k > 0$$

Hint: Argue that Y_{t-1} is independent of e_t . Then use

$$Cov(Y_t, Y_{t-1}) = Cov(cY_{t-1} + e_t, Y_{t-1})$$

- (d) For large t , argue that

$$Var(Y_t) \approx \frac{\sigma_e^2}{1 - c^2} \quad \text{and} \quad Corr(Y_t, Y_{t-k}) \approx c^k \quad \text{for } k > 0$$

so that $\{Y_t\}$ could be called **asymptotically stationary**.

- (e) Suppose now that we alter the initial condition and put $Y_1 = \frac{e_1}{\sqrt{1 - c^2}}$. Show that now $\{Y_t\}$ is stationary.

- 2.23** Two processes $\{Z_t\}$ and $\{Y_t\}$ are said to be **independent** if for any time points t_1, t_2, \dots, t_m and s_1, s_2, \dots, s_n the random variables $\{Z_{t_1}, Z_{t_2}, \dots, Z_{t_m}\}$ are independent of the random variables $\{Y_{s_1}, Y_{s_2}, \dots, Y_{s_n}\}$. Show that if $\{Z_t\}$ and $\{Y_t\}$ are independent stationary processes, then $W_t = Z_t + Y_t$ is stationary.
- 2.24** Let $\{X_t\}$ be a time series in which we are interested. However, because the measurement process itself is not perfect, we actually observe $Y_t = X_t + e_t$. We assume that $\{X_t\}$ and $\{e_t\}$ are independent processes. We call X_t the **signal** and e_t the **measurement noise or error process**.

If $\{X_t\}$ is stationary with autocorrelation function ρ_k , show that $\{Y_t\}$ is also stationary with

$$\text{Corr}(Y_t, Y_{t-k}) = \frac{\rho_k}{1 + \sigma_e^2/\sigma_X^2} \quad \text{for } k \geq 1$$

We call σ_X^2/σ_e^2 the **signal-to-noise ratio**, or SNR. Note that the larger the SNR, the closer the autocorrelation function of the observed process $\{Y_t\}$ is to the autocorrelation function of the desired signal $\{X_t\}$.

- 2.25** Suppose $Y_t = \beta_0 + \sum_{i=1}^k [A_i \cos(2\pi f_i t) + B_i \sin(2\pi f_i t)]$, where $\beta_0, f_1, f_2, \dots, f_k$ are constants and $A_1, A_2, \dots, A_k, B_1, B_2, \dots, B_k$ are independent random variables with zero means and variances $\text{Var}(A_i) = \text{Var}(B_i) = \sigma_i^2$. Show that $\{Y_t\}$ is stationary and find its covariance function.
- 2.26** Define the function $\Gamma_{t,s} = \frac{1}{2}E[(Y_t - Y_s)^2]$. In geostatistics, $\Gamma_{t,s}$ is called the *semivariogram*.
- (a) Show that for a stationary process $\Gamma_{t,s} = \gamma_0 - \gamma_{|t-s|}$.
 - (b) A process is said to be *intrinsically stationary* if $\Gamma_{t,s}$ depends only on the time difference $|t-s|$. Show that the random walk process is intrinsically stationary.
- 2.27** For a fixed, positive integer r and constant ϕ , consider the time series defined by $Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \dots + \phi^r e_{t-r}$.
- (a) Show that this process is stationary for any value of ϕ .
 - (b) Find the autocorrelation function.
- 2.28** (Random cosine wave extended) Suppose that

$$Y_t = R \cos(2\pi(f t + \Phi)) \quad \text{for } t = 0, \pm 1, \pm 2, \dots$$

where $0 < f < \frac{1}{2}$ is a fixed frequency and R and Φ are uncorrelated random variables and with Φ uniformly distributed on the interval $(0,1)$.

- (a) Show that $E(Y_t) = 0$ for all t .
- (b) Show that the process is stationary with $\gamma_k = \frac{1}{2}E(R^2)\cos(2\pi f k)$.

Hint: Use the calculations leading up to Equation (2.3.4), on page 19.

2.29 (Random cosine wave extended further) Suppose that

$$Y_t = \sum_{j=1}^m R_j \cos[2\pi(f_j t + \Phi_j)] \quad \text{for } t = 0, \pm 1, \pm 2, \dots$$

where $0 < f_1 < f_2 < \dots < f_m < \frac{1}{2}$ are m fixed frequencies, and $R_1, \Phi_1, R_2, \Phi_2, \dots, R_m, \Phi_m$ are uncorrelated random variables with each Φ_j uniformly distributed on the interval $(0, 1)$.

(a) Show that $E(Y_t) = 0$ for all t .

(b) Show that the process is stationary with $\gamma_k = \frac{1}{2} \sum_{j=1}^m E(R_j^2) \cos(2\pi f_j k)$.
Hint: Do Exercise 2.28 first.

2.30 (Mathematical statistics required) Suppose that

$$Y_t = R \cos[2\pi(ft + \Phi)] \quad \text{for } t = 0, \pm 1, \pm 2, \dots$$

where R and Φ are independent random variables and f is a fixed frequency. The phase Φ is assumed to be uniformly distributed on $(0, 1)$, and the amplitude R has a Rayleigh distribution with pdf $f(r) = re^{-r^2/2}$ for $r > 0$. Show that for each time point t , Y_t has a normal distribution. (Hint: Let $Y = R \cos[2\pi(ft + \Phi)]$ and $X = R \sin[2\pi(ft + \Phi)]$. Now find the joint distribution of X and Y . It can also be shown that all of the finite dimensional distributions are multivariate normal and hence the process is strictly stationary.)

Appendix A: Expectation, Variance, Covariance, and Correlation

In this appendix, we define expectation for continuous random variables. However, all of the properties described hold for all types of random variables, discrete, continuous, or otherwise. Let X have probability density function $f(x)$ and let the pair (X, Y) have joint probability density function $f(x, y)$.

The **expected value** of X is defined as $E(X) = \int_{-\infty}^{\infty} xf(x)dx$.
(If $\int_{-\infty}^{\infty} |x|f(x)dx < \infty$; otherwise $E(X)$ is undefined.) $E(X)$ is also called the **expectation** of X or the **mean** of X and is often denoted μ or μ_X .

Properties of Expectation

If $h(x)$ is a function such that $\int_{-\infty}^{\infty} |h(x)|f(x)dx < \infty$, it may be shown that

$$E[h(X)] = \int_{-\infty}^{\infty} h(x)f(x)dx$$

Similarly, if $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |h(x,y)|f(x, y)dxdy < \infty$, it may be shown that

$$E[h(X, Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y)f(x, y)dxdy \quad (2.A.1)$$

As a corollary to Equation (2.A.1), we easily obtain the important result

$$E(aX + bY + c) = aE(X) + bE(Y) + c \quad (2.A.2)$$

We also have

$$E(XY) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xyf(x, y)dxdy \quad (2.A.3)$$

The **variance** of a random variable X is defined as

$$Var(X) = E\{[X - E(X)]^2\} \quad (2.A.4)$$

(provided $E(X^2)$ exists). The variance of X is often denoted by σ^2 or σ_X^2 .

Properties of Variance

$$Var(X) \geq 0 \quad (2.A.5)$$

$$Var(a + bX) = b^2 Var(X) \quad (2.A.6)$$

If X and Y are independent, then

$$Var(X + Y) = Var(X) + Var(Y) \quad (2.A.7)$$

In general, it may be shown that

$$Var(X) = E(X^2) - [E(X)]^2 \quad (2.A.8)$$

The positive square root of the variance of X is called the **standard deviation** of X and is often denoted by σ or σ_X . The random variable $(X - \mu_X)/\sigma_X$ is called the **standardized version** of X . The mean and standard deviation of a standardized variable are always zero and one, respectively.

The **covariance** of X and Y is defined as $Cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)]$.

Properties of Covariance

$$Cov(a + bX, c + dY) = bdCov(X, Y) \quad (2.A.9)$$

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y) \quad (2.A.10)$$

$$Cov(X + Y, Z) = Cov(X, Z) + Cov(Y, Z) \quad (2.A.11)$$

$$Cov(X, X) = Var(X) \quad (2.A.12)$$

$$Cov(X, Y) = Cov(Y, X) \quad (2.A.13)$$

If X and Y are independent,

$$Cov(X, Y) = 0 \quad (2.A.14)$$

The **correlation coefficient** of X and Y , denoted by $\text{Corr}(X, Y)$ or ρ , is defined as

$$\rho = \text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

Alternatively, if X^* is a standardized X and Y^* is a standardized Y , then $\rho = E(X^*Y^*)$.

Properties of Correlation

$$-1 \leq \text{Corr}(X, Y) \leq 1 \quad (2.A.15)$$

$$\text{Corr}(a + bX, c + dY) = \text{sign}(bd)\text{Corr}(X, Y)$$

$$\text{where } \text{sign}(bd) = \begin{cases} 1 & \text{if } bd > 0 \\ 0 & \text{if } bd = 0 \\ -1 & \text{if } bd < 0 \end{cases} \quad (2.A.16)$$

$\text{Corr}(X, Y) = \pm 1$ if and only if there are constants a and b such that $\text{Pr}(Y = a + bX) = 1$.

CHAPTER 3

TRENDS

In a general time series, the mean function is a totally arbitrary function of time. In a stationary time series, the mean function must be constant in time. Frequently we need to take the middle ground and consider mean functions that are relatively simple (but not constant) functions of time. These trends are considered in this chapter.

3.1 Deterministic Versus Stochastic Trends

“Trends” can be quite elusive. The same time series may be viewed quite differently by different analysts. The simulated random walk shown in Exhibit 2.1 might be considered to display a general upward trend. However, we know that the random walk process has zero mean for all time. The perceived trend is just an artifact of the strong positive correlation between the series values at nearby time points and the increasing variance in the process as time goes by. A second and third simulation of exactly the same process might well show completely different “trends.” We ask you to produce some additional simulations in the exercises. Some authors have described such trends as **stochastic trends** (see Box, Jenkins, and Reinsel, 1994), although there is no generally accepted definition of a stochastic trend.

The average monthly temperature series plotted in Exhibit 1.7 on page 6, shows a cyclical or seasonal trend, but here the reason for the trend is clear—the Northern Hemisphere’s changing inclination toward the sun. In this case, a possible model might be $Y_t = \mu_t + X_t$, where μ_t is a deterministic function that is periodic with period 12; that is μ_t , should satisfy

$$\mu_t = \mu_{t-12} \quad \text{for all } t$$

We might assume that X_t , the unobserved variation around μ_t , has zero mean for all t so that indeed μ_t is the mean function for the observed series Y_t . We could describe this model as having a **deterministic trend** as opposed to the stochastic trend considered earlier. In other situations we might hypothesize a deterministic trend that is linear in time (that is, $\mu_t = \beta_0 + \beta_1 t$) or perhaps a quadratic time trend, $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$. Note that an implication of the model $Y_t = \mu_t + X_t$ with $E(X_t) = 0$ for all t is that the deterministic trend μ_t applies for all time. Thus, if $\mu_t = \beta_0 + \beta_1 t$, we are assuming that the *same* linear time trend applies forever. We should therefore have good reasons for assuming such a model—not just because the series looks somewhat linear over the time period observed.

In this chapter, we consider methods for modeling deterministic trends. Stochastic trends will be discussed in Chapter 5, and stochastic seasonal models will be discussed in Chapter 10. Many authors use the word trend only for a slowly changing mean function, such as a linear time trend, and use the term seasonal component for a mean function that varies cyclically. We do not find it useful to make such distinctions here.

3.2 Estimation of a Constant Mean

We first consider the simple situation where a constant mean function is assumed. Our model may then be written as

$$Y_t = \mu + X_t \quad (3.2.1)$$

where $E(X_t) = 0$ for all t . We wish to estimate μ with our observed time series Y_1, Y_2, \dots, Y_n . The most common estimate of μ is the sample mean or average defined as

$$\bar{Y} = \frac{1}{n} \sum_{t=1}^n Y_t \quad (3.2.2)$$

Under the minimal assumptions of Equation (3.2.1), we see that $E(\bar{Y}) = \mu$; therefore \bar{Y} is an unbiased estimate of μ . To investigate the precision of \bar{Y} as an estimate of μ , we need to make further assumptions concerning X_t .

Suppose that $\{Y_t\}$, (or, equivalently, $\{X_t\}$) of Equation (3.2.1)) is a stationary time series with autocorrelation function ρ_k . Then, by Exercise 2.17, we have

$$\begin{aligned} Var(\bar{Y}) &= \frac{\gamma_0}{n} \left[\sum_{k=-n+1}^{n-1} \left(1 - \frac{|k|}{n}\right) \rho_k \right] \\ &= \frac{\gamma_0}{n} \left[1 + 2 \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \rho_k \right] \end{aligned} \quad (3.2.3)$$

Notice that the first factor, γ_0/n , is the process (population) variance divided by the sample size—a concept with which we are familiar in simpler random sampling contexts. If the series $\{X_t\}$ of Equation (3.2.1) is just white noise, then $\rho_k = 0$ for $k > 0$ and $Var(\bar{Y})$ reduces to simply γ_0/n .

In the (stationary) moving average model $Y_t = e_t - \frac{1}{2}e_{t-1}$, we find that $\rho_1 = -0.4$ and $\rho_k = 0$ for $k > 1$. In this case, we have

$$\begin{aligned} Var(\bar{Y}) &= \frac{\gamma_0}{n} \left[1 + 2 \left(1 - \frac{1}{n}\right) (-0.4) \right] \\ &= \frac{\gamma_0}{n} \left[1 - 0.8 \left(\frac{n-1}{n}\right) \right] \end{aligned}$$

For values of n usually occurring in time series ($n > 50$, say), the factor $(n-1)/n$ will be close to 1, so that we have

$$Var(\bar{Y}) \approx 0.2 \frac{\gamma_0}{n}$$

We see that the negative correlation at lag 1 has improved the estimation of the mean compared with the estimation obtained in the white noise (random sample) situation. Because the series tends to oscillate back and forth across the mean, the sample mean obtained is more precise.

On the other hand, if $\rho_k \geq 0$ for all $k \geq 1$, we see from Equation (3.2.3) that $Var(\bar{Y})$ will be larger than γ_0/n . Here the positive correlations make estimation of the mean *more* difficult than in the white noise case. In general, some correlations will be positive and some negative, and Equation (3.2.3) must be used to assess the total effect.

For many stationary processes, the autocorrelation function decays quickly enough with increasing lags that

$$\sum_{k=0}^{\infty} |\rho_k| < \infty \quad (3.2.4)$$

(The random cosine wave of Chapter 2 is an exception.)

Under assumption (3.2.4) and given a large sample size n , the following useful approximation follows from Equation (3.2.3) (See Anderson, 1971, p. 459, for example)

$$Var(\bar{Y}) \approx \frac{\gamma_0}{n} \left[\sum_{k=-\infty}^{\infty} \rho_k \right] \quad \text{for large } n \quad (3.2.5)$$

Notice that to this approximation the variance is inversely proportional to the sample size n .

As an example, suppose that $\rho_k = \phi^{|k|}$ for all k , where ϕ is a number strictly between -1 and $+1$. Summing a geometric series yields

$$Var(\bar{Y}) \approx \frac{(1+\phi)\gamma_0}{(1-\phi)n} \quad (3.2.6)$$

For a nonstationary process (but with a constant mean), the precision of the sample mean as an estimate of μ can be strikingly different. As a useful example, suppose that in Equation (3.2.1) $\{X_t\}$ is a random walk process as described in Chapter 2. Then directly from Equation (2.2.8) we have

$$\begin{aligned} Var(\bar{Y}) &= \frac{1}{n^2} Var \left[\sum_{i=1}^n Y_i \right] \\ &= \frac{1}{n^2} Var \left[\sum_{i=1}^n \sum_{j=1}^i e_j \right] \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{n^2} \text{Var}(e_1 + 2e_2 + 3e_3 + \cdots + ne_n) \\
 &= \frac{\sigma_e^2}{n^2} \sum_{k=1}^n k^2
 \end{aligned}$$

so that

$$\text{Var}(\bar{Y}) = \sigma_e^2(2n+1)\frac{(n+1)}{6n} \quad (3.2.7)$$

Notice that in this special case the variance of our estimate of the mean actually *increases* as the sample size n increases. Clearly this is unacceptable, and we need to consider other estimation techniques for nonstationary series.

3.3 Regression Methods

The classical statistical method of regression analysis may be readily used to estimate the parameters of common nonconstant mean trend models. We shall consider the most useful ones: linear, quadratic, seasonal means, and cosine trends.

Linear and Quadratic Trends in Time

Consider the deterministic time trend expressed as

$$\mu_t = \beta_0 + \beta_1 t \quad (3.3.1)$$

where the *slope* and *intercept*, β_1 and β_0 respectively, are unknown parameters. The classical least squares (or regression) method is to choose as estimates of β_1 and β_0 values that minimize

$$Q(\beta_0, \beta_1) = \sum_{t=1}^n [Y_t - (\beta_0 + \beta_1 t)]^2$$

The solution may be obtained in several ways, for example, by computing the partial derivatives with respect to both β 's, setting the results equal to zero, and solving the resulting linear equations for the β 's. Denoting the solutions by $\hat{\beta}_0$ and $\hat{\beta}_1$, we find that

$$\begin{aligned}
 \hat{\beta}_1 &= \frac{\sum_{t=1}^n (Y_t - \bar{Y})(t - \bar{t})}{\sum_{t=1}^n (t - \bar{t})^2} \\
 \hat{\beta}_0 &= \bar{Y} - \hat{\beta}_1 \bar{t}
 \end{aligned} \quad (3.3.2)$$

where $\bar{t} = (n+1)/2$ is the average of $1, 2, \dots, n$. These formulas can be simplified somewhat, and various versions of the formulas are well-known. However, we assume that

the computations will be done by statistical software and we will not pursue other expressions for $\hat{\beta}_0$ and $\hat{\beta}_1$ here.

Example

Consider the random walk process that was shown in Exhibit 2.1. Suppose we (mistakenly) treat this as a linear time trend and estimate the slope and intercept by least-squares regression. Using statistical software we obtain Exhibit 3.1.

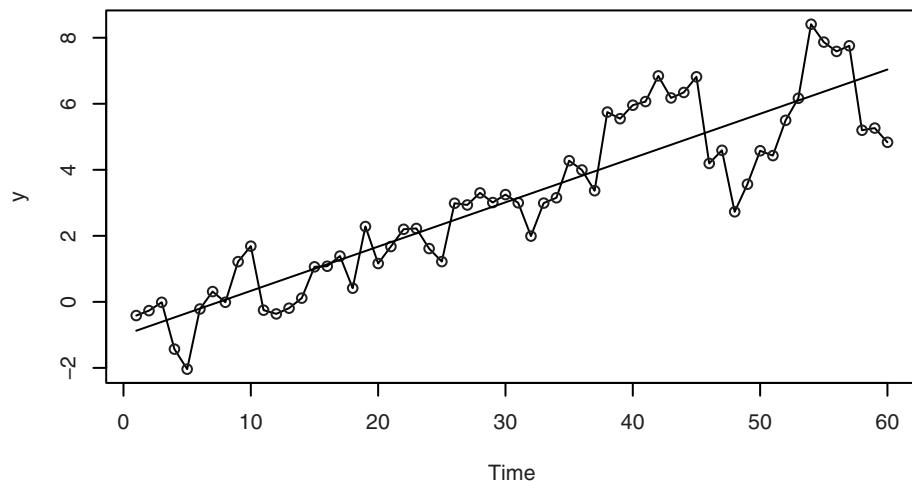
Exhibit 3.1 Least Squares Regression Estimates for Linear Time Trend

	Estimate	Std. Error	t value	Pr(> t)
Intercept	-1.008	0.2972	-3.39	0.00126
Time	0.1341	0.00848	15.82	< 0.0001

```
> data(rwalk)
> model1=lm(rwalk~time(rwalk))
> summary(model1)
```

So here the estimated slope and intercept are $\hat{\beta}_1 = 0.1341$ and $\hat{\beta}_0 = -1.008$, respectively. Exhibit 3.2 displays the random walk with the least squares regression trend line superimposed. We will interpret more of the regression output later in Section 3.5 on page 40 and see that fitting a line to these data is not appropriate.

Exhibit 3.2 Random Walk with Linear Time Trend



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> plot(rwalk, type='o', ylab='y')
> abline(model1) # add the fitted least squares line from model1
```

Cyclical or Seasonal Trends

Consider now modeling and estimating seasonal trends, such as for the average monthly temperature data in Exhibit 1.7. Here we assume that the observed series can be represented as

$$Y_t = \mu_t + X_t$$

where $E(X_t) = 0$ for all t .

The most general assumption for μ_t with monthly seasonal data is that there are 12 constants (parameters), β_1, β_2, \dots , and β_{12} , giving the expected average temperature for each of the 12 months. We may write

$$\mu_t = \begin{cases} \beta_1 & \text{for } t = 1, 13, 25, \dots \\ \beta_2 & \text{for } t = 2, 14, 26, \dots \\ \vdots & \\ \beta_{12} & \text{for } t = 12, 24, 36, \dots \end{cases} \quad (3.3.3)$$

This is sometimes called a **seasonal means** model.

As an example of this model consider the average monthly temperature data shown in Exhibit 1.7 on page 6. To fit such a model, we need to set up indicator variables (sometimes called dummy variables) that indicate the month to which each of the data points pertains. The procedure for doing this will depend on the particular statistical software that you use. We also need to note that the model as stated does not contain an intercept term, and the software will need to know this also. Alternatively, we could use an intercept and leave out any one of the β 's in Equation (3.3.3).

Exhibit 3.3 displays the results of fitting the seasonal means model to the temperature data. Here the t -values and $Pr(>|t|)$ -values reported are of little interest since they relate to testing the null hypotheses that the β 's are zero—not an interesting hypothesis in this case.

Exhibit 3.3 Regression Results for the Seasonal Means Model

	Estimate	Std. Error	t-value	Pr(> t)
January	16.608	0.987	16.8	< 0.0001
February	20.650	0.987	20.9	< 0.0001
March	32.475	0.987	32.9	< 0.0001
April	46.525	0.987	47.1	< 0.0001
May	58.092	0.987	58.9	< 0.0001
June	67.500	0.987	68.4	< 0.0001
July	71.717	0.987	72.7	< 0.0001

	Estimate	Std. Error	t-value	$Pr(> t)$
August	69.333	0.987	70.2	< 0.0001
September	61.025	0.987	61.8	< 0.0001
October	50.975	0.987	51.6	< 0.0001
November	36.650	0.987	37.1	< 0.0001
December	23.642	0.987	24.0	< 0.0001

```

> data(tempdub)
> month.=season(tempdub) # period added to improve table display
> model2=lm(tempdub~month.-1) # -1 removes the intercept term
> summary(model2)

```

Exhibit 3.4 shows how the results change when we fit a model *with* an intercept term. The software omits the January coefficient in this case. Now the February coefficient is interpreted as the difference between February and January average temperatures, the March coefficient is the difference between March and January average temperatures, and so forth. Once more, the *t*-values and $Pr(>|t|)$ (*p*-values) are testing hypotheses of little interest in this case. Notice that the Intercept coefficient plus the February coefficient here equals the February coefficient displayed in Exhibit 3.3.

Exhibit 3.4 Results for Seasonal Means Model with an Intercept

	Estimate	Std. Error	t-value	$Pr(> t)$
Intercept	16.608	0.987	16.83	< 0.0001
February	4.042	1.396	2.90	0.00443
March	15.867	1.396	11.37	< 0.0001
April	29.917	1.396	21.43	< 0.0001
May	41.483	1.396	29.72	< 0.0001
June	50.892	1.396	36.46	< 0.0001
July	55.108	1.396	39.48	< 0.0001
August	52.725	1.396	37.78	< 0.0001
September	44.417	1.396	31.82	< 0.0001
October	34.367	1.396	24.62	< 0.0001
November	20.042	1.396	14.36	< 0.0001
December	7.033	1.396	5.04	< 0.0001

```

> model3=lm(tempdub~month.) # January is dropped automatically
> summary(model3)

```

Cosine Trends

The seasonal means model for monthly data consists of 12 independent parameters and does not take the shape of the seasonal trend into account at all. For example, the fact that the March and April means are quite similar (and different from the June and July means) is not reflected in the model. In some cases, seasonal trends can be modeled economically with cosine curves that incorporate the smooth change expected from one time period to the next while still preserving the seasonality.

Consider the cosine curve with equation

$$\mu_t = \beta \cos(2\pi ft + \Phi) \quad (3.3.4)$$

We call β (> 0) the *amplitude*, f the *frequency*, and Φ the *phase* of the curve. As t varies, the curve oscillates between a maximum of β and a minimum of $-\beta$. Since the curve repeats itself exactly every $1/f$ time units, $1/f$ is called the period of the cosine wave. As noted in Chapter 2, Φ serves to set the arbitrary origin on the time axis. For monthly data with time indexed as 1, 2, ..., the most important frequency is $f = 1/12$, because such a cosine wave will repeat itself every 12 months. We say that the *period* is 12.

Equation (3.3.4) is inconvenient for estimation because the parameters β and Φ do not enter the expression linearly. Fortunately, a trigonometric identity is available that reparameterizes (3.3.4) more conveniently, namely

$$\beta \cos(2\pi ft + \Phi) = \beta_1 \cos(2\pi ft) + \beta_2 \sin(2\pi ft) \quad (3.3.5)$$

where

$$\beta = \sqrt{\beta_1^2 + \beta_2^2}, \quad \Phi = \text{atan}(-\beta_2/\beta_1) \quad (3.3.6)$$

and, conversely,

$$\beta_1 = \beta \cos(\Phi), \quad \beta_2 = \beta \sin(\Phi) \quad (3.3.7)$$

To estimate the parameters β_1 and β_2 with regression techniques, we simply use $\cos(2\pi ft)$ and $\sin(2\pi ft)$ as regressors or predictor variables.

The simplest such model for the trend would be expressed as

$$\mu_t = \beta_0 + \beta_1 \cos(2\pi ft) + \beta_2 \sin(2\pi ft) \quad (3.3.8)$$

Here the constant term, β_0 , can be meaningfully thought of as a cosine with frequency zero.

In any practical example, we must be careful how we measure time, as our choice of time measurement will affect the values of the frequencies of interest. For example, if we have monthly data but use 1, 2, 3, ... as our time scale, then $1/12$ would be the most interesting frequency, with a corresponding period of 12 months. However, if we measure time by year and fractional year, say 1980 for January, 1980.08333 for February of 1980, and so forth, then a frequency of 1 corresponds to an annual or 12 month periodicity.

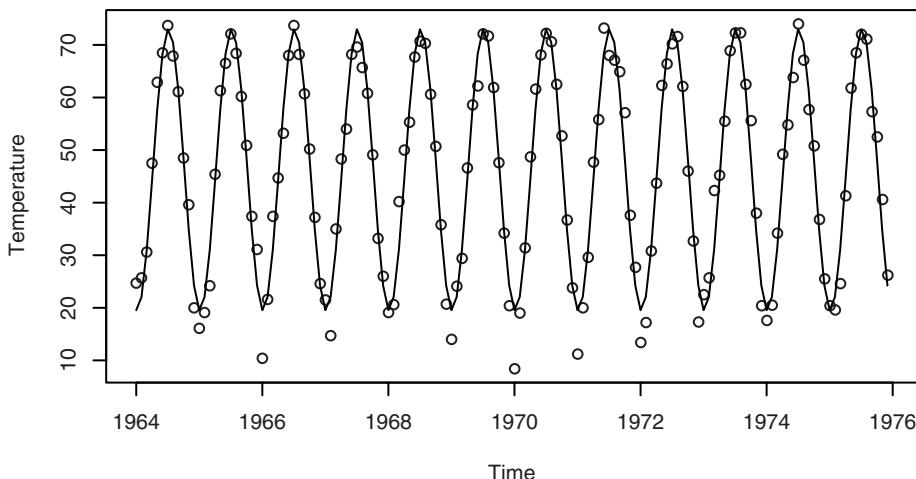
Exhibit 3.5 is an example of fitting a cosine curve at the fundamental frequency to the average monthly temperature series.

Exhibit 3.5 Cosine Trend Model for Temperature Series

Coefficient	Estimate	Std. Error	t-value	$Pr(> t)$
Intercept	46.2660	0.3088	149.82	< 0.0001
$\cos(2\pi t)$	-26.7079	0.4367	-61.15	< 0.0001
$\sin(2\pi t)$	-2.1697	0.4367	-4.97	<0.0001

```
> har.=harmonic(tempdub, 1)
> model4=lm(tempdub~har.)
> summary(model4)
```

In this output, time is measured in years, with 1964 as the starting value and a frequency of 1 per year. A graph of the time series values together with the fitted cosine curve is shown in Exhibit 3.6. The trend fits the data quite well with the exception of most of the January values, where the observations are lower than the model would predict.

Exhibit 3.6 Cosine Trend for the Temperature Series

```
> win.graph(width=4.875, height=2.5, pointsize=8)
> plot(ts(fitted(model4), freq=12, start=c(1964,1)),
      ylab='Temperature', type='l',
      ylim=range(c(fitted(model4), tempdub))); points(tempdub)
> # ylim ensures that the y axis range fits the raw data and the
  fitted values
```

Additional cosine functions at other frequencies will frequently be used to model cyclical trends. For monthly series, the higher harmonic frequencies, such as 2/12 and 3/12, are especially pertinent and will sometimes improve the fit at the expense of add-

ing more parameters to the model. In fact, it may be shown that any periodic trend with period 12 may be expressed exactly by the sum of six pairs of cosine-sine functions. These ideas are discussed in detail in Fourier analysis or spectral analysis. We pursue these ideas further in Chapters 13 and 14.

3.4 Reliability and Efficiency of Regression Estimates

We assume that the series is represented as $Y_t = \mu_t + X_t$, where μ_t is a deterministic trend of the kind considered above and $\{X_t\}$ is a zero-mean stationary process with autocovariance and autocorrelation functions γ_k and ρ_k , respectively. Ordinary regression estimates parameters in a linear model according to the criterion of least squares regardless of whether we are fitting linear time trends, seasonal means, cosine curves, or whatever.

We first consider the easiest case—the seasonal means. As mentioned earlier, the least squares estimates of the seasonal means are just seasonal averages; thus, if we have N (complete) years of monthly data, we can write the estimate for the mean for the j th season as

$$\hat{\beta}_j = \frac{1}{N} \sum_{i=0}^{N-1} Y_{j+12i}$$

Since $\hat{\beta}_j$ is an average like \bar{Y} but uses only every 12th observation, Equation (3.2.3) can be easily modified to give $Var(\hat{\beta}_j)$. We replace n by N (years) and ρ_k by ρ_{12k} to get

$$Var(\hat{\beta}_j) = \frac{\gamma_0}{N} \left[1 + 2 \sum_{k=1}^{N-1} \left(1 - \frac{k}{N} \right) \rho_{12k} \right] \quad \text{for } j = 1, 2, \dots, 12 \quad (3.4.1)$$

We notice that if $\{X_t\}$ is white noise, then $Var(\hat{\beta}_j)$ reduces to γ_0/N , as expected. Furthermore, if several ρ_k are nonzero but $\rho_{12k} = 0$, then we still have $Var(\hat{\beta}_j) = \gamma_0/N$. In any case, only the seasonal autocorrelations, $\rho_{12}, \rho_{24}, \rho_{36}, \dots$, enter into Equation (3.4.1). Since N will rarely be very large (except perhaps for quarterly data), approximations like those shown in Equation (3.2.5) will usually not be useful.

We turn now to the cosine trends expressed as in Equation (3.3.8). For any frequency of the form $f = m/n$, where m is an integer satisfying $1 \leq m < n/2$, explicit expressions are available for the estimates $\hat{\beta}_1$ and $\hat{\beta}_2$, the amplitudes of the cosine and sine:

$$\hat{\beta}_1 = \frac{2}{n} \sum_{t=1}^n \left[\cos\left(\frac{2\pi mt}{n}\right) Y_t \right], \quad \hat{\beta}_2 = \frac{2}{n} \sum_{t=1}^n \left[\sin\left(\frac{2\pi mt}{n}\right) Y_t \right] \quad (3.4.2)$$

(These are effectively the correlations between the time series $\{Y_t\}$ and the cosine and sine waves with frequency m/n .)

Because these are linear functions of $\{Y_t\}$, we may evaluate their variances using Equation (2.2.6). We find

$$Var(\hat{\beta}_1) = \frac{2\gamma_0}{n} \left[1 + \frac{4}{n} \sum_{s=2}^n \sum_{t=1}^{s-1} \cos\left(\frac{2\pi mt}{n}\right) \cos\left(\frac{2\pi ms}{n}\right) \rho_{s-t} \right] \quad (3.4.3)$$

where we have used the fact that $\sum_{t=1}^n [\cos(2\pi mt/n)]^2 = n/2$. However, the double sum in Equation (3.4.3) does not, in general, reduce further. A similar expression holds for $Var(\hat{\beta}_2)$ if we replace the cosines by sines.

If $\{X_t\}$ is white noise, we get just $2\gamma_0/n$. If $\rho_1 \neq 0$, $\rho_k = 0$ for $k > 1$, and $m/n = 1/12$, then the variance reduces to

$$Var(\hat{\beta}_1) = \frac{2\gamma_0}{n} \left[1 + \frac{4\rho_1}{n} \sum_{t=1}^{n-1} \cos\left(\frac{\pi t}{6}\right) \cos\left(\frac{\pi t+1}{6}\right) \right] \quad (3.4.4)$$

To illustrate the effect of the cosine terms, we have calculated some representative values:

n	$Var(\hat{\beta}_1)$
25	$\left(\frac{2\gamma_0}{n}\right)(1 + 1.71\rho_1)$
50	$\left(\frac{2\gamma_0}{n}\right)(1 + 1.75\rho_1)$
500	$\left(\frac{2\gamma_0}{n}\right)(1 + 1.73\rho_1)$
∞	$\left(\frac{2\gamma_0}{n}\right)\left(1 + 2\rho_1 \cos\left(\frac{\pi}{6}\right)\right) = \left(\frac{2\gamma_0}{n}\right)(1 + 1.732\rho_1)$

If $\rho_1 = -0.4$, then the large sample multiplier in Equation (3.4.5) is $1+1.732(-0.4) = 0.307$ and the variance is reduced by about 70% when compared with the white noise case.

In some circumstances, seasonal means and cosine trends could be considered as competing models for a cyclical trend. If the simple cosine model is an adequate model, how much do we lose if we use the less parsimonious seasonal means model? To approach this problem, we must first consider how to compare the models. The parameters themselves are not directly comparable, but we can compare the estimates of the trend at comparable time points.

Consider the two estimates for the trend in January; that is, μ_1 . With seasonal means, this estimate is just the January average, which has variance given by Equation (3.4.1). With the cosine trend model, the corresponding estimate is

$$\hat{\mu}_1 = \hat{\beta}_0 + \hat{\beta}_1 \cos\left(\frac{2\pi}{12}\right) + \hat{\beta}_2 \sin\left(\frac{2\pi}{12}\right)$$

To compute the variance of this estimate, we need one more fact: With this model, the estimates $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are uncorrelated.[†] This follows from the orthogonality relationships of the cosines and sines involved. See Bloomfield (1976) or Fuller (1996) for more details. For the cosine model, then, we have

$$Var(\hat{\mu}_1) = Var(\hat{\beta}_0) + Var(\hat{\beta}_1) \left[\cos\left(\frac{2\pi}{12}\right) \right]^2 + Var(\hat{\beta}_2) \left[\sin\left(\frac{2\pi}{12}\right) \right]^2 \quad (3.4.6)$$

For our first comparison, assume that the stochastic component is white noise. Then the variance of our estimate in the seasonal means model is just γ_0/N . For the cosine model, we use Equation (3.4.6), and Equation (3.4.4) and its sine equivalent, to obtain

$$\begin{aligned} Var(\hat{\mu}_1) &= \frac{\gamma_0}{n} \left\{ 1 + 2 \left[\cos\left(\frac{\pi}{6}\right) \right]^2 + 2 \left[\sin\left(\frac{\pi}{6}\right) \right]^2 \right\} \\ &= 3 \frac{\gamma_0}{n} \end{aligned}$$

since $(\cos\theta)^2 + (\sin\theta)^2 = 1$. Thus the ratio of the standard deviation in the cosine model to that in the seasonal means model is

$$\sqrt{\frac{3\gamma_0/n}{\gamma_0/N}} = \sqrt{\frac{3N}{n}}$$

In particular, for the monthly temperature series, we have $n = 144$ and $N = 12$; thus, the ratio is

$$\sqrt{\frac{3(12)}{144}} = 0.5$$

Thus, in the cosine model, we estimate the January effect with a standard deviation that is only half as large as it would be if we estimated with a seasonal means model—a substantial gain. (Of course, this assumes that the cosine trend plus white noise model is the correct model.)

Suppose now that the stochastic component is such that $\rho_1 \neq 0$ but $\rho_k = 0$ for $k > 1$. With a seasonal means model, the variance of the estimated January effect will be unchanged (see Equation (3.4.1) on page 36). For the cosine trend model, if we have a reasonably large sample size, we may use Equation (3.4.5), an identical expression for $Var(\hat{\beta}_2)$, and Equation (3.2.3) on page 28 for $Var(\hat{\beta}_0)$ to obtain

[†] This assumes that $1/12$ is a “Fourier frequency”; that is, it is of the form m/n . Otherwise, these estimates are only approximately uncorrelated.

$$\begin{aligned} Var(\hat{\mu}_1) &= \frac{\gamma_0}{n} \left\{ 1 + 2\rho_1 + 2 \left[1 + 2\rho_1 \cos\left(\frac{2\pi}{12}\right) \right] \right\} \\ &= \frac{\gamma_0}{n} \left\{ 3 + 2\rho_1 \left[1 + 2 \cos\left(\frac{\pi}{6}\right) \right] \right\} \end{aligned}$$

If $\rho_1 = -0.4$, then we have $0.814\gamma_0/n$, and the ratio of the standard deviation in the cosine case to the standard deviation in the seasonal means case is

$$\sqrt{\left[\frac{(0.814\gamma_0)/n}{\gamma_0/N} \right]} = \sqrt{\frac{0.814N}{n}}$$

If we take $n = 144$ and $N = 12$, the ratio is

$$\sqrt{\frac{0.814(12)}{144}} = 0.26$$

a very substantial reduction indeed!

We now turn to linear time trends. For these trends, an alternative formula to Equation (3.3.2) on page 30 for $\hat{\beta}_1$ is more convenient. It can be shown that the least squares estimate of the slope may be written

$$\hat{\beta}_1 = \frac{\sum_{t=1}^n (t - \bar{t}) Y_t}{\sum_{t=1}^n (t - \bar{t})^2} \quad (3.4.7)$$

Since the estimate is a linear combination of Y -values, some progress can be made in evaluating its variance. We have

$$Var(\hat{\beta}_1) = \frac{12\gamma_0}{n(n^2-1)} \left[1 + \frac{24}{n(n^2-1)} \sum_{s=2}^n \sum_{t=1}^{s-1} (t - \bar{t})(s - \bar{t}) \rho_{s-t} \right] \quad (3.4.8)$$

where we have used $\sum_{t=1}^n (t - \bar{t})^2 = n(n^2 - 1)/12$. Again the double sum does not in general reduce.

To illustrate the effect of Equation (3.4.8), consider again the case where $\rho_1 \neq 0$ but $\rho_k = 0$ for $k > 1$. Then, after some algebraic manipulation, again involving the sum of consecutive integers and their squares, Equation (3.4.8) can be reduced to

$$Var(\hat{\beta}_1) = \frac{12\gamma_0}{n(n^2-1)} \left[1 + 2\rho_1 \left(1 - \frac{3}{n} \right) \right]$$

For large n , we can neglect the $3/n$ term and use

$$Var(\hat{\beta}_1) = \frac{12\gamma_0(1 + 2\rho_1)}{n(n^2-1)} \quad (3.4.9)$$

If $\rho_1 = -0.4$, then $1 + 2\rho_1 = 0.2$, and then the variance of $\hat{\beta}_1$ is only 20% of what it would be if $\{X_t\}$ were white noise. Of course, if $\rho_1 > 0$, then the variance would be larger than for the white noise case.

We turn now to comparing the least squares estimates with the so-called **best linear unbiased estimates** (BLUE) or the **generalized least squares** (GLS) estimates. If the stochastic component $\{X_t\}$ is not white noise, estimates of the unknown parameters in the trend function may be made; they are linear functions of the data, are unbiased, and have the smallest variances among all such estimates—the so-called BLUE or GLS estimates. These estimates and their variances can be expressed fairly explicitly by using certain matrices and their inverses. (Details may be found in Draper and Smith (1981).) However, constructing these estimates requires complete knowledge of the covariance function of the stochastic component, a function that is unknown in virtually all real applications. It is possible to iteratively estimate the covariance function for $\{X_t\}$ based on a preliminary estimate of the trend. The trend is then estimated again using the estimated covariance function for $\{X_t\}$ and thus iterated to an approximate BLUE for the trend. This method will not be pursued here, however.

Fortunately, there are some results based on large sample sizes that support the use of the simpler least squares estimates for the types of trends that we have considered. In particular, we have the following result (see Fuller (1996), pp. 476–480, for more details): We assume that the trend is either a polynomial in time, a trigonometric polynomial, seasonal means, or a linear combination of these. Then, for a very general stationary stochastic component $\{X_t\}$, the least squares estimates for the trend have the same variance as the best linear unbiased estimates for large sample sizes.

Although the simple least squares estimates may be asymptotically efficient, it does not follow that the estimated standard deviations of the coefficients as printed out by all regression routines are correct. We shall elaborate on this point in the next section. We also caution the reader that the result above is restricted to certain kinds of trends and cannot, in general, be extended to regression on arbitrary predictor variables, such as other time series. For example, Fuller (1996, pp. 518–522) shows that if $Y_t = \beta Z_t + X_t$, where $\{X_t\}$ has a simple stochastic structure but $\{Z_t\}$ is also a stationary series, then the least squares estimate of β can be very inefficient and biased even for large samples.

3.5 Interpreting Regression Output

We have already noted that the standard regression routines calculate least squares estimates of the unknown regression coefficients—the betas. As such, the estimates are reasonable under minimal assumptions on the stochastic component $\{X_t\}$. However, some of the properties of the regression output depend heavily on the usual regression assumption that $\{X_t\}$ is white noise, and some depend on the further assumption that $\{X_t\}$ is approximately normally distributed. We begin with the items that depend least on the assumptions.

Consider the regression output shown in Exhibit 3.7. We shall write $\hat{\mu}_t$ for the estimated trend regardless of the assumed parametric form for μ_t . For example, for the linear time trend, we have $\mu_t = \beta_0 + \beta_1 t$. For each t , the unobserved stochastic component

X_t can be estimated (predicted) by $\hat{Y}_t = \hat{\mu}_t$. If the $\{X_t\}$ process has constant variance, then we can estimate the standard deviation of X_t , namely $\sqrt{\gamma_0}$, by the **residual standard deviation**

$$s = \sqrt{\frac{1}{n-p} \sum_{t=1}^n (Y_t - \hat{\mu}_t)^2} \quad (3.5.1)$$

where p is the number of parameters estimated in $\hat{\mu}_t$ and $n - p$ is the so-called *degrees of freedom* for s . The value of s gives an absolute measure of the goodness of fit of the estimated trend—the smaller the value of s , the better the fit. However, a value of s of, say, 60.74 is somewhat difficult to interpret.

A unitless measure of the goodness of fit of the trend is the value of R^2 , also called the **coefficient of determination** or multiple R -squared. One interpretation of R^2 is that it is the square of the sample correlation coefficient between the observed series and the estimated trend. It is also the fraction of the variation in the series that is explained by the estimated trend. Exhibit 3.7 is a more complete regression output when fitting the straight line to the random walk data. This extends what we saw in Exhibit 3.1 on page 31.

Exhibit 3.7 Regression Output for Linear Trend Fit of Random Walk

	Estimate	Std. Error	t-value	Pr(> t)
Intercept	-1.007888	0.297245	-3.39	0.00126
Time	0.134087	0.008475	15.82	< 0.0001

Residual standard error 1.137 with 58 degrees of freedom

Multiple R-Squared 0.812

Adjusted R-squared 0.809

F-statistic 250.3 with 1 and 58 df; p -value < 0.0001

```
> model1=lm(rwalk~time(rwalk))
> summary(model1)
```

According to Exhibit 3.7, about 81% of the variation in the random walk series is explained by the linear time trend. The adjusted R-squared value is a small adjustment to R^2 that yields an approximately unbiased estimate based on the number of parameters estimated in the trend. It is useful for comparing models with different numbers of parameters. Various formulas for computing R^2 may be found in any book on regression, such as Draper and Smith (1981). The standard deviations of the coefficients labeled Std. Error on the output need to be interpreted carefully. They are appropriate only when the stochastic component is white noise—the usual regression assumption.

For example, in Exhibit 3.7 the value 1.137 is obtained from the square root of the value given by Equation (3.4.8) when $\rho_k = 0$ for $k > 0$ and with γ_0 estimated by s^2 , that is, to within rounding,

$$0.008475 = \sqrt{\frac{12(1.137)^2}{60(60^2 - 1)}}$$

The important point is that these standard deviations assume a white noise stochastic component that will rarely be true for time series.

The t -values or t -ratios shown in Exhibit 3.7 are just the estimated regression coefficients, each divided by their respective standard errors. If the stochastic component is normally distributed white noise, then these ratios provide appropriate test statistics for checking the significance of the regression coefficients. In each case, the null hypothesis is that the corresponding unknown regression coefficient is zero. The significance levels and p -values are determined from the t -distribution with $n - p$ degrees of freedom.

3.6 Residual Analysis

As we have already noted, the unobserved stochastic component $\{X_t\}$ can be estimated, or predicted, by the **residual**

$$\hat{X}_t = Y_t - \hat{\mu}_t \quad (3.6.1)$$

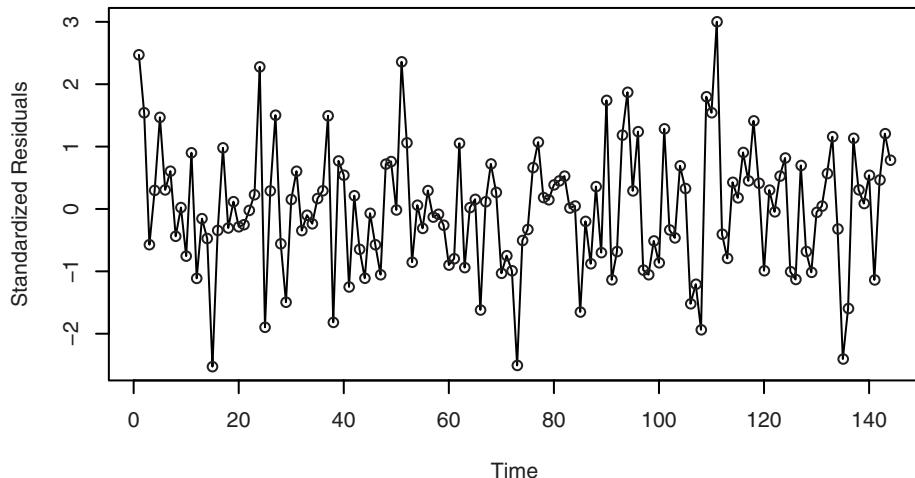
Predicted is really a better term. We reserve the term estimate for the guess of an unknown parameter and the term predictor for an estimate of an unobserved random variable. We call \hat{X}_t the residual corresponding to the t th observation. If the trend model is reasonably correct, then the residuals should behave roughly like the true stochastic component, and various assumptions about the stochastic component can be assessed by looking at the residuals. If the stochastic component is white noise, then the residuals should behave roughly like independent (normal) random variables with zero mean and standard deviation s . Since a least squares fit of any trend containing a constant term automatically produces residuals with a zero mean, we might consider standardizing the residuals as \hat{X}_t/s . However, most statistics software will produce standardized residuals using a more complicated standard error in the denominator that takes into account the specific regression model being fit.

With the residuals or standardized residuals in hand, the next step is to examine various residual plots. We first look at the plot of the residuals over time. If the data are possibly seasonal, we should use plotting symbols as we did in Exhibit 1.9 on page 7, so that residuals associated with the same season can be identified easily.

We will use the monthly average temperature series which we fitted with seasonal means as our first example to illustrate some of the ideas of residual analysis. Exhibit 1.7 on page 6 shows the time series plot of that series. Exhibit 3.8 shows a time series plot for the standardized residuals of the monthly temperature data fitted by seasonal means. If the stochastic component is white noise and the trend is adequately modeled, we would expect such a plot to suggest a rectangular scatter with no discernible trends whatsoever. There are no striking departures from randomness apparent in this display.

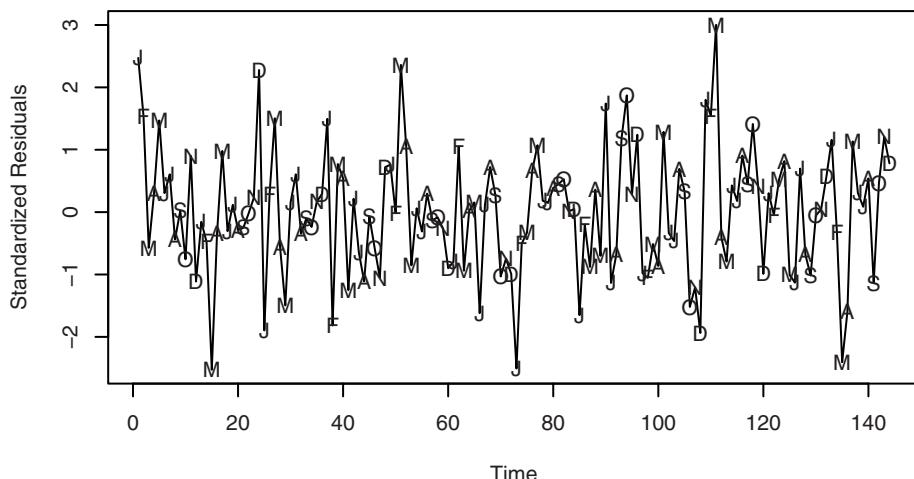
Exhibit 3.9 repeats the time series plot but now with seasonal plotting symbols. Again there are no apparent patterns relating to different months of the year.

Exhibit 3.8 Residuals versus Time for Temperature Seasonal Means



```
> plot(y=rstudent(model3),x=as.vector(time(tempdub)),
       xlab='Time',ylab='Standardized Residuals',type='o')
```

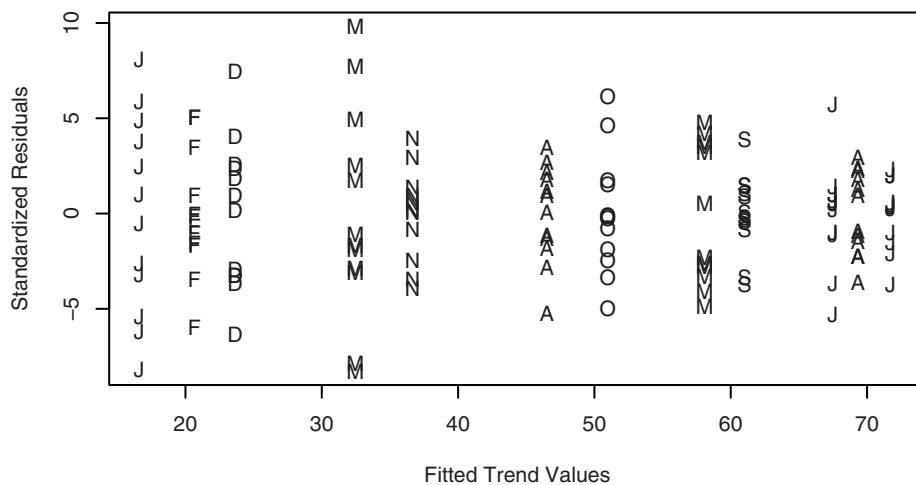
Exhibit 3.9 Residuals versus Time with Seasonal Plotting Symbols



```
> plot(y=rstudent(model3),x=as.vector(time(tempdub)),xlab='Time',
       ylab='Standardized Residuals',type='l')
> points(y=rstudent(model3),x=as.vector(time(tempdub)),
       pch=as.vector(season(tempdub)))
```

Next we look at the standardized residuals versus the corresponding trend estimate, or fitted value, as in Exhibit 3.10. Once more we are looking for patterns. Are small residuals associated with small fitted trend values and large residuals with large fitted trend values? Is there less variation for residuals associated with certain sized fitted trend values or more variation with other fitted trend values? There is somewhat more variation for the March residuals and less for November, but Exhibit 3.10 certainly does not indicate any dramatic patterns that would cause us to doubt the seasonal means model.

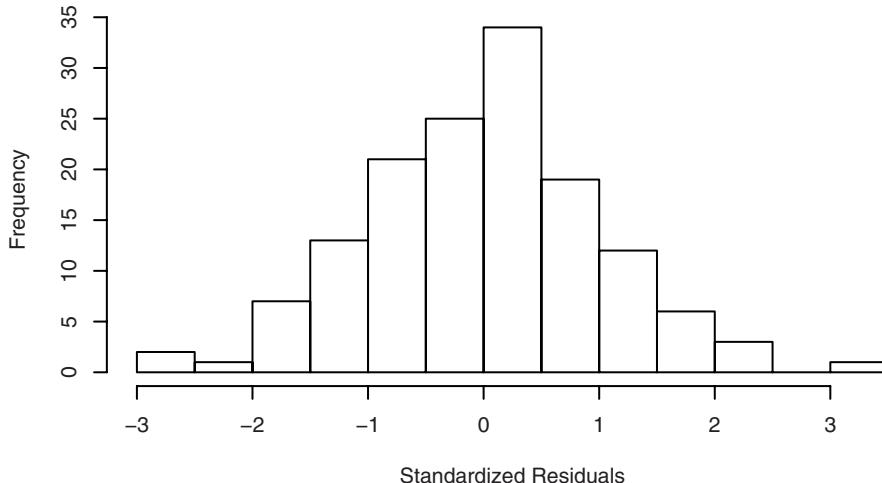
Exhibit 3.10 Standardized Residuals versus Fitted Values for the Temperature Seasonal Means Model



```
> plot(y=rstudent(model3),x=as.vector(fitted(model3)),
       xlab='Fitted Trend Values',
       ylab='Standardized Residuals',type='n')
> points(y=rstudent(model3),x=as.vector(fitted(model3)),
          pch=as.vector(season(tempdub)))
```

Gross nonnormality can be assessed by plotting a histogram of the residuals or standardized residuals. Exhibit 3.11 displays a frequency histogram of the standardized residuals from the seasonal means model for the temperature series. The plot is somewhat symmetric and tails off at both the high and low ends as a normal distribution does.

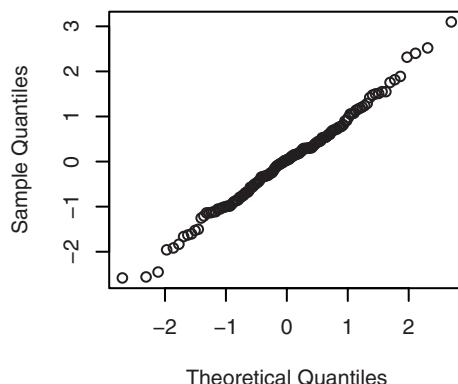
Exhibit 3.11 Histogram of Standardized Residuals from Seasonal Means Model



```
> hist(rstudent(model3), xlab='Standardized Residuals')
```

Normality can be checked more carefully by plotting the so-called normal scores or quantile-quantile (QQ) plot. Such a plot displays the quantiles of the data versus the theoretical quantiles of a normal distribution. With normally distributed data, the QQ plot looks approximately like a straight line. Exhibit 3.12 shows the QQ normal scores plot for the standardized residuals from the seasonal means model for the temperature series. The straight-line pattern here supports the assumption of a normally distributed stochastic component in this model.

Exhibit 3.12 Q-Q Plot: Standardized Residuals of Seasonal Means Model



```
> win.graph(width=2.5, height=2.5, pointsize=8)
> qqnorm(rstudent(model3))
```

An excellent test of normality is known as the Shapiro-Wilk test.[†] It essentially calculates the correlation between the residuals and the corresponding normal quantiles. The lower this correlation, the more evidence we have against normality. Applying that test to these residuals gives a test statistic of $W = 0.9929$ with a p -value of 0.6954. We cannot reject the null hypothesis that the stochastic component of this model is normally distributed.

Independence in the stochastic component can be tested in several ways. The **runs test** examines the residuals in sequence to look for patterns—patterns that would give evidence against independence. Runs above or below their median are counted. A small number of runs would indicate that neighboring residuals are positively dependent and tend to “hang together” over time. On the other hand, too many runs would indicate that the residuals oscillate back and forth across their median. Then neighboring residuals are negatively dependent. So either too few or too many runs lead us to reject independence. Performing a runs test[‡] on these residuals produces the following values: observed runs = 65, expected runs = 72.875, which leads to a p -value of 0.216 and we cannot reject independence of the stochastic component in this seasonal means model.

The Sample Autocorrelation Function

Another very important diagnostic tool for examining dependence is the sample autocorrelation function. Consider any sequence of data Y_1, Y_2, \dots, Y_n —whether residuals, standardized residuals, original data, or some transformation of data. Tentatively assuming stationarity, we would like to estimate the autocorrelation function ρ_k for a variety of lags $k = 1, 2, \dots$. The obvious way to do this is to compute the sample correlation between the pairs k units apart in time. That is, among $(Y_1, Y_{1+k}), (Y_2, Y_{2+k}), (Y_3, Y_{3+k}), \dots$, and (Y_{n-k}, Y_n) . However, we modify this slightly, taking into account that we are assuming stationarity, which implies a common mean and variance for the series. With this in mind, we define the **sample autocorrelation function**, r_k , at lag k as

$$r_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \quad \text{for } k = 1, 2, \dots \quad (3.6.2)$$

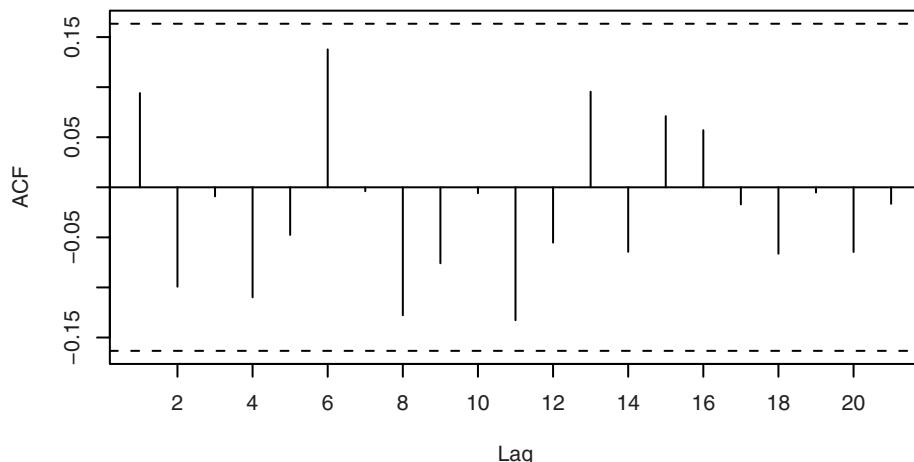
Notice that we used the “grand mean,” \bar{Y} , in all places and have also divided by the “grand sum of squares” rather than the product of the two separate standard deviations used in the ordinary correlation coefficient. We also note that the denominator is a sum of n squared terms while the numerator contains only $n - k$ cross products. For a variety of reasons, this has become the standard definition for the sample autocorrelation function. A plot of r_k versus lag k is often called a **correlogram**.

[†] Royston, P. (1982) “An Extension of Shapiro and Wilk’s W Test for Normality to Large Samples.” *Applied Statistics*, **31**, 115–124.

[‡] R code: `runs(rstudent(model3))`

In our present context, we are interested in discovering possible dependence in the stochastic component; therefore the sample autocorrelation function for the standardized residuals is of interest. Exhibit 3.13 displays the sample autocorrelation for the standardized residuals from the seasonal means model of the temperature series. All values are within the horizontal dashed lines, which are placed at zero plus and minus two approximate standard errors of the sample autocorrelations, namely $\pm 2/\sqrt{n}$. The values of r_k are, of course, estimates of ρ_k . As such, they have their own sampling distributions, standard errors, and other properties. For now we shall use r_k as a descriptive tool and defer discussion of those topics until Chapters 6 and 8. According to Exhibit 3.13, for $k = 1, 2, \dots, 21$, none of the hypotheses $\rho_k = 0$ can be rejected at the usual significance levels, and it is reasonable to infer that the stochastic component of the series is white noise.

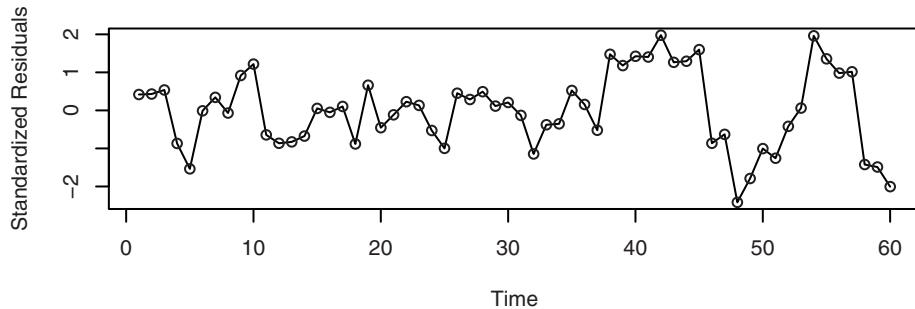
Exhibit 3.13 Sample Autocorrelation of Residuals of Seasonal Means Model



```
> win.graph(width=4.875,height=3,pointsize=8)
> acf(rstudent(model3))
```

As a second example consider the standardized residuals from fitting a straight line to the random walk time series. Recall Exhibit 3.2 on page 31, which shows the data and fitted line. A time series plot of the standardized residuals is shown in Exhibit 3.14.

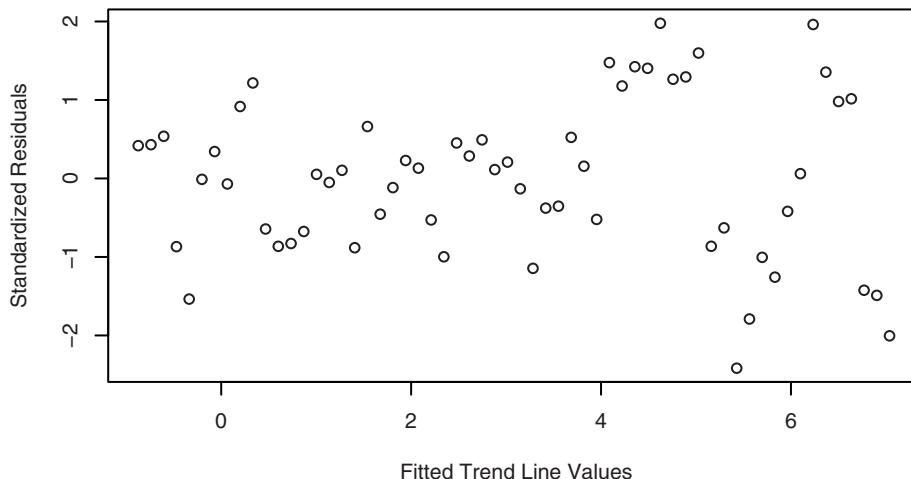
Exhibit 3.14 Residuals from Straight Line Fit of the Random Walk



```
> plot(y=rstudent(model1),x=as.vector(time(rwalk)),
       ylab='Standardized Residuals',xlab='Time',type='o')
```

In this plot, the residuals “hang together” too much for white noise—the plot is too smooth. Furthermore, there seems to be more variation in the last third of the series than in the first two-thirds. Exhibit 3.15 shows a similar effect with larger residuals associated with larger fitted values.

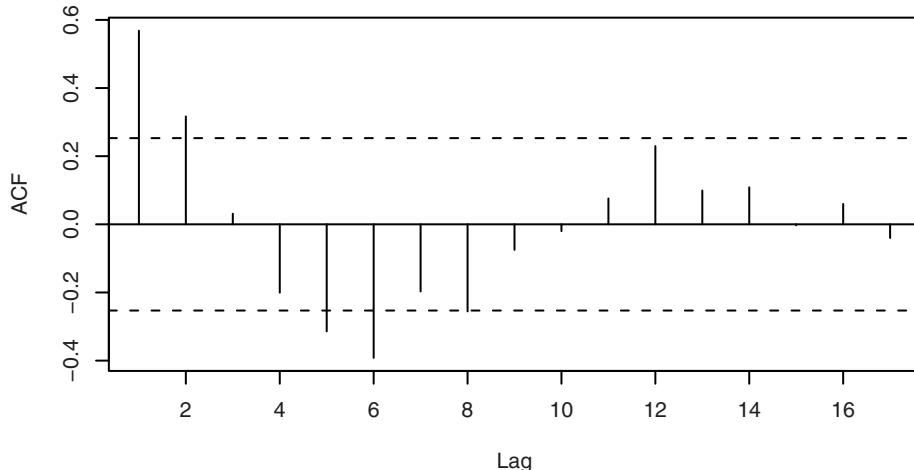
Exhibit 3.15 Residuals versus Fitted Values from Straight Line Fit



```
> win.graph(width=4.875, height=3,pointsize=8)
> plot(y=rstudent(model1),x=fitted(model1),
       ylab='Standardized Residuals',xlab='Fitted Trend Line Values',
       type='p')
```

The sample autocorrelation function of the standardized residuals, shown in Exhibit 3.16, confirms the smoothness of the time series plot that we observed in Exhibit 3.14. The lag 1 and lag 2 autocorrelations exceed two standard errors above zero and the lag 5 and lag 6 autocorrelations more than two standard errors below zero. This is not what we expect from a white noise process.

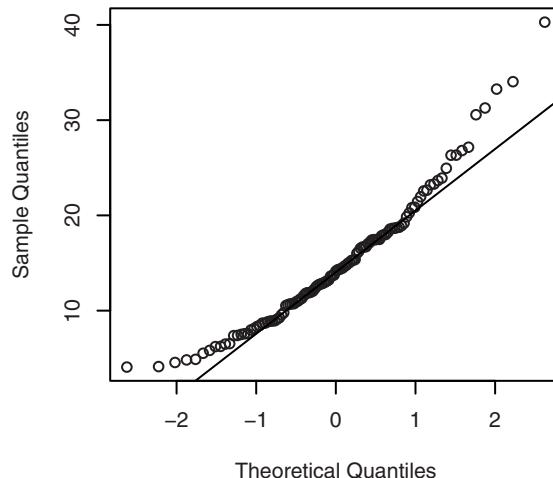
Exhibit 3.16 Sample Autocorrelation of Residuals from Straight Line Model



```
> acf(rstudent(model1))
```

Finally, we return to the annual rainfall in Los Angeles shown in Exhibit 1.1 on page 2. We found no evidence of dependence in that series, but we now look for evidence against normality. Exhibit 3.17 displays the normal quantile-quantile plot for that series. We see considerable curvature in the plot. A line passing through the first and third normal quartiles helps point out the departure from a straight line in the plot.

Exhibit 3.17 Quantile-Quantile Plot of Los Angeles Annual Rainfall Series



```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(larain); qqline(larain)
```

3.7 Summary

This chapter is concerned with describing, modeling, and estimating deterministic trends in time series. The simplest deterministic “trend” is a constant-mean function. Methods of estimating a constant mean were given but, more importantly, assessment of the accuracy of the estimates under various conditions was considered. Regression methods were then pursued to estimate trends that are linear or quadratic in time. Methods for modeling cyclical or seasonal trends came next, and the reliability and efficiency of all of these regression methods were investigated. The final section began our study of residual analysis to investigate the quality of the fitted model. This section also introduced the important sample autocorrelation function, which we will revisit throughout the remainder of the book.

EXERCISES

- 3.1 Verify Equation (3.3.2) on page 30, for the least squares estimates of β_0 and of β_1 when the model $Y_t = \beta_0 + \beta_1 t + X_t$ is considered.
- 3.2 Suppose $Y_t = \mu + e_t - e_{t-1}$. Find $Var(\bar{Y})$. Note any unusual results. In particular, compare your answer to what would have been obtained if $Y_t = \mu + e_t$. (Hint: You may avoid Equation (3.2.3) on page 28 by first doing some algebraic simplification on $\sum_{t=1}^n (e_t - e_{t-1})$.)

- 3.3 Suppose $Y_t = \mu + e_t + e_{t-1}$. Find $\text{Var}(\bar{Y})$. Compare your answer to what would have been obtained if $Y_t = \mu + e_t$. Describe the effect that the autocorrelation in $\{Y_t\}$ has on $\text{Var}(\bar{Y})$.
- 3.4 The data file `hours` contains monthly values of the average hours worked per week in the U.S. manufacturing sector for July 1982 through June 1987.
- Display and interpret the time series plot for these data.
 - Now construct a time series plot that uses separate plotting symbols for the various months. Does your interpretation change from that in part (a)?
- 3.5 The data file `wages` contains monthly values of the average hourly wages (in dollars) for workers in the U.S. apparel and textile products industry for July 1981 through June 1987.
- Display and interpret the time series plot for these data.
 - Use least squares to fit a linear time trend to this time series. Interpret the regression output. Save the standardized residuals from the fit for further analysis.
 - Construct and interpret the time series plot of the standardized residuals from part (b).
 - Use least squares to fit a quadratic time trend to the wages time series. Interpret the regression output. Save the standardized residuals from the fit for further analysis.
 - Construct and interpret the time series plot of the standardized residuals from part (d).
- 3.6 The data file `beersales` contains monthly U.S. beer sales (in millions of barrels) for the period January 1975 through December 1990.
- Display and interpret the plot the time series plot for these data.
 - Now construct a time series plot that uses separate plotting symbols for the various months. Does your interpretation change from that in part (a)?
 - Use least squares to fit a seasonal-means trend to this time series. Interpret the regression output. Save the standardized residuals from the fit for further analysis.
 - Construct and interpret the time series plot of the standardized residuals from part (c). Be sure to use proper plotting symbols to check on seasonality in the standardized residuals.
 - Use least squares to fit a seasonal-means plus quadratic time trend to the beer sales time series. Interpret the regression output. Save the standardized residuals from the fit for further analysis.
 - Construct and interpret the time series plot of the standardized residuals from part (e). Again use proper plotting symbols to check for any remaining seasonality in the residuals.
- 3.7 The data file `winnebago` contains monthly unit sales of recreational vehicles from Winnebago, Inc., from November 1966 through February 1972.
- Display and interpret the time series plot for these data.
 - Use least squares to fit a line to these data. Interpret the regression output. Plot the standardized residuals from the fit as a time series. Interpret the plot.
 - Now take natural logarithms of the monthly sales figures and display and

- interpret the time series plot of the transformed values.
- (d) Use least squares to fit a line to the logged data. Display and interpret the time series plot of the standardized residuals from this fit.
- (e) Now use least squares to fit a seasonal-means plus linear time trend to the logged sales time series and save the standardized residuals for further analysis. Check the statistical significance of each of the regression coefficients in the model.
- (f) Display the time series plot of the standardized residuals obtained in part (e). Interpret the plot.
- 3.8** The data file *retail* lists total U.K. (United Kingdom) retail sales (in billions of pounds) from January 1986 through March 2007. The data are not “seasonally adjusted,” and year 2000 = 100 is the base year.
- (a) Display and interpret the time series plot for these data. Be sure to use plotting symbols that permit you to look for seasonality.
- (b) Use least squares to fit a seasonal-means plus linear time trend to this time series. Interpret the regression output and save the standardized residuals from the fit for further analysis.
- (c) Construct and interpret the time series plot of the standardized residuals from part (b). Be sure to use proper plotting symbols to check on seasonality.
- 3.9** The data file *prescrip* gives monthly U.S. prescription costs for the months August 1986 to March 1992. These data are from the State of New Jersey’s Prescription Drug Program and are the cost per prescription claim.
- (a) Display and interpret the time series plot for these data. Use plotting symbols that permit you to look for seasonality.
- (b) Calculate and plot the sequence of month-to-month percentage changes in the prescription costs. Again, use plotting symbols that permit you to look for seasonality.
- (c) Use least squares to fit a cosine trend with fundamental frequency 1/12 to the percentage change series. Interpret the regression output. Save the standardized residuals.
- (d) Plot the sequence of standardized residuals to investigate the adequacy of the cosine trend model. Interpret the plot.
- 3.10** (Continuation of Exercise 3.4) Consider the *hours* time series again.
- (a) Use least squares to fit a quadratic trend to these data. Interpret the regression output and save the standardized residuals for further analysis.
- (b) Display a sequence plot of the standardized residuals and interpret. Use monthly plotting symbols so that possible seasonality may be readily identified.
- (c) Perform the Runs test of the standardized residuals and interpret the results.
- (d) Calculate and interpret the sample autocorrelations for the standardized residuals.
- (e) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.

- 3.11** (Continuation of Exercise 3.5) Return to the `wages` series.
- (a) Consider the residuals from a least squares fit of a quadratic time trend.
 - (b) Perform a runs test on the standardized residuals and interpret the results.
 - (c) Calculate and interpret the sample autocorrelations for the standardized residuals.
 - (d) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.
- 3.12** (Continuation of Exercise 3.6) Consider the time series in the data file `beersales`.
- (a) Obtain the residuals from the least squares fit of the seasonal-means plus quadratic time trend model.
 - (b) Perform a runs test on the standardized residuals and interpret the results.
 - (c) Calculate and interpret the sample autocorrelations for the standardized residuals.
 - (d) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.
- 3.13** (Continuation of Exercise 3.7) Return to the `winnebago` time series.
- (a) Calculate the least squares residuals from a seasonal-means plus linear time trend model on the logarithms of the sales time series.
 - (b) Perform a runs test on the standardized residuals and interpret the results.
 - (c) Calculate and interpret the sample autocorrelations for the standardized residuals.
 - (d) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.
- 3.14** (Continuation of Exercise 3.8) The data file `retail` contains U.K. monthly retail sales figures.
- (a) Obtain the least squares residuals from a seasonal-means plus linear time trend model.
 - (b) Perform a runs test on the standardized residuals and interpret the results.
 - (c) Calculate and interpret the sample autocorrelations for the standardized residuals.
 - (d) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.
- 3.15** (Continuation of Exercise 3.9) Consider again the `prescrip` time series.
- (a) Save the standardized residuals from a least squares fit of a cosine trend with fundamental frequency 1/12 to the percentage change time series.
 - (b) Perform a runs test on the standardized residuals and interpret the results.
 - (c) Calculate and interpret the sample autocorrelations for the standardized residuals.
 - (d) Investigate the normality of the standardized residuals (error terms). Consider histograms and normal probability plots. Interpret the plots.

- 3.16** Suppose that a stationary time series, $\{Y_t\}$, has an autocorrelation function of the form $\rho_k = \phi^k$ for $k > 0$, where ϕ is a constant in the range $(-1, +1)$.

(a) Show that $Var(\bar{Y}) = \frac{\gamma_0}{n} \left[\frac{1+\phi}{1-\phi} - \frac{2\phi(1-\phi^n)}{(1-\phi)^2} \right]$.

(Hint: Use Equation (3.2.3) on page 28, the finite geometric sum

$$\sum_{k=0}^n \phi^k = \frac{1-\phi^{n+1}}{1-\phi}, \text{ and the related sum } \sum_{k=0}^n k\phi^{k-1} = \frac{d}{d\phi} \left[\sum_{k=0}^n \phi^k \right].$$

(b) If n is large, argue that $Var(\bar{Y}) \approx \frac{\gamma_0}{n} \left[\frac{1+\phi}{1-\phi} \right]$.

- (c) Plot $(1+\phi)/(1-\phi)$ for ϕ over the range -1 to $+1$. Interpret the plot in terms of the precision in estimating the process mean.

- 3.17** Verify Equation (3.2.6) on page 29. (Hint: You will need the fact that

$$\sum_{k=0}^{\infty} \phi^k = \frac{1}{1-\phi} \text{ for } -1 < \phi < +1.$$

- 3.18** Verify Equation (3.2.7) on page 30. (Hint: You will need the two sums

$$\sum_{t=1}^n t = \frac{n(n+1)}{2} \text{ and } \sum_{t=1}^n t^2 = \frac{n(n+1)(2n+1)}{6}.$$

CHAPTER 4

MODELS FOR STATIONARY TIME SERIES

This chapter discusses the basic concepts of a broad class of parametric time series models—the autoregressive moving average (ARMA) models. These models have assumed great importance in modeling real-world processes.

4.1 General Linear Processes

We will always let $\{Y_t\}$ denote the observed time series. From here on we will also let $\{e_t\}$ represent an unobserved white noise series, that is, a sequence of identically distributed, zero-mean, independent random variables. For much of our work, the assumption of independence could be replaced by the weaker assumption that the $\{e_t\}$ are uncorrelated random variables, but we will not pursue that slight generality.

A *general linear process*, $\{Y_t\}$, is one that can be represented as a weighted linear combination of present and past white noise terms as

$$Y_t = e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots \quad (4.1.1)$$

If the right-hand side of this expression is truly an infinite series, then certain conditions must be placed on the ψ -weights for the right-hand side to be meaningful mathematically. For our purposes, it suffices to assume that

$$\sum_{i=1}^{\infty} \psi_i^2 < \infty \quad (4.1.2)$$

We should also note that since $\{e_t\}$ is unobservable, there is no loss in the generality of Equation (4.1.2) if we assume that the coefficient on e_t is 1; effectively, $\psi_0 = 1$.

An important nontrivial example to which we will return often is the case where the ψ 's form an exponentially decaying sequence

$$\psi_j = \phi^j$$

where ϕ is a number strictly between -1 and $+1$. Then

$$Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \dots$$

For this example,

$$E(Y_t) = E(e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \dots) = 0$$

so that $\{Y_t\}$ has a constant mean of zero. Also,

$$\begin{aligned} \text{Var}(Y_t) &= \text{Var}(e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \dots) \\ &= \text{Var}(e_t) + \phi^2 \text{Var}(e_{t-1}) + \phi^4 \text{Var}(e_{t-2}) + \dots \\ &= \sigma_e^2(1 + \phi^2 + \phi^4 + \dots) \\ &= \frac{\sigma_e^2}{1 - \phi^2} \quad (\text{by summing a geometric series}) \end{aligned}$$

Furthermore,

$$\begin{aligned} \text{Cov}(Y_t, Y_{t-1}) &= \text{Cov}(e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \dots, e_{t-1} + \phi e_{t-2} + \phi^2 e_{t-3} + \dots) \\ &= \text{Cov}(\phi e_{t-1}, e_{t-1}) + \text{Cov}(\phi^2 e_{t-2}, \phi e_{t-2}) + \dots \\ &= \phi \sigma_e^2 + \phi^3 \sigma_e^2 + \phi^5 \sigma_e^2 + \dots \\ &= \phi \sigma_e^2(1 + \phi^2 + \phi^4 + \dots) \\ &= \frac{\phi \sigma_e^2}{1 - \phi^2} \quad (\text{again summing a geometric series}) \end{aligned}$$

Thus

$$\text{Corr}(Y_t, Y_{t-1}) = \left[\frac{\phi \sigma_e^2}{1 - \phi^2} \right] / \left[\frac{\sigma_e^2}{1 - \phi^2} \right] = \phi$$

$$\text{In a similar manner, we can find } \text{Cov}(Y_t, Y_{t-k}) = \frac{\phi^k \sigma_e^2}{1 - \phi^2}$$

and thus

$$\text{Corr}(Y_t, Y_{t-k}) = \phi^k \quad (4.1.3)$$

It is important to note that the process defined in this way is stationary—the autocovariance structure depends only on time lag and not on absolute time. For a general linear process, $Y_t = e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots$, calculations similar to those done above yield the following results:

$$E(Y_t) = 0 \quad \gamma_k = \text{Cov}(Y_t, Y_{t-k}) = \sigma_e^2 \sum_{i=0}^{\infty} \psi_i \psi_{i+k} \quad k \geq 0 \quad (4.1.4)$$

with $\psi_0 = 1$. A process with a nonzero mean μ may be obtained by adding μ to the right-hand side of Equation (4.1.1). Since the mean does not affect the covariance properties of a process, we assume a zero mean until we begin fitting models to data.

4.2 Moving Average Processes

In the case where only a finite number of the ψ -weights are nonzero, we have what is called a moving average process. In this case, we change notation[†] somewhat and write

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (4.2.1)$$

We call such a series a **moving average of order q** and abbreviate the name to MA(q). The terminology moving average arises from the fact that Y_t is obtained by applying the weights $1, -\theta_1, -\theta_2, \dots, -\theta_q$ to the variables $e_t, e_{t-1}, e_{t-2}, \dots, e_{t-q}$ and then moving the weights and applying them to $e_{t+1}, e_t, e_{t-1}, \dots, e_{t-q+1}$ to obtain Y_{t+1} and so on. Moving average models were first considered by Slutsky (1927) and Wold (1938).

The First-Order Moving Average Process

We consider in detail the simple but nevertheless important moving average process of order 1, that is, the MA(1) series. Rather than specialize the formulas in Equation (4.1.4), it is instructive to rederive the results. The model is $Y_t = e_t - \theta e_{t-1}$. Since only one θ is involved, we drop the redundant subscript 1. Clearly $E(Y_t) = 0$ and $Var(Y_t) = \sigma_e^2(1 + \theta^2)$. Now

$$\begin{aligned} Cov(Y_t, Y_{t-1}) &= Cov(e_t - \theta e_{t-1}, e_{t-1} - \theta e_{t-2}) \\ &= Cov(-\theta e_{t-1}, e_{t-1}) = -\theta \sigma_e^2 \end{aligned}$$

and

$$\begin{aligned} Cov(Y_t, Y_{t-2}) &= Cov(e_t - \theta e_{t-1}, e_{t-2} - \theta e_{t-3}) \\ &= 0 \end{aligned}$$

since there are no e 's with subscripts in common between Y_t and Y_{t-2} . Similarly, $Cov(Y_t, Y_{t-k}) = 0$ whenever $k \geq 2$; that is, *the process has no correlation beyond lag 1*. This fact will be important later when we need to choose suitable models for real data.

In summary, for an MA(1) model $Y_t = e_t - \theta e_{t-1}$,

$$\left. \begin{array}{l} E(Y_t) = 0 \\ \gamma_0 = Var(Y_t) = \sigma_e^2(1 + \theta^2) \\ \gamma_1 = -\theta \sigma_e^2 \\ \rho_1 = (-\theta)/(1 + \theta^2) \\ \gamma_k = \rho_k = 0 \quad \text{for } k \geq 2 \end{array} \right\} \quad (4.2.2)$$

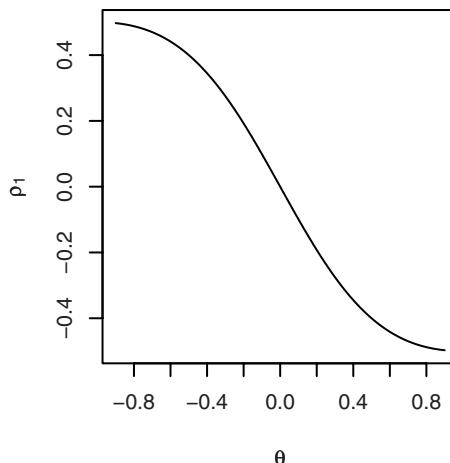
[†] The reason for this change will be evident later on. Some statistical software, for example R, uses plus signs before the thetas. Check with yours to see which convention it uses.

Some numerical values for ρ_1 versus θ in Equation (4.2.2) help illustrate the possibilities. Note that the ρ_1 values for negative θ can be obtained by simply negating the value given for the corresponding positive θ -value.

θ	$\rho_1 = -\theta/(1 + \theta^2)$	θ	$\rho_1 = -\theta/(1 + \theta^2)$
0.1	-0.099	0.6	-0.441
0.2	-0.192	0.7	-0.470
0.3	-0.275	0.8	-0.488
0.4	-0.345	0.9	-0.497
0.5	-0.400	1.0	-0.500

A calculus argument shows that the largest value that ρ_1 can attain is $\rho_1 = \frac{1}{2}$ when $\theta = -1$ and the smallest value is $\rho_1 = -\frac{1}{2}$, which occurs when $\theta = +1$ (see Exercise 4.3). Exhibit 4.1 displays a graph of the lag 1 autocorrelation values for θ ranging from -1 to $+1$.

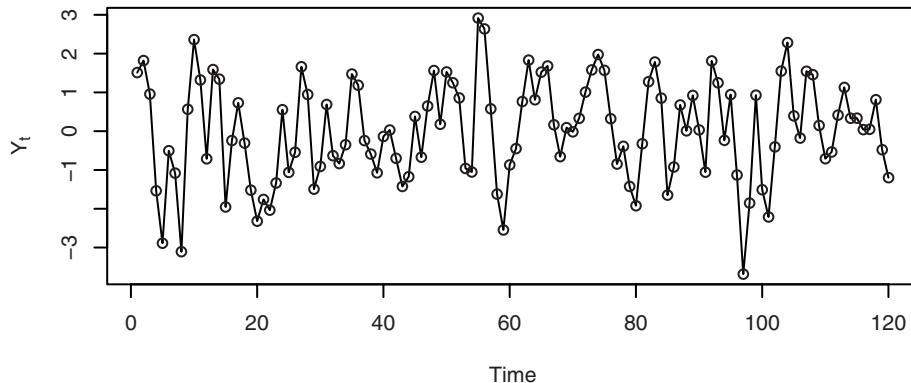
Exhibit 4.1 Lag 1 Autocorrelation of an MA(1) Process for Different θ



Exercise 4.4 asks you to show that when any nonzero value of θ is replaced by $1/\theta$, the *same* value for ρ_1 is obtained. For example, ρ_1 is the same for $\theta = \frac{1}{2}$ as for $\theta = 1/(\frac{1}{2}) = 2$. If we knew that an MA(1) process had $\rho_1 = 0.4$, we still could not tell the precise value of θ . We will return to this troublesome point when we discuss *invertibility* in Section 4.5 on page 79.

Exhibit 4.2 shows a time plot of a simulated MA(1) series with $\theta = -0.9$ and normally distributed white noise. Recall from Exhibit 4.1 that $\rho_1 = 0.4972$ for this model; thus there is moderately strong positive correlation at lag 1. This correlation is evident in the plot of the series since consecutive observations tend to be closely related. If an observation is above the mean level of the series, then the next observation also tends to be above the mean. The plot is relatively smooth over time, with only occasional large fluctuations.

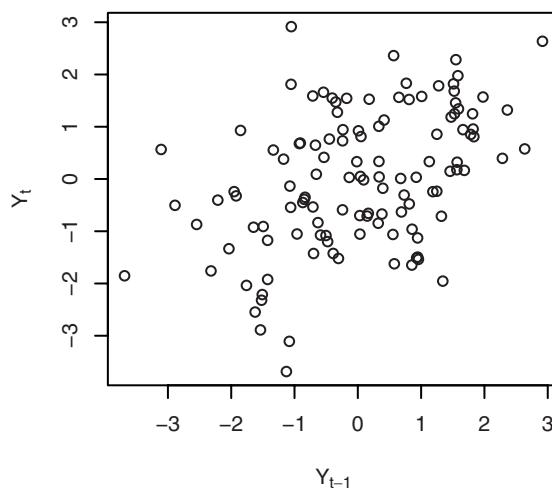
Exhibit 4.2 Time Plot of an MA(1) Process with $\theta = -0.9$



```
> win.graph(width=4.875, height=3, pointsize=8)
> data(ma1.2.s); plot(ma1.2.s, ylab=expression(Y[t]), type='o')
```

The lag 1 autocorrelation is even more apparent in Exhibit 4.3, which plots Y_t versus Y_{t-1} . Note the moderately strong upward trend in this plot.

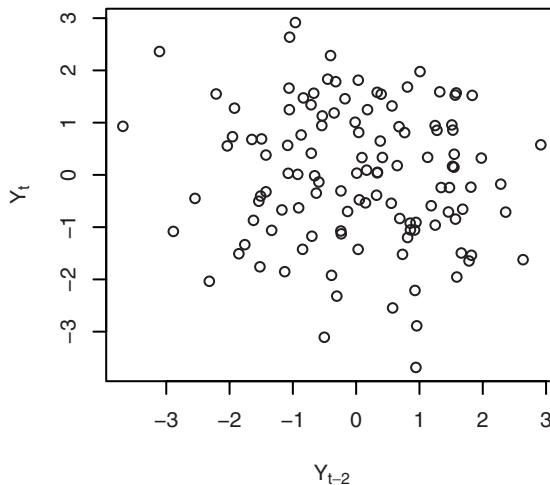
Exhibit 4.3 Plot of Y_t versus Y_{t-1} for MA(1) Series in Exhibit 4.2



```
> win.graph(width=3, height=3, pointsize=8)
> plot(y=ma1.2.s, x=zlag(ma1.2.s), ylab=expression(Y[t]),
       xlab=expression(Y[t-1]), type='p')
```

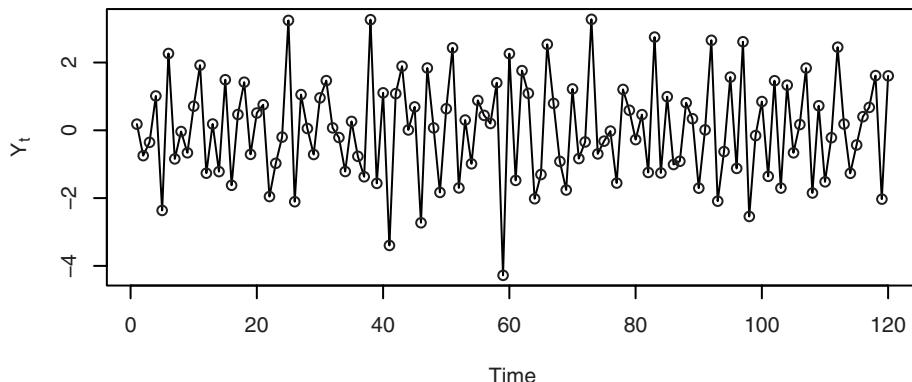
The plot of Y_t versus Y_{t-2} in Exhibit 4.4 gives a strong visualization of the zero autocorrelation at lag 2 for this model.

Exhibit 4.4 Plot of Y_t versus Y_{t-2} for MA(1) Series in Exhibit 4.2



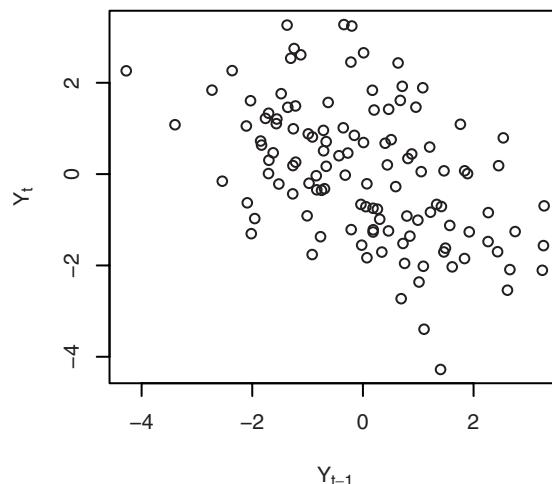
```
> plot(y=ma1.2.s,x=zlag(ma1.2.s,2),ylab=expression(Y[t]),
      xlab=expression(Y[t-2]),type='p')
```

A somewhat different series is shown in Exhibit 4.5. This is a simulated MA(1) series with $\theta = +0.9$. Recall from Exhibit 4.1 that $\rho_1 = -0.497$ for this model; thus there is moderately strong negative correlation at lag 1. This correlation can be seen in the plot of the series since consecutive observations tend to be on opposite sides of the zero mean. If an observation is above the mean level of the series, then the next observation tends to be below the mean. The plot is quite jagged over time—especially when compared with the plot in Exhibit 4.2.

Exhibit 4.5 Time Plot of an MA(1) Process with $\theta = +0.9$ 

```
> win.graph(width=4.875, height=3, pointsize=8)
> data(ma1.1.s)
> plot(ma1.1.s, ylab=expression(Y[t]), type='o')
```

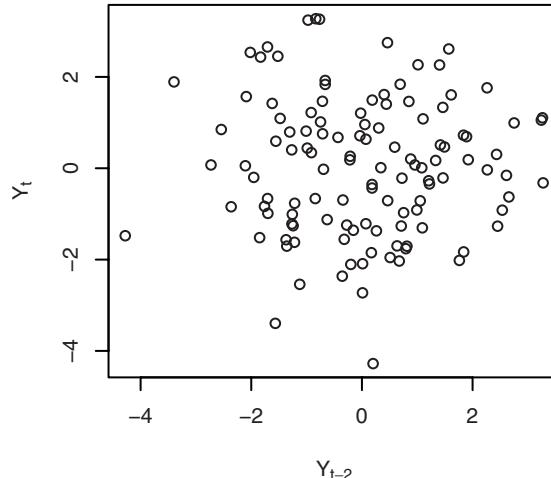
The negative lag 1 autocorrelation is even more apparent in the lag plot of Exhibit 4.6.

Exhibit 4.6 Plot of Y_t versus Y_{t-1} for MA(1) Series in Exhibit 4.5

```
> win.graph(width=3, height=3, pointsize=8)
> plot(y=ma1.1.s, x=zlag(ma1.1.s), ylab=expression(Y[t]),
>       xlab=expression(Y[t-1]), type='p')
```

The plot of Y_t versus Y_{t-2} in Exhibit 4.7 displays the zero autocorrelation at lag 2 for this model.

Exhibit 4.7 Plot of Y_t versus Y_{t-2} for MA(1) Series in Exhibit 4.5



```
> plot(y=ma1.1.s,x=zlag(ma1.1.s,2),ylab=expression(Y[t]),  
xlab=expression(Y[t-2]),type='p')
```

MA(1) processes have no autocorrelation beyond lag 1, but by increasing the order of the process, we can obtain higher-order correlations.

The Second-Order Moving Average Process

Consider the moving average process of order 2:

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}$$

Here

$$\gamma_0 = \text{Var}(Y_t) = \text{Var}(e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}) = (1 + \theta_1^2 + \theta_2^2)\sigma_e^2$$

$$\begin{aligned}\gamma_1 &= \text{Cov}(Y_t, Y_{t-1}) = \text{Cov}(e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}, e_{t-1} - \theta_1 e_{t-2} - \theta_2 e_{t-3}) \\ &= \text{Cov}(-\theta_1 e_{t-1}, e_{t-1}) + \text{Cov}(-\theta_1 e_{t-2}, -\theta_2 e_{t-2}) \\ &= [-\theta_1 + (-\theta_1)(-\theta_2)]\sigma_e^2 \\ &= (-\theta_1 + \theta_1 \theta_2)\sigma_e^2\end{aligned}$$

and

$$\begin{aligned}
 \gamma_2 &= \text{Cov}(Y_t, Y_{t-2}) = \text{Cov}(e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}, e_{t-2} - \theta_1 e_{t-3} - \theta_2 e_{t-4}) \\
 &= \text{Cov}(-\theta_2 e_{t-2}, e_{t-2}) \\
 &= -\theta_2 \sigma_e^2
 \end{aligned}$$

Thus, for an MA(2) process,

$$\begin{aligned}
 \rho_1 &= \frac{-\theta_1 + \theta_1 \theta_2}{1 + \theta_1^2 + \theta_2^2} \\
 \rho_2 &= \frac{-\theta_2}{1 + \theta_1^2 + \theta_2^2} \\
 \rho_k &= 0 \text{ for } k = 3, 4, \dots
 \end{aligned} \tag{4.2.3}$$

For the specific case $Y_t = e_t - e_{t-1} + 0.6e_{t-2}$, we have

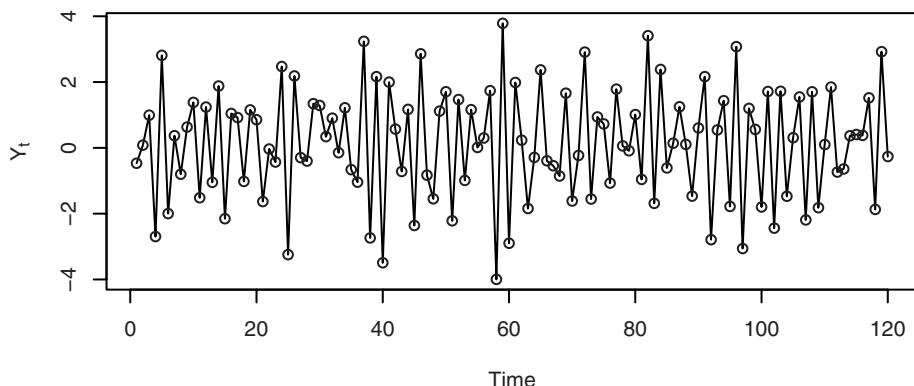
$$\rho_1 = \frac{-1 + (1)(-0.6)}{1 + (1)^2 + (-0.6)^2} = \frac{-1.6}{2.36} = -0.678$$

and

$$\rho_2 = \frac{0.6}{2.36} = 0.254$$

A time plot of a simulation of this MA(2) process is shown in Exhibit 4.8. The series tends to move back and forth across the mean in one time unit. This reflects the fairly strong negative autocorrelation at lag 1.

Exhibit 4.8 Time Plot of an MA(2) Process with $\theta_1 = 1$ and $\theta_2 = -0.6$



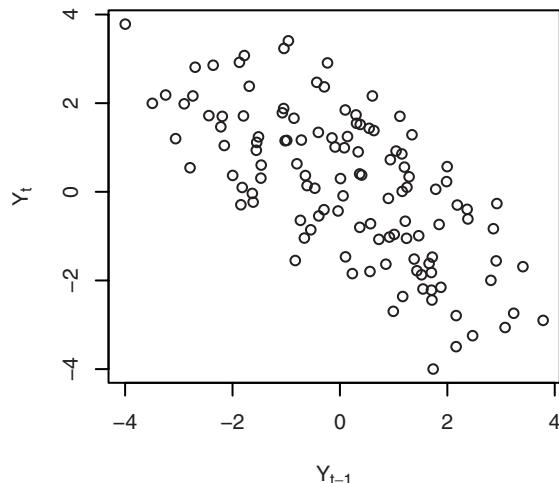
```

> win.graph(width=4.875, height=3, pointsize=8)
> data(ma2.s); plot(ma2.s, ylab=expression(Y[t]), type='o')

```

The plot in Exhibit 4.9 reflects that negative autocorrelation quite dramatically.

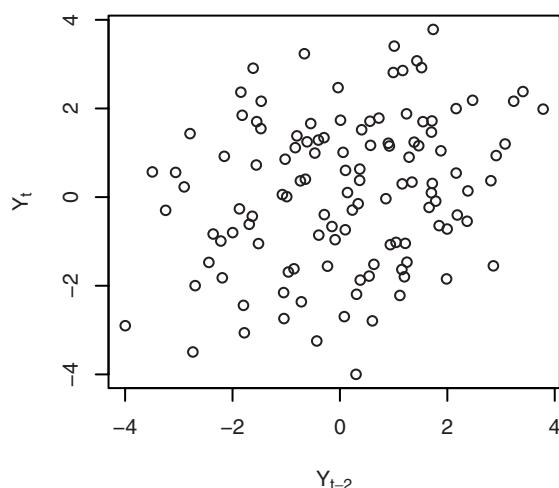
Exhibit 4.9 Plot of Y_t versus Y_{t-1} for MA(2) Series in Exhibit 4.8



```
> win.graph(width=3,height=3,pointsizes=8)
> plot(y=ma2.s,x=zlag(ma2.s),ylab=expression(Y[t]),
       xlab=expression(Y[t-1]),type='p')
```

The weak positive autocorrelation at lag 2 is displayed in Exhibit 4.10.

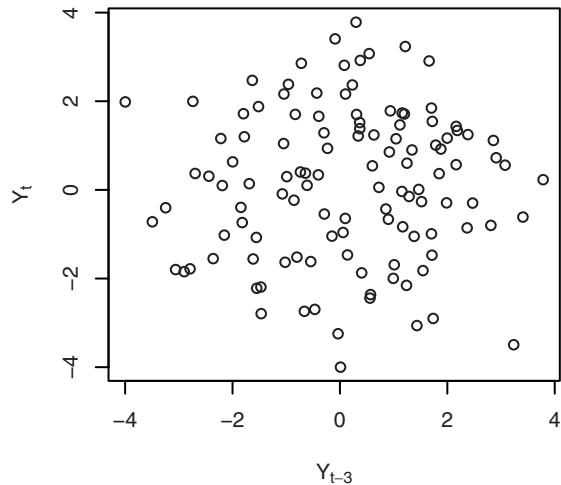
Exhibit 4.10 Plot of Y_t versus Y_{t-2} for MA(2) Series in Exhibit 4.8



```
> plot(y=ma2.s,x=zlag(ma2.s,2),ylab=expression(Y[t]),
       xlab=expression(Y[t-2]),type='p')
```

Finally, the lack of autocorrelation at lag 3 is apparent from the scatterplot in Exhibit 4.11.

Exhibit 4.11 Plot of Y_t versus Y_{t-3} for MA(2) Series in Exhibit 4.8



```
> plot(y=ma2.s,x=zlag(ma2.s,3),ylab=expression(Y[t]),
       xlab=expression(Y[t-3]),type='p')
```

The General MA(q) Process

For the general MA(q) process $Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q}$, similar calculations show that

$$\gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)\sigma_e^2 \quad (4.2.4)$$

and

$$\rho_k = \begin{cases} \frac{-\theta_k + \theta_1\theta_{k+1} + \theta_2\theta_{k+2} + \cdots + \theta_{q-k}\theta_q}{1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2} & \text{for } k = 1, 2, \dots, q \\ 0 & \text{for } k > q \end{cases} \quad (4.2.5)$$

where the numerator of ρ_q is just $-\theta_q$. The autocorrelation function “cuts off” after lag q ; that is, it is zero. Its shape can be almost anything for the earlier lags. Another type of process, the autoregressive process, provides models for alternative autocorrelation patterns.

4.3 Autoregressive Processes

Autoregressive processes are as their name suggests—regressions on themselves. Specifically, a p th-order **autoregressive process** $\{Y_t\}$ satisfies the equation

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t \quad (4.3.1)$$

The current value of the series Y_t is a linear combination of the p most recent past values of itself plus an “innovation” term e_t that incorporates everything new in the series at time t that is not explained by the past values. Thus, for every t , we assume that e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$. Yule (1926) carried out the original work on autoregressive processes.[†]

The First-Order Autoregressive Process

Again, it is instructive to consider the first-order model, abbreviated AR(1), in detail. Assume the series is stationary and satisfies

$$Y_t = \phi Y_{t-1} + e_t \quad (4.3.2)$$

where we have dropped the subscript 1 from the coefficient ϕ for simplicity. As usual, in these initial chapters, we assume that the process mean has been subtracted out so that the series mean is zero. The conditions for stationarity will be considered later.

We first take variances of both sides of Equation (4.3.2) and obtain

$$\gamma_0 = \phi^2 \gamma_0 + \sigma_e^2$$

Solving for γ_0 yields

$$\gamma_0 = \frac{\sigma_e^2}{1 - \phi^2} \quad (4.3.3)$$

Notice the immediate implication that $\phi^2 < 1$ or that $|\phi| < 1$. Now take Equation (4.3.2), multiply both sides by Y_{t-k} ($k = 1, 2, \dots$), and take expected values

$$E(Y_{t-k} Y_t) = \phi E(Y_{t-k} Y_{t-1}) + E(e_t Y_{t-k})$$

or

$$\gamma_k = \phi \gamma_{k-1} + E(e_t Y_{t-k})$$

Since the series is assumed to be stationary with zero mean, and since e_t is independent of Y_{t-k} , we obtain

$$E(e_t Y_{t-k}) = E(e_t) E(Y_{t-k}) = 0$$

and so

[†] Recall that we are assuming that Y_t has zero mean. We can always introduce a nonzero mean by replacing Y_t by $Y_t - \mu$ throughout our equations.

$$\gamma_k = \phi\gamma_{k-1} \quad \text{for } k = 1, 2, 3, \dots \quad (4.3.4)$$

Setting $k = 1$, we get $\gamma_1 = \phi\gamma_0 = \phi\sigma_e^2/(1-\phi^2)$. With $k = 2$, we obtain $\gamma_2 = \phi^2\sigma_e^2/(1-\phi^2)$. Now it is easy to see that in general

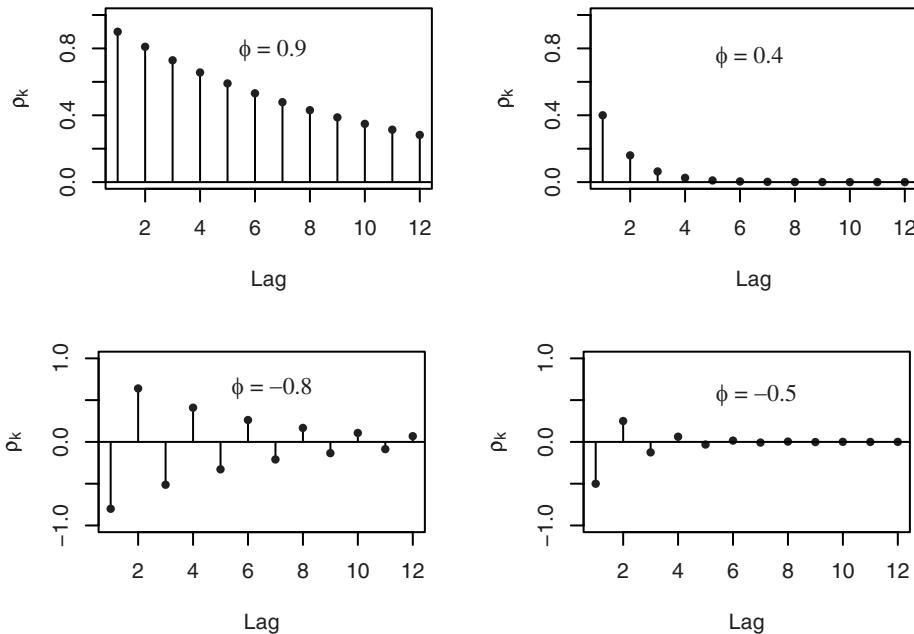
$$\gamma_k = \phi^k \frac{\sigma_e^2}{1-\phi^2} \quad (4.3.5)$$

and thus

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \phi^k \quad \text{for } k = 1, 2, 3, \dots \quad (4.3.6)$$

Since $|\phi| < 1$, the magnitude of the autocorrelation function decreases exponentially as the number of lags, k , increases. If $0 < \phi < 1$, all correlations are positive; if $-1 < \phi < 0$, the lag 1 autocorrelation is negative ($\rho_1 = \phi$) and the signs of successive autocorrelations alternate from positive to negative, with their magnitudes decreasing exponentially. Portions of the graphs of several autocorrelation functions are displayed in Exhibit 4.12.

Exhibit 4.12 Autocorrelation Functions for Several AR(1) Models

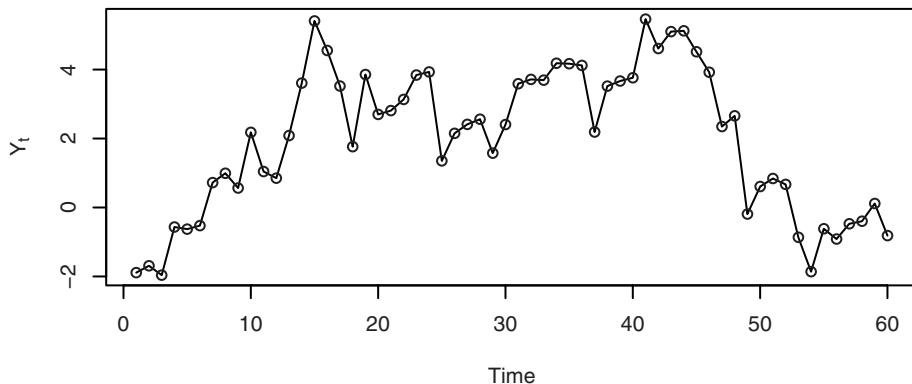


Notice that for ϕ near ± 1 , the exponential decay is quite slow (for example, $(0.9)^6 = 0.53$), but for smaller ϕ , the decay is quite rapid (for example, $(0.4)^6 = 0.00410$). With ϕ near ± 1 , the strong correlation will extend over many lags and produce a relatively

smooth series if ϕ is positive and a very jagged series if ϕ is negative.

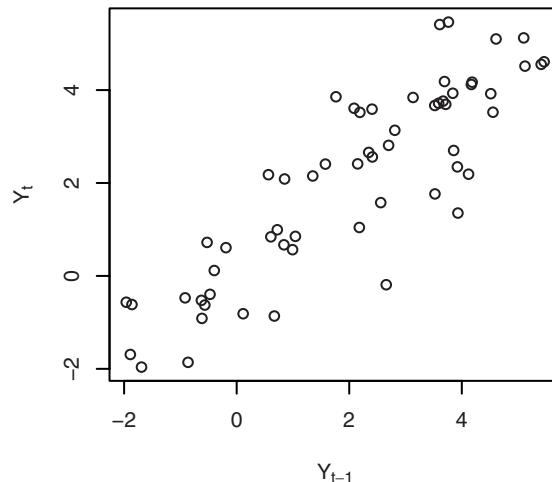
Exhibit 4.13 displays the time plot of a simulated AR(1) process with $\phi = 0.9$. Notice how infrequently the series crosses its theoretical mean of zero. There is a lot of inertia in the series—it hangs together, remaining on the same side of the mean for extended periods. An observer might claim that the series has several trends. We know that in fact the theoretical mean is zero for all time points. The illusion of trends is due to the strong autocorrelation of neighboring values of the series.

Exhibit 4.13 Time Plot of an AR(1) Series with $\phi = 0.9$



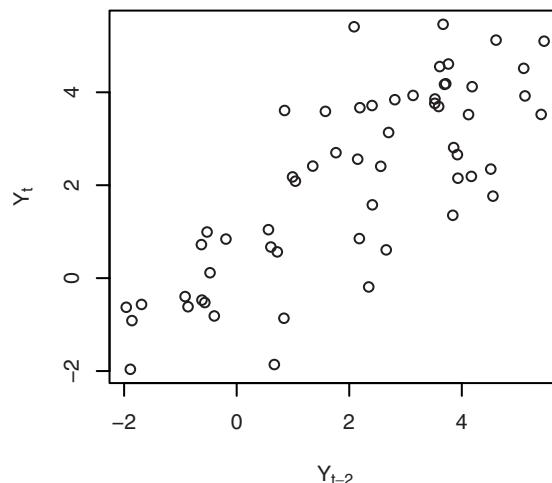
```
> win.graph(width=4.875, height=3, pointsize=8)
> data(ar1.s); plot(ar1.s, ylab=expression(Y[t]), type='o')
```

The smoothness of the series and the strong autocorrelation at lag 1 are depicted in the lag plot shown in Exhibit 4.14.

Exhibit 4.14 Plot of Y_t versus Y_{t-1} for AR(1) Series of Exhibit 4.13

```
> win.graph(width=3, height=3, pointsize=8)
> plot(y=ar1.s, x=zlag(ar1.s), ylab=expression(Y[t]),
       xlab=expression(Y[t-1]), type='p')
```

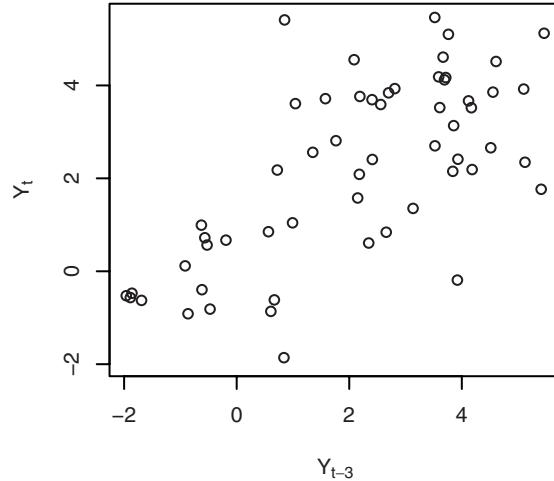
This AR(1) model also has strong positive autocorrelation at lag 2, namely $\rho_2 = (0.9)^2 = 0.81$. Exhibit 4.15 shows this quite well.

Exhibit 4.15 Plot of Y_t versus Y_{t-2} for AR(1) Series of Exhibit 4.13

```
> plot(y=ar1.s, x=zlag(ar1.s, 2), ylab=expression(Y[t]),
       xlab=expression(Y[t-2]), type='p')
```

Finally, at lag 3, the autocorrelation is still quite high: $\rho_3 = (0.9)^3 = 0.729$. Exhibit 4.16 confirms this for this particular series.

Exhibit 4.16 Plot of Y_t versus Y_{t-3} for AR(1) Series of Exhibit 4.13



```
> plot(y=ar1.s,x=zlag(ar1.s,3),ylab=expression(Y[t]),
       xlab=expression(Y[t-3]),type='p')
```

The General Linear Process Version of the AR(1) Model

The recursive definition of the AR(1) process given in Equation (4.3.2) is extremely useful for interpreting the model. For other purposes, it is convenient to express the AR(1) model as a general linear process as in Equation (4.1.1). The recursive definition is valid for all t . If we use this equation with t replaced by $t-1$, we get $Y_{t-1} = \phi Y_{t-2} + e_{t-1}$. Substituting this into the original expression gives

$$\begin{aligned} Y_t &= \phi(\phi Y_{t-2} + e_{t-1}) + e_t \\ &= e_t + \phi e_{t-1} + \phi^2 Y_{t-2} \end{aligned}$$

If we repeat this substitution into the past, say $k-1$ times, we get

$$Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \cdots + \phi^{k-1} e_{t-k+1} + \phi^k Y_{t-k} \quad (4.3.7)$$

Assuming $|\phi| < 1$ and letting k increase without bound, it seems reasonable (this is almost a rigorous proof) that we should obtain the infinite series representation

$$Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \phi^3 e_{t-3} + \cdots \quad (4.3.8)$$

This is in the form of the general linear process of Equation (4.1.1) with $\psi_j = \phi^j$, which we already investigated in Section 4.1 on page 55. Note that this representation reemphasizes the need for the restriction $|\phi| < 1$.

Stationarity of an AR(1) Process

It can be shown that, subject to the restriction that e_t be independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$ and that $\sigma_e^2 > 0$, the solution of the AR(1) defining recursion $Y_t = \phi Y_{t-1} + e_t$ will be stationary if and only if $|\phi| < 1$. The requirement $|\phi| < 1$ is usually called the **stationarity condition** for the AR(1) process (See Box, Jenkins, and Reinsel, 1994, p. 54; Nelson, 1973, p. 39; and Wei, 2005, p. 32) even though more than stationarity is involved. See especially Exercises 4.16, 4.18, and 4.25.

At this point, we should note that the autocorrelation function for the AR(1) process has been derived in two different ways. The first method used the general linear process representation leading up to Equation (4.1.3). The second method used the defining recursion $Y_t = \phi Y_{t-1} + e_t$ and the development of Equations (4.3.4), (4.3.5), and (4.3.6). A third derivation is obtained by multiplying both sides of Equation (4.3.7) by Y_{t-k} , taking expected values of both sides, and using the fact that $e_t, e_{t-1}, e_{t-2}, \dots, e_{t-(k-1)}$ are independent of Y_{t-k} . The second method should be especially noted since it will generalize nicely to higher-order processes.

The Second-Order Autoregressive Process

Now consider the series satisfying

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t \quad (4.3.9)$$

where, as usual, we assume that e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$. To discuss stationarity, we introduce the **AR characteristic polynomial**

$$\phi(x) = 1 - \phi_1 x - \phi_2 x^2$$

and the corresponding **AR characteristic equation**

$$1 - \phi_1 x - \phi_2 x^2 = 0$$

We recall that a quadratic equation always has two roots (possibly complex).

Stationarity of the AR(2) Process

It may be shown that, subject to the condition that e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$, a stationary solution to Equation (4.3.9) exists if and only if the roots of the AR characteristic equation exceed 1 in absolute value (modulus). We sometimes say that the roots should lie outside the unit circle in the complex plane. This statement will generalize to the p th-order case without change.[†]

[†] It also applies in the first-order case, where the AR characteristic equation is just $1 - \phi x = 0$ with root $1/\phi$, which exceeds 1 in absolute value if and only if $|\phi| < 1$.

In the second-order case, the roots of the quadratic characteristic equation are easily found to be

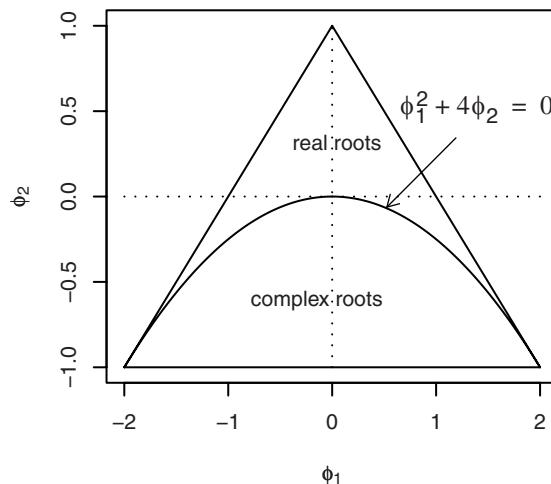
$$\frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \quad (4.3.10)$$

For stationarity, we require that these roots exceed 1 in absolute value. In Appendix B, page 84, we show that this will be true if and only if three conditions are satisfied:

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1 \quad (4.3.11)$$

As with the AR(1) model, we call these the **stationarity conditions** for the AR(2) model. This stationarity region is displayed in Exhibit 4.17.

Exhibit 4.17 Stationarity Parameter Region for AR(2) Process



The Autocorrelation Function for the AR(2) Process

To derive the autocorrelation function for the AR(2) case, we take the defining recursive relationship of Equation (4.3.9), multiply both sides by Y_{t-k} , and take expectations. Assuming stationarity, zero means, and that e_t is independent of Y_{t-k} , we get

$$\gamma_k = \phi_1 \gamma_{k-1} + \phi_2 \gamma_{k-2} \quad \text{for } k = 1, 2, 3, \dots \quad (4.3.12)$$

or, dividing through by γ_0 ,

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} \quad \text{for } k = 1, 2, 3, \dots \quad (4.3.13)$$

Equations (4.3.12) and/or (4.3.13) are usually called the **Yule-Walker equations**, especially the set of two equations obtained for $k = 1$ and 2. Setting $k = 1$ and using $\rho_0 = 1$ and $\rho_{-1} = \rho_1$, we get $\rho_1 = \phi_1 + \phi_2 \rho_1$ and so

$$\rho_1 = \frac{\phi_1}{1 - \phi_2} \quad (4.3.14)$$

Using the now known values for ρ_1 (and ρ_0), Equation (4.3.13) can be used with $k = 2$ to obtain

$$\begin{aligned} \rho_2 &= \phi_1 \rho_1 + \phi_2 \rho_0 \\ &= \frac{\phi_2(1 - \phi_2) + \phi_1^2}{1 - \phi_2} \end{aligned} \quad (4.3.15)$$

Successive values of ρ_k may be easily calculated numerically from the recursive relationship of Equation (4.3.13).

Although Equation (4.3.13) is very efficient for calculating autocorrelation values numerically from given values of ϕ_1 and ϕ_2 , for other purposes it is desirable to have a more explicit formula for ρ_k . The form of the explicit solution depends critically on the roots of the characteristic equation $1 - \phi_1 x - \phi_2 x^2 = 0$. Denoting the reciprocals of these roots by G_1 and G_2 , it is shown in Appendix B, page 84, that

$$G_1 = \frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{2} \quad \text{and} \quad G_2 = \frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{2}$$

For the case $G_1 \neq G_2$, it can be shown that we have

$$\rho_k = \frac{(1 - G_2^2)G_1^{k+1} - (1 - G_1^2)G_2^{k+1}}{(G_1 - G_2)(1 + G_1 G_2)} \quad \text{for } k \geq 0 \quad (4.3.16)$$

If the roots are complex (that is, if $\phi_1^2 + 4\phi_2 < 0$), then ρ_k may be rewritten as

$$\rho_k = R^k \frac{\sin(\Theta k + \Phi)}{\sin(\Phi)} \quad \text{for } k \geq 0 \quad (4.3.17)$$

where $R = \sqrt{-\phi_2}$ and Θ and Φ are defined by $\cos(\Theta) = \phi_1 / (2\sqrt{-\phi_2})$ and $\tan(\Phi) = [(1 - \phi_2) / (1 + \phi_2)]$.

For completeness, we note that if the roots are equal ($\phi_1^2 + 4\phi_2 = 0$), then we have

$$\rho_k = \left(1 + \frac{1 + \phi_2}{1 - \phi_2} k\right) \left(\frac{\phi_1}{2}\right)^k \quad \text{for } k = 0, 1, 2, \dots \quad (4.3.18)$$

A good discussion of the derivations of these formulas can be found in Fuller (1996, Section 2.5).

The specific details of these formulas are of little importance to us. We need only note that the autocorrelation function can assume a wide variety of shapes. In all cases, the magnitude of ρ_k dies out exponentially fast as the lag k increases. In the case of complex roots, ρ_k displays a damped sine wave behavior with **damping factor** R , $0 \leq R < 1$, **frequency** Θ , and **phase** Φ . Illustrations of the possible shapes are given in Exhibit 4.18. (The R function ARMAacf discussed on page 450 is useful for plotting.)

Exhibit 4.18 Autocorrelation Functions for Several AR(2) Models

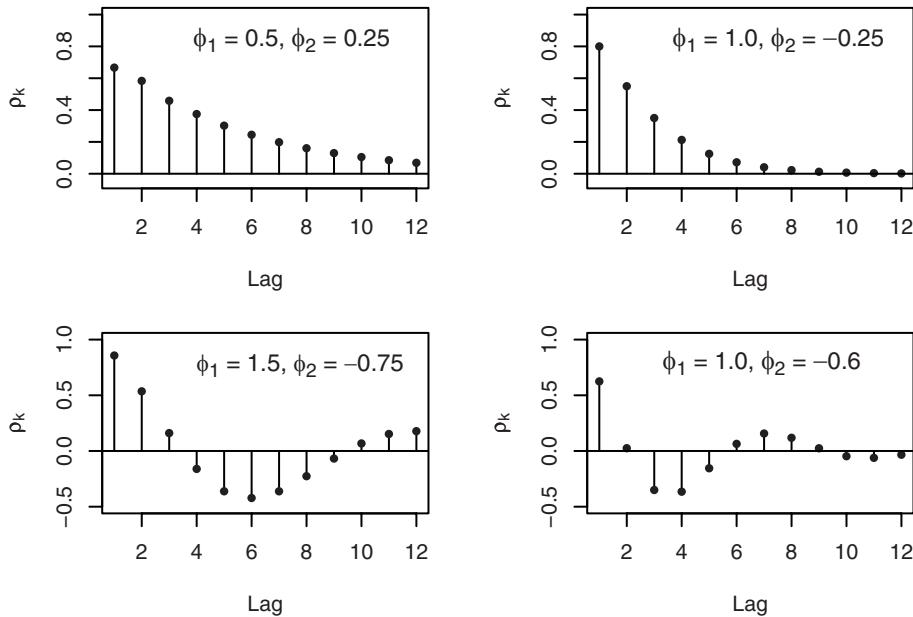
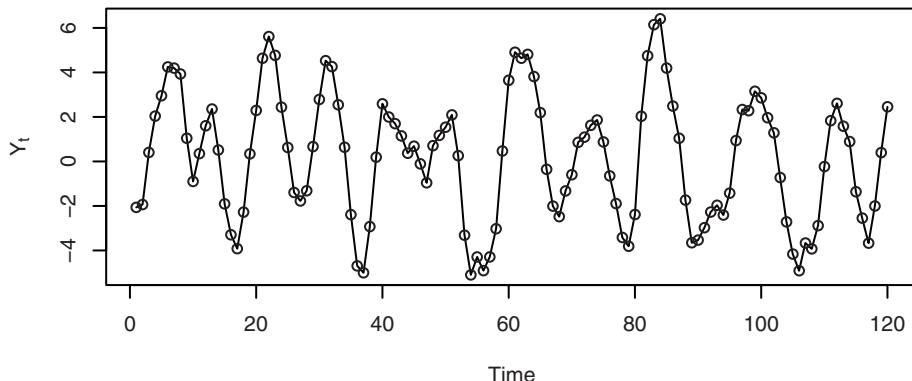


Exhibit 4.19 displays the time plot of a simulated AR(2) series with $\phi_1 = 1.5$ and $\phi_2 = -0.75$. The periodic behavior of ρ_k shown in Exhibit 4.18 is clearly reflected in the nearly periodic behavior of the series with the same period of $360/30 = 12$ time units. If Θ is measured in radians, $2\pi/\Theta$ is sometimes called the **quasi-period** of the AR(2) process.

Exhibit 4.19 Time Plot of an AR(2) Series with $\phi_1 = 1.5$ and $\phi_2 = -0.75$



```
> win.graph(width=4.875, height=3, pointsize=8)
> data(ar2.s); plot(ar2.s, ylab=expression(Y[t]), type='o')
```

The Variance for the AR(2) Model

The process variance γ_0 can be expressed in terms of the model parameters ϕ_1 , ϕ_2 , and σ_e^2 as follows: Taking the variance of both sides of Equation (4.3.9) yields

$$\gamma_0 = (\phi_1^2 + \phi_2^2)\gamma_0 + 2\phi_1\phi_2\gamma_1 + \sigma_e^2 \quad (4.3.19)$$

Setting $k = 1$ in Equation (4.3.12) gives a second linear equation for γ_0 and γ_1 , $\gamma_1 = \phi_1\gamma_0 + \phi_2\gamma_1$, which can be solved simultaneously with Equation (4.3.19) to obtain

$$\begin{aligned} \gamma_0 &= \frac{(1 - \phi_2)\sigma_e^2}{(1 - \phi_2)(1 - \phi_1^2 - \phi_2^2) - 2\phi_2\phi_1^2} \\ &= \left(\frac{1 - \phi_2}{1 + \phi_2}\right) \frac{\sigma_e^2}{(1 - \phi_2)^2 - \phi_1^2} \end{aligned} \quad (4.3.20)$$

The ψ -Coefficients for the AR(2) Model

The ψ -coefficients in the general linear process representation for an AR(2) series are more complex than for the AR(1) case. However, we can substitute the general linear process representation using Equation (4.1.1) for Y_t , for Y_{t-1} , and for Y_{t-2} into $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$. If we then equate coefficients of e_j , we get the recursive relationships

$$\left. \begin{aligned} \psi_0 &= 1 \\ \psi_1 - \phi_1\psi_0 &= 0 \\ \psi_j - \phi_1\psi_{j-1} - \phi_2\psi_{j-2} &= 0 \quad \text{for } j = 2, 3, \dots \end{aligned} \right\} \quad (4.3.21)$$

These may be solved recursively to obtain $\psi_0 = 1$, $\psi_1 = \phi_1$, $\psi_2 = \phi_1^2 + \phi_2$, and so on. These relationships provide excellent numerical solutions for the ψ -coefficients for given numerical values of ϕ_1 and ϕ_2 .

One can also show that, for $G_1 \neq G_2$, an explicit solution is

$$\psi_j = \frac{G_1^{j+1} - G_2^{j+1}}{G_1 - G_2} \quad (4.3.22)$$

where, as before, G_1 and G_2 are the reciprocals of the roots of the AR characteristic equation. If the roots are complex, Equation (4.3.22) may be rewritten as

$$\psi_j = R^j \left\{ \frac{\sin[(j+1)\Theta]}{\sin(\Theta)} \right\} \quad (4.3.23)$$

a damped sine wave with the same damping factor R and frequency Θ as in Equation (4.3.17) for the autocorrelation function.

For completeness, we note that if the roots are equal, then

$$\psi_j = (1+j)\phi_1^j \quad (4.3.24)$$

The General Autoregressive Process

Consider now the p th-order autoregressive model

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t \quad (4.3.25)$$

with AR characteristic polynomial

$$\phi(x) = 1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p \quad (4.3.26)$$

and corresponding AR characteristic equation

$$1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p = 0 \quad (4.3.27)$$

As noted earlier, assuming that e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$ a stationary solution to Equation (4.3.27) exists if and only if the p roots of the AR characteristic equation each exceed 1 in absolute value (modulus). Other relationships between polynomial roots and coefficients may be used to show that the following two inequalities are necessary for stationarity. That is, for the roots to be greater than 1 in modulus, it is necessary, but not sufficient, that both

$$\left. \begin{array}{l} \phi_1 + \phi_2 + \cdots + \phi_p < 1 \\ \text{and} \quad |\phi_p| < 1 \end{array} \right\} \quad (4.3.28)$$

Assuming stationarity and zero means, we may multiply Equation (4.3.25) by Y_{t-k} , take expectations, divide by γ_0 , and obtain the important recursive relationship

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \phi_3 \rho_{k-3} + \cdots + \phi_p \rho_{k-p} \quad \text{for } k \geq 1 \quad (4.3.29)$$

Putting $k = 1, 2, \dots$, and p into Equation (4.3.29) and using $\rho_0 = 1$ and $\rho_{-k} = \rho_k$, we get the general **Yule-Walker equations**

$$\left. \begin{array}{l} \rho_1 = \phi_1 + \phi_2 \rho_1 + \phi_3 \rho_2 + \cdots + \phi_p \rho_{p-1} \\ \rho_2 = \phi_1 \rho_1 + \phi_2 + \phi_3 \rho_1 + \cdots + \phi_p \rho_{p-2} \\ \vdots \\ \rho_p = \phi_1 \rho_{p-1} + \phi_2 \rho_{p-2} + \phi_3 \rho_{p-3} + \cdots + \phi_p \end{array} \right\} \quad (4.3.30)$$

Given numerical values for $\phi_1, \phi_2, \dots, \phi_p$, these linear equations can be solved to obtain numerical values for $\rho_1, \rho_2, \dots, \rho_p$. Then Equation (4.3.29) can be used to obtain numerical values for ρ_k at any number of higher lags.

Noting that

$$E(e_t Y_t) = E[e_t(\phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t)] = E(e_t^2) = \sigma_e^2$$

we may multiply Equation (4.3.25) by Y_t , take expectations, and find

$$\gamma_0 = \phi_1 \gamma_1 + \phi_2 \gamma_2 + \cdots + \phi_p \gamma_p + \sigma_e^2$$

which, using $\rho_k = \gamma_k/\gamma_0$, can be written as

$$\gamma_0 = \frac{\sigma_e^2}{1 - \phi_1\rho_1 - \phi_2\rho_2 - \cdots - \phi_p\rho_p} \quad (4.3.31)$$

and express the process variance γ_0 in terms of the parameters $\sigma_e^2, \phi_1, \phi_2, \dots, \phi_p$, and the now known values of $\rho_1, \rho_2, \dots, \rho_p$. Of course, explicit solutions for ρ_k are essentially impossible in this generality, but we can say that ρ_k will be a linear combination of exponentially decaying terms (corresponding to the real roots of the characteristic equation) and damped sine wave terms (corresponding to the complex roots of the characteristic equation).

Assuming stationarity, the process can also be expressed in the general linear process form of Equation (4.1.1), but the ψ -coefficients are complicated functions of the parameters $\phi_1, \phi_2, \dots, \phi_p$. The coefficients can be found numerically; see Appendix C on page 85.

4.4 The Mixed Autoregressive Moving Average Model

If we assume that the series is partly autoregressive and partly moving average, we obtain a quite general time series model. In general, if

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (4.4.1)$$

we say that $\{Y_t\}$ is a mixed **autoregressive moving average** process of orders p and q , respectively; we abbreviate the name to ARMA(p, q). As usual, we discuss an important special case first.[†]

The ARMA(1,1) Model

The defining equation can be written

$$Y_t = \phi Y_{t-1} + e_t - \theta e_{t-1} \quad (4.4.2)$$

To derive Yule-Walker type equations, we first note that

$$\begin{aligned} E(e_t Y_t) &= E[e_t(\phi Y_{t-1} + e_t - \theta e_{t-1})] \\ &= \sigma_e^2 \end{aligned}$$

and

[†] In mixed models, we assume that there are no common factors in the autoregressive and moving average polynomials. If there were, we could cancel them and the model would reduce to an ARMA model of lower order. For ARMA(1,1), this means $\theta \neq \phi$.

$$\begin{aligned}
E(e_{t-1}Y_t) &= E[e_{t-1}(\phi Y_{t-1} + e_t - \theta e_{t-1})] \\
&= \phi\sigma_e^2 - \theta\sigma_e^2 \\
&= (\phi - \theta)\sigma_e^2
\end{aligned}$$

If we multiply Equation (4.4.2) by Y_{t-k} and take expectations, we have

$$\left. \begin{aligned}
\gamma_0 &= \phi\gamma_1 + [1 - \theta(\phi - \theta)]\sigma_e^2 \\
\gamma_1 &= \phi\gamma_0 - \theta\sigma_e^2 \\
\gamma_k &= \phi\gamma_{k-1} \quad \text{for } k \geq 2
\end{aligned} \right\} \quad (4.4.3)$$

Solving the first two equations yields

$$\gamma_0 = \frac{(1 - 2\phi\theta + \theta^2)}{1 - \phi^2}\sigma_e^2 \quad (4.4.4)$$

and solving the simple recursion gives

$$\rho_k = \frac{(1 - \theta\phi)(\phi - \theta)}{1 - 2\phi\theta + \theta^2}\phi^{k-1} \quad \text{for } k \geq 1 \quad (4.4.5)$$

Note that this autocorrelation function decays exponentially as the lag k increases. The damping factor is ϕ , but the decay starts from initial value ρ_1 , which also depends on θ . This is in contrast to the AR(1) autocorrelation, which also decays with damping factor ϕ but always from initial value $\rho_0 = 1$. For example, if $\phi = 0.8$ and $\theta = 0.4$, then $\rho_1 = 0.523$, $\rho_2 = 0.418$, $\rho_3 = 0.335$, and so on. Several shapes for ρ_k are possible, depending on the sign of ρ_1 and the sign of ϕ .

The general linear process form of the model can be obtained in the same manner that led to Equation (4.3.8). We find

$$Y_t = e_t + (\phi - \theta) \sum_{j=1}^{\infty} \phi^{j-1} e_{t-j}, \quad (4.4.6)$$

that is,

$$\psi_j = (\phi - \theta)\phi^{j-1} \quad \text{for } j \geq 1$$

We should now mention the obvious stationarity condition $|\phi| < 1$, or equivalently the root of the AR characteristic equation $1 - \phi x = 0$ must exceed unity in absolute value.

For the general ARMA(p, q) model, we state the following facts without proof: Subject to the condition that e_t is independent of $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$, a stationary solution to Equation (4.4.1) exists if and only if all the roots of the AR characteristic equation $\phi(x) = 0$ exceed unity in modulus.

If the stationarity conditions are satisfied, then the model can also be written as a general linear process with ψ -coefficients determined from

$$\left. \begin{aligned} \psi_0 &= 1 \\ \psi_1 &= -\theta_1 + \phi_1 \\ \psi_2 &= -\theta_2 + \phi_2 + \phi_1 \psi_1 \\ &\vdots \\ \psi_j &= -\theta_j + \phi_p \psi_{j-p} + \phi_{p-1} \psi_{j-p+1} + \cdots + \phi_1 \psi_{j-1} \end{aligned} \right\} \quad (4.4.7)$$

where we take $\psi_j = 0$ for $j < 0$ and $\theta_j = 0$ for $j > q$.

Again assuming stationarity, the autocorrelation function can easily be shown to satisfy

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \cdots + \phi_p \rho_{k-p} \quad \text{for } k > q \quad (4.4.8)$$

Similar equations can be developed for $k = 1, 2, 3, \dots, q$ that involve $\theta_1, \theta_2, \dots, \theta_q$. An algorithm suitable for numerical computation of the complete autocorrelation function is given in Appendix C on page 85. (This algorithm is implemented in the R function named ARMAacf.)

4.5 Invertibility

We have seen that for the MA(1) process we get exactly the same autocorrelation function if θ is replaced by $1/\theta$. In the exercises, we find a similar problem with nonuniqueness for the MA(2) model. This lack of uniqueness of MA models, given their autocorrelation functions, must be addressed before we try to infer the values of parameters from observed time series. It turns out that this nonuniqueness is related to the seemingly unrelated question stated next.

An autoregressive process can always be reexpressed as a general linear process through the ψ -coefficients so that an AR process may also be thought of as an infinite-order moving average process. However, for some purposes, the autoregressive representations are also convenient. Can a moving average model be reexpressed as an autoregression?

To fix ideas, consider an MA(1) model:

$$Y_t = e_t - \theta e_{t-1} \quad (4.5.1)$$

First rewriting this as $e_t = Y_t + \theta e_{t-1}$ and then replacing t by $t-1$ and substituting for e_{t-1} above, we get

$$\begin{aligned} e_t &= Y_t + \theta(Y_{t-1} + \theta e_{t-2}) \\ &= Y_t + \theta Y_{t-1} + \theta^2 e_{t-2} \end{aligned}$$

If $|\theta| < 1$, we may continue this substitution “infinitely” into the past and obtain the expression [compare with Equations (4.3.7) and (4.3.8)]

$$e_t = Y_t + \theta Y_{t-1} + \theta^2 Y_{t-2} + \cdots$$

or

$$Y_t = (-\theta Y_{t-1} - \theta^2 Y_{t-2} - \theta^3 Y_{t-3} - \dots) + e_t \quad (4.5.2)$$

If $|\theta| < 1$, we see that the MA(1) model can be inverted into an infinite-order autoregressive model. We say that the MA(1) model is invertible if and only if $|\theta| < 1$.

For a general MA(q) or ARMA(p, q) model, we define the **MA characteristic polynomial** as

$$\theta(x) = 1 - \theta_1 x - \theta_2 x^2 - \theta_3 x^3 - \dots - \theta_q x^q \quad (4.5.3)$$

and the corresponding **MA characteristic equation**

$$1 - \theta_1 x - \theta_2 x^2 - \theta_3 x^3 - \dots - \theta_q x^q = 0 \quad (4.5.4)$$

It can be shown that the MA(q) model is **invertible**; that is, there are coefficients π_j such that

$$Y_t = \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \pi_3 Y_{t-3} + \dots + e_t \quad (4.5.5)$$

if and only if the roots of the MA characteristic equation exceed 1 in modulus. (Compare this with stationarity of an AR model.)

It may also be shown that there is only one set of parameter values that yield an invertible MA process with a given autocorrelation function. For example, $Y_t = e_t + 2e_{t-1}$ and $Y_t = e_t + \frac{1}{2}e_{t-1}$ both have the same autocorrelation function, but only the second one with root -2 is invertible. From here on, we will restrict our attention to the physically sensible class of invertible models.

For a general ARMA(p, q) model, we require both stationarity and invertibility.

4.6 Summary

This chapter introduces the simple but very useful autoregressive, moving average (ARMA) time series models. The basic statistical properties of these models were derived in particular for the important special cases of moving averages of orders 1 and 2 and autoregressive processes of orders 1 and 2. Stationarity and invertibility issues have been pursued for these cases. Properties of mixed ARMA models have also been investigated. You should be well-versed in the autocorrelation properties of these models and the various representations of the models.

EXERCISES

- 4.1** Use first principles to find the autocorrelation function for the stationary process defined by

$$Y_t = 5 + e_t - \frac{1}{2}e_{t-1} + \frac{1}{4}e_{t-2}$$

- 4.2** Sketch the autocorrelation functions for the following MA(2) models with parameters as specified:

- (a) $\theta_1 = 0.5$ and $\theta_2 = 0.4$.
- (b) $\theta_1 = 1.2$ and $\theta_2 = -0.7$.
- (c) $\theta_1 = -1$ and $\theta_2 = -0.6$.

- 4.3** Verify that for an MA(1) process

$$\max_{-\infty < \theta < \infty} \rho_1 = 0.5 \quad \text{and} \quad \min_{-\infty < \theta < \infty} \rho_1 = -0.5$$

- 4.4** Show that when θ is replaced by $1/\theta$, the autocorrelation function for an MA(1) process does not change.

- 4.5** Calculate and sketch the autocorrelation functions for each of the following AR(1) models. Plot for sufficient lags that the autocorrelation function has nearly died out.

- (a) $\phi_1 = 0.6$.
- (b) $\phi_1 = -0.6$.
- (c) $\phi_1 = 0.95$. (Do out to 20 lags.)
- (d) $\phi_1 = 0.3$.

- 4.6** Suppose that $\{Y_t\}$ is an AR(1) process with $-1 < \phi < +1$.

- (a) Find the autocovariance function for $W_t = \nabla Y_t = Y_t - Y_{t-1}$ in terms of ϕ and σ_e^2 .

- (b) In particular, show that $Var(W_t) = 2\sigma_e^2/(1+\phi)$.

- 4.7** Describe the important characteristics of the autocorrelation function for the following models: (a) MA(1), (b) MA(2), (c) AR(1), (d) AR(2), and (e) ARMA(1,1).

- 4.8** Let $\{Y_t\}$ be an AR(2) process of the special form $Y_t = \phi_2 Y_{t-2} + e_t$. Use first principles to find the range of values of ϕ_2 for which the process is stationary.

- 4.9** Use the recursive formula of Equation (4.3.13) to calculate and then sketch the autocorrelation functions for the following AR(2) models with parameters as specified. In each case, specify whether the roots of the characteristic equation are real or complex. If the roots are complex, find the damping factor, R , and frequency, Θ , for the corresponding autocorrelation function when expressed as in Equation (4.3.17), on page 73.

- (a) $\phi_1 = 0.6$ and $\phi_2 = 0.3$.
- (b) $\phi_1 = -0.4$ and $\phi_2 = 0.5$.
- (c) $\phi_1 = 1.2$ and $\phi_2 = -0.7$.
- (d) $\phi_1 = -1$ and $\phi_2 = -0.6$.
- (e) $\phi_1 = 0.5$ and $\phi_2 = -0.9$.
- (f) $\phi_1 = -0.5$ and $\phi_2 = -0.6$.

- 4.10** Sketch the autocorrelation functions for each of the following ARMA models:
- ARMA(1,1) with $\phi = 0.7$ and $\theta = 0.4$.
 - ARMA(1,1) with $\phi = 0.7$ and $\theta = -0.4$.
- 4.11** For the ARMA(1,2) model $Y_t = 0.8Y_{t-1} + e_t + 0.7e_{t-1} + 0.6e_{t-2}$, show that
- $\rho_k = 0.8\rho_{k-1}$ for $k > 2$.
 - $\rho_2 = 0.8\rho_1 + 0.6\sigma_e^2/\gamma_0$.
- 4.12** Consider two MA(2) processes, one with $\theta_1 = \theta_2 = 1/6$ and another with $\theta_1 = -1$ and $\theta_2 = 6$.
- Show that these processes have the same autocorrelation function.
 - How do the roots of the corresponding characteristic polynomials compare?
- 4.13** Let $\{Y_t\}$ be a stationary process with $\rho_k = 0$ for $k > 1$. Show that we must have $|\rho_1| \leq 1/2$. (Hint: Consider $\text{Var}(Y_{n+1} + Y_n + \dots + Y_1)$ and then $\text{Var}(Y_{n+1} - Y_n + Y_{n-1} - \dots \pm Y_1)$. Use the fact that both of these must be nonnegative for all n .)
- 4.14** Suppose that $\{Y_t\}$ is a zero mean, stationary process with $|\rho_1| < 0.5$ and $\rho_k = 0$ for $k > 1$. Show that $\{Y_t\}$ must be representable as an MA(1) process. That is, show that there is a white noise sequence $\{e_t\}$ such that $Y_t = e_t - \theta e_{t-1}$, where ρ_1 is correct and e_t is uncorrelated with Y_{t-k} for $k > 0$. (Hint: Choose θ such that $|\theta| < 1$ and $\rho_1 = -\theta/(1 + \theta^2)$; then let $e_t = \sum_{j=0}^{\infty} \theta^j Y_{t-j}$. If we assume that $\{Y_t\}$ is a normal process, e_t will also be normal, and zero correlation is equivalent to independence.)
- 4.15** Consider the AR(1) model $Y_t = \phi Y_{t-1} + e_t$. Show that if $|\phi| = 1$ the process cannot be stationary. (Hint: Take variances of both sides.)
- 4.16** Consider the “nonstationary” AR(1) model $Y_t = 3Y_{t-1} + e_t$.
- Show that $Y_t = -\sum_{j=1}^{\infty} (\frac{1}{3})^j e_{t+j}$ satisfies the AR(1) equation.
 - Show that the process defined in part (a) is stationary.
 - In what way is this solution unsatisfactory?
- 4.17** Consider a process that satisfies the AR(1) equation $Y_t = 1/2Y_{t-1} + e_t$.
- Show that $Y_t = 10(\frac{1}{2})^t + e_t + \frac{1}{2}e_{t-1} + (\frac{1}{2})^2 e_{t-2} + \dots$ is a solution of the AR(1) equation.
 - Is the solution given in part (a) stationary?
- 4.18** Consider a process that satisfies the zero-mean, “stationary” AR(1) equation $Y_t = \phi Y_{t-1} + e_t$ with $-1 < \phi < +1$. Let c be any nonzero constant, and define $W_t = Y_t + c\phi^t$.
- Show that $E(W_t) = c\phi^t$.
 - Show that $\{W_t\}$ satisfies the “stationary” AR(1) equation $W_t = \phi W_{t-1} + e_t$.
 - Is $\{W_t\}$ stationary?
- 4.19** Consider an MA(6) model with $\theta_1 = 0.5$, $\theta_2 = -0.25$, $\theta_3 = 0.125$, $\theta_4 = -0.0625$, $\theta_5 = 0.03125$, and $\theta_6 = -0.015625$. Find a much simpler model that has nearly the same ψ -weights.
- 4.20** Consider an MA(7) model with $\theta_1 = 1$, $\theta_2 = -0.5$, $\theta_3 = 0.25$, $\theta_4 = -0.125$, $\theta_5 = 0.0625$, $\theta_6 = -0.03125$, and $\theta_7 = 0.015625$. Find a much simpler model that has nearly the same ψ -weights.

- 4.21** Consider the model $Y_t = e_{t-1} - e_{t-2} + 0.5e_{t-3}$.
- Find the autocovariance function for this process.
 - Show that this is a certain ARMA(p, q) process in disguise. That is, identify values for p and q and for the θ 's and ϕ 's such that the ARMA(p, q) process has the same statistical properties as $\{Y_t\}$.
- 4.22** Show that the statement “The roots of $1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p = 0$ are greater than 1 in absolute value” is equivalent to the statement “The roots of $x^p - \phi_1 x^{p-1} - \phi_2 x^{p-2} - \cdots - \phi_p = 0$ are less than 1 in absolute value.” (Hint: If G is a root of one equation, is $1/G$ a root of the other?)
- 4.23** Suppose that $\{Y_t\}$ is an AR(1) process with $\rho_1 = \phi$. Define the sequence $\{b_t\}$ as $b_t = Y_t - \phi Y_{t+1}$.
- Show that $Cov(b_t, b_{t-k}) = 0$ for all t and k .
 - Show that $Cov(b_t, Y_{t+k}) = 0$ for all t and $k > 0$.
- 4.24** Let $\{e_t\}$ be a zero-mean, unit-variance white noise process. Consider a process that begins at time $t = 0$ and is defined recursively as follows. Let $Y_0 = c_1 e_0$ and $Y_1 = c_2 Y_0 + e_1$. Then let $Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + e_t$ for $t > 1$ as in an AR(2) process.
- Show that the process mean is zero.
 - For particular values of ϕ_1 and ϕ_2 within the stationarity region for an AR(2) model, show how to choose c_1 and c_2 so that both $Var(Y_0) = Var(Y_1)$ and the lag 1 autocorrelation between Y_1 and Y_0 match that of a stationary AR(2) process with parameters ϕ_1 and ϕ_2 .
 - Once the process $\{Y_t\}$ is generated, show how to transform it to a new process that has any desired mean and variance. (This exercise suggests a convenient method for simulating stationary AR(2) processes.)
- 4.25** Consider an “AR(1)” process satisfying $Y_t = \phi Y_{t-1} + e_t$, where ϕ can be **any** number and $\{e_t\}$ is a white noise process such that e_t is independent of the past $\{Y_{t-1}, Y_{t-2}, \dots\}$. Let Y_0 be a random variable with mean μ_0 and variance σ_0^2 .
- Show that for $t > 0$ we can write
- $$Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \phi^3 e_{t-3} + \cdots + \phi^{t-1} e_1 + \phi^t Y_0.$$
- Show that for $t > 0$ we have $E(Y_t) = \phi^t \mu_0$.
 - Show that for $t > 0$
- $$Var(Y_t) = \begin{cases} \frac{1 - \phi^{2t}}{1 - \phi^2} \sigma_e^2 + \phi^{2t} \sigma_0^2 & \text{for } \phi \neq 1 \\ t \sigma_e^2 + \sigma_0^2 & \text{for } \phi = 1 \end{cases}$$
- Suppose now that $\mu_0 = 0$. Argue that, if $\{Y_t\}$ is stationary, we must have $\phi \neq 1$.
 - Continuing to suppose that $\mu_0 = 0$, show that, if $\{Y_t\}$ is stationary, then $Var(Y_t) = \sigma_e^2 / (1 - \phi^2)$ and so we must have $|\phi| < 1$.

Appendix B: The Stationarity Region for an AR(2) Process

In the second-order case, the roots of the quadratic characteristic polynomial are easily found to be

$$\frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \quad (4.B.1)$$

For stationarity we require that these roots exceed 1 in absolute value. We now show that this will be true if and only if three conditions are satisfied:

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1 \quad (4.B.2)$$

Proof: Let the reciprocals of the roots be denoted G_1 and G_2 . Then

$$\begin{aligned} G_1 &= \frac{2\phi_2}{-\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}} = \frac{2\phi_2}{-\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}} \left[\frac{-\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{-\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}} \right] \\ &= \frac{2\phi_2(-\phi_1 + \sqrt{\phi_1^2 + 4\phi_2})}{\phi_1^2 - (\phi_1^2 + 4\phi_2)} = \frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{2} \end{aligned}$$

Similarly,

$$G_2 = \frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{2}$$

We now divide the proof into two cases corresponding to real and complex roots. The roots will be real if and only if $\phi_1^2 + 4\phi_2 \geq 0$.

I. Real Roots: $|G_i| < 1$ for $i = 1$ and 2 if and only if

$$-1 < \frac{\phi_1 - \sqrt{\phi_1^2 + 4\phi_2}}{2} < \frac{\phi_1 + \sqrt{\phi_1^2 + 4\phi_2}}{2} < 1$$

or

$$-2 < \phi_1 - \sqrt{\phi_1^2 + 4\phi_2} < \phi_1 + \sqrt{\phi_1^2 + 4\phi_2} < 2.$$

Consider just the first inequality. Now $-2 < \phi_1 - \sqrt{\phi_1^2 + 4\phi_2}$ if and only if $\sqrt{\phi_1^2 + 4\phi_2} < \phi_1 + 2$ if and only if $\phi_1^2 + 4\phi_2 < \phi_1^2 + 4\phi_1 + 4$ if and only if $\phi_2 < \phi_1 + 1$, or $\phi_2 - \phi_1 < 1$.

The inequality $\phi_1 + \sqrt{\phi_1^2 + 4\phi_2} < 2$ is treated similarly and leads to $\phi_2 + \phi_1 < 1$.

These equations together with $\phi_1^2 + 4\phi_2 \geq 0$ define the stationarity region for the real root case shown in Exhibit 4.17.

II. Complex Roots: Now $\phi_1^2 + 4\phi_2 < 0$. Here G_1 and G_2 will be complex conjugates and $|G_1| = |G_2| < 1$ if and only if $|G_1|^2 < 1$. But $|G_1|^2 = [\phi_1^2 + (-\phi_1^2 - 4\phi_2)]/4 = -\phi_2$ so that $\phi_2 > -1$. This together with the inequality $\phi_1^2 + 4\phi_2 < 0$ defines the part of the stationarity region for complex roots shown in Exhibit 4.17 and establishes Equation (4.3.11). This completes the proof.

Appendix C: The Autocorrelation Function for ARMA(p, q)

Let $\{Y_t\}$ be a stationary, invertible ARMA(p, q) process. Recall that we can always write such a process in general linear process form as

$$Y_t = \sum_{j=0}^{\infty} \psi_j e_{t-j} \quad (4.C.1)$$

where the ψ -weights can be obtained recursively from Equations (4.4.7), on page 79. We then have

$$E(Y_{t+k} e_t) = E\left(\sum_{j=0}^{\infty} \psi_j e_{t+k-j} e_t\right) = \psi_k \sigma_e^2 \text{ for } k \geq 0 \quad (4.C.2)$$

Thus the autocovariance must satisfy

$$\begin{aligned} \gamma_k &= E(Y_{t+k} Y_t) = E\left[\left(\sum_{j=1}^p \phi_j Y_{t+k-j} - \sum_{j=0}^q \theta_j e_{t+k-j}\right) Y_t\right] \\ &= \sum_{j=1}^p \phi_j \gamma_{k-j} - \sigma_e^2 \sum_{j=k}^q \theta_j \psi_{j-k} \end{aligned} \quad (4.C.3)$$

where $\theta_0 = -1$ and the last sum is absent if $k > q$. Setting $k = 0, 1, \dots, p$ and using $\gamma_{-k} = \gamma_k$ leads to $p + 1$ linear equations in $\gamma_0, \gamma_1, \dots, \gamma_p$.

$$\left. \begin{aligned} \gamma_0 &= \phi_1 \gamma_1 + \phi_2 \gamma_2 + \dots + \phi_p \gamma_p - \sigma_e^2 (\theta_0 + \theta_1 \psi_1 + \dots + \theta_q \psi_q) \\ \gamma_1 &= \phi_1 \gamma_0 + \phi_2 \gamma_1 + \dots + \phi_p \gamma_{p-1} - \sigma_e^2 (\theta_1 + \theta_2 \psi_1 + \dots + \theta_q \psi_{q-1}) \\ &\vdots \\ \gamma_p &= \phi_1 \gamma_{p-1} + \phi_2 \gamma_{p-2} + \dots + \phi_p \gamma_0 - \sigma_e^2 (\theta_p + \theta_{p+1} \psi_1 + \dots + \theta_q \psi_{q-p}) \end{aligned} \right\} \quad (4.C.4)$$

where $\theta_j = 0$ if $j > q$.

For a given set of parameter values σ_e^2 , ϕ 's, and θ 's (and hence ψ 's), we can solve the linear equations to obtain $\gamma_0, \gamma_1, \dots, \gamma_p$. The values of γ_k for $k > p$ can then be evaluated from the recursion in Equations (4.4.8), on page 79. Finally, ρ_k is obtained from $\rho_k = \gamma_k / \gamma_0$.

CHAPTER 5

MODELS FOR NONSTATIONARY TIME SERIES

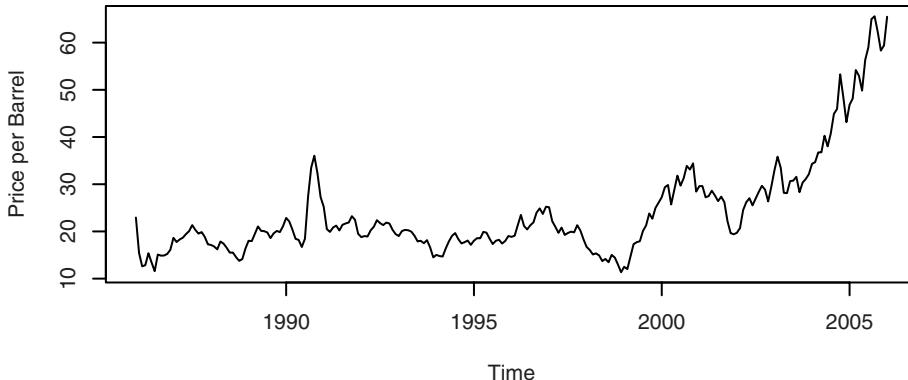
Any time series without a constant mean over time is nonstationary. Models of the form

$$Y_t = \mu_t + X_t$$

where μ_t is a nonconstant mean function and X_t is a zero-mean, stationary series, were considered in Chapter 3. As stated there, such models are reasonable only if there are good reasons for believing that the deterministic trend is appropriate “forever.” That is, just because a segment of the series looks like it is increasing (or decreasing) approximately linearly, do we believe that the linearity is intrinsic to the process and will persist in the future? Frequently in applications, particularly in business and economics, we cannot legitimately assume a deterministic trend. Recall the random walk displayed in Exhibit 2.1, on page 14. The time series appears to have a strong upward trend that might be linear in time. However, also recall that the random walk process has a constant, zero mean and contains no deterministic trend at all.

As an example consider the monthly price of a barrel of crude oil from January 1986 through January 2006. Exhibit 5.1 displays the time series plot. The series displays considerable variation, especially since 2001, and a stationary model does not seem to be reasonable. We will discover in Chapters 6, 7, and 8 that no deterministic trend model works well for this series but one of the nonstationary models that have been described as containing stochastic trends does seem reasonable. This chapter discusses such models. Fortunately, as we shall see, many stochastic trends can be modeled with relatively few parameters.

Exhibit 5.1 Monthly Price of Oil: January 1986–January 2006



```
> win.graph(width=4.875, height=3, pointsize=8)
> data(oil.price)
> plot(oil.price, ylab='Price per Barrel', type='l')
```

5.1 Stationarity Through Differencing

Consider again the AR(1) model

$$Y_t = \phi Y_{t-1} + e_t \quad (5.1.1)$$

We have seen that assuming e_t is a true “innovation” (that is, e_t is uncorrelated with Y_{t-1}, Y_{t-2}, \dots), we must have $|\phi| < 1$. What can we say about solutions to Equation (5.1.1) if $|\phi| \geq 1$? Consider in particular the equation

$$Y_t = 3Y_{t-1} + e_t \quad (5.1.2)$$

Iterating into the past as we have done before yields

$$Y_t = e_t + 3e_{t-1} + 3^2e_{t-2} + \cdots + 3^{t-1}e_1 + 3^t Y_0 \quad (5.1.3)$$

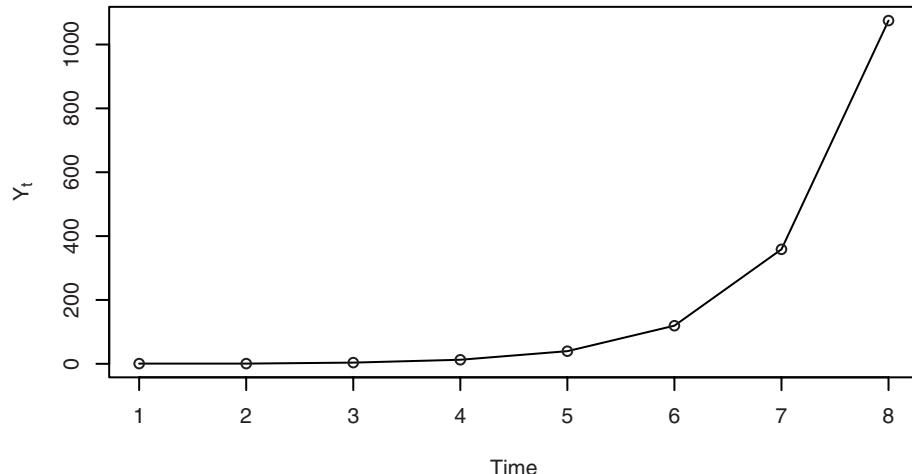
We see that the influence of distant past values of Y_t and e_t does not die out—indeed, the weights applied to Y_0 and e_1 grow exponentially large. In Exhibit 5.2, we show the values for a very short simulation of such a series. Here the white noise sequence was generated as standard normal variables and we used $Y_0 = 0$ as an initial condition.

Exhibit 5.2 Simulation of the Explosive “AR(1) Model” $Y_t = 3Y_{t-1} + e_t$

t	1	2	3	4	5	6	7	8
e_t	0.63	-1.25	1.80	1.51	1.56	0.62	0.64	-0.98
Y_t	0.63	0.64	3.72	12.67	39.57	119.33	358.63	1074.91

Exhibit 5.3 shows the time series plot of this explosive AR(1) simulation.

Exhibit 5.3 An Explosive “AR(1)” Series



```
> data(explode.s)
> plot(explode.s, ylab=expression(Y[t]), type='o')
```

The explosive behavior of such a model is also reflected in the model’s variance and covariance functions. These are easily found to be

$$\text{Var}(Y_t) = \frac{1}{8}(9^t - 1)\sigma_e^2 \quad (5.1.4)$$

and

$$\text{Cov}(Y_t, Y_{t-k}) = \frac{3^k}{8}(9^{t-k} - 1)\sigma_e^2 \quad (5.1.5)$$

respectively. Notice that we have

$$\text{Corr}(Y_t, Y_{t-k}) = 3^k \left(\frac{9^{t-k} - 1}{9^t - 1} \right) \approx 1 \quad \text{for large } t \text{ and moderate } k$$

The same general exponential growth or explosive behavior will occur for any ϕ such that $|\phi| > 1$. A more reasonable type of nonstationarity obtains when $\phi = 1$. If $\phi = 1$, the AR(1) model equation is

$$Y_t = Y_t + e_t \quad (5.1.6)$$

This is the relationship satisfied by the random walk process of Chapter 2 (Equation (2.2.9) on page 12). Alternatively, we can rewrite this as

$$\nabla Y_t = e_t \quad (5.1.7)$$

where $\nabla Y_t = Y_t - Y_{t-1}$ is the **first difference** of Y_t . The random walk then is easily extended to a more general model whose first difference is some stationary process—not just white noise.

Several somewhat different sets of assumptions can lead to models whose first difference is a stationary process. Suppose

$$Y_t = M_t + X_t \quad (5.1.8)$$

where M_t is a series that is changing only slowly over time. Here M_t could be either deterministic or stochastic. If we assume that M_t is approximately constant over every two consecutive time points, we might estimate (predict) M_t at t by choosing β_0 so that

$$\sum_{j=0}^1 (Y_{t-j} - \beta_0) {}^2$$

is minimized. This clearly leads to

$$\hat{M}_t = \frac{1}{2}(Y_t + Y_{t-1})$$

and the “detrended” series at time t is then

$$Y_t - \hat{M}_t = Y_t - \frac{1}{2}(Y_t + Y_{t-1}) = \frac{1}{2}(Y_t - Y_{t-1}) = \frac{1}{2}\nabla Y_t$$

This is a constant multiple of the first difference, ∇Y_t .[†]

A second set of assumptions might be that M_t in Equation (5.1.8) is stochastic and changes slowly over time governed by a random walk model. Suppose, for example, that

$$Y_t = M_t + e_t \quad \text{with} \quad M_t = M_{t-1} + \varepsilon_t \quad (5.1.9)$$

where $\{e_t\}$ and $\{\varepsilon_t\}$ are independent white noise series. Then

$$\begin{aligned} \nabla Y_t &= \nabla M_t + \nabla e_t \\ &= \varepsilon_t + e_t - e_{t-1} \end{aligned}$$

which would have the autocorrelation function of an MA(1) series with

$$\rho_1 = -\{1/[2 + (\sigma_\varepsilon^2/\sigma_e^2)]\} \quad (5.1.10)$$

In either of these situations, we are led to the study of ∇Y_t as a stationary process.

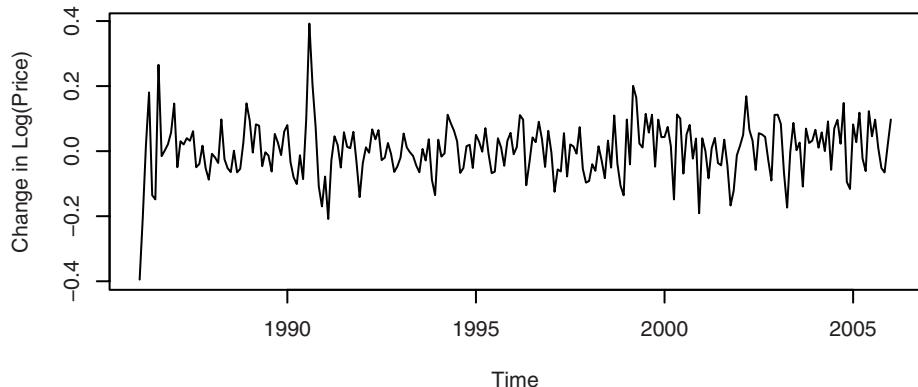
Returning to the oil price time series, Exhibit 5.4 displays the time series plot of the differences of logarithms of that series.[‡] The differenced series looks much more stationary when compared with the original time series shown in Exhibit 5.1, on page 88.

[†] A more complete labeling of this difference would be that it is a **first difference at lag 1**.

[‡] In Section 5.4 on page 98 we will see why logarithms are often a convenient transformation.

(We will also see later that there are outliers in this series that need to be considered to produce an adequate model.)

Exhibit 5.4 The Difference Series of the Logs of the Oil Price Time



```
> plot(diff(log(oil.price)), ylab='Change in Log(Price)', type='l')
```

We can also make assumptions that lead to stationary second-difference models. Again we assume that Equation (5.1.8) on page 90, holds, but now assume that M_t is linear in time over three consecutive time points. We can now estimate (predict) M_t at the middle time point t by choosing $\beta_{0,t}$ and $\beta_{1,t}$ to minimize

$$\sum_{j=-1}^1 (Y_{t-j} - (\beta_{0,t} + j\beta_{1,t}))^2$$

The solution yields

$$\hat{M}_t = \frac{1}{3}(Y_{t+1} + Y_t + Y_{t-1})$$

and thus the detrended series is

$$\begin{aligned} Y_t - \hat{M}_t &= Y_t - \left(\frac{Y_{t+1} + Y_t + Y_{t-1}}{3} \right) \\ &= \left(-\frac{1}{3} \right) (Y_{t+1} - 2Y_t + Y_{t-1}) \\ &= \left(-\frac{1}{3} \right) \nabla(\nabla Y_{t+1}) \\ &= \left(-\frac{1}{3} \right) \nabla^2(Y_{t+1}) \end{aligned}$$

a constant multiple of the centered **second difference** of Y_t . Notice that we have differenced twice, but both differences are at lag 1.

Alternatively, we might assume that

$$Y_t = M_t + e_t, \quad \text{where} \quad M_t = M_{t-1} + W_t \quad \text{and} \quad W_t = W_{t-1} + \varepsilon_t \quad (5.1.11)$$

with $\{e_t\}$ and $\{\varepsilon_t\}$ independent white noise time series. Here the stochastic trend M_t is such that its “rate of change,” ∇M_t , is changing slowly over time. Then

$$\nabla Y_t = \nabla M_t + \nabla e_t = W_t + \nabla e_t$$

and

$$\begin{aligned} \nabla^2 Y_t &= \nabla W_t + \nabla^2 e_t \\ &= \varepsilon_t + (e_t - e_{t-1}) - (e_{t-1} - e_{t-2}) \\ &= \varepsilon_t + e_t - 2e_{t-1} + e_{t-2} \end{aligned}$$

which has the autocorrelation function of an MA(2) process. The important point is that the second difference of the nonstationary process $\{Y_t\}$ is stationary. This leads us to the general definition of the important integrated autoregressive moving average time series models.

5.2 ARIMA Models

A time series $\{Y_t\}$ is said to follow an **integrated autoregressive moving average** model if the d th difference $W_t = \nabla^d Y_t$ is a stationary ARMA process. If $\{W_t\}$ follows an ARMA(p, q) model, we say that $\{Y_t\}$ is an ARIMA(p, d, q) process. Fortunately, for practical purposes, we can usually take $d = 1$ or at most 2.

Consider then an ARIMA($p, 1, q$) process. With $W_t = Y_t - Y_{t-1}$, we have

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (5.2.1)$$

or, in terms of the observed series,

$$\begin{aligned} Y_t - Y_{t-1} &= \phi_1(Y_{t-1} - Y_{t-2}) + \phi_2(Y_{t-2} - Y_{t-3}) + \cdots + \phi_p(Y_{t-p} - Y_{t-p-1}) \\ &\quad + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \end{aligned}$$

which we may rewrite as

$$\begin{aligned} Y_t &= (1 + \phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + (\phi_3 - \phi_2)Y_{t-3} + \cdots \\ &\quad + (\phi_p - \phi_{p-1})Y_{t-p} - \phi_p Y_{t-p-1} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \end{aligned} \quad (5.2.2)$$

We call this the **difference equation form** of the model. Notice that it appears to be an ARMA($p + 1, q$) process. However, the characteristic polynomial satisfies

$$\begin{aligned} 1 - (1 + \phi_1)x - (\phi_2 - \phi_1)x^2 - (\phi_3 - \phi_2)x^3 - \cdots - (\phi_p - \phi_{p-1})x^p + \phi_p x^{p+1} \\ = (1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p)(1 - x) \end{aligned}$$

which can be easily checked. This factorization clearly shows the root at $x = 1$, which implies nonstationarity. The remaining roots, however, are the roots of the characteristic polynomial of the *stationary* process ∇Y_t .

Explicit representations of the observed series in terms of either W_t or the white noise series underlying W_t are more difficult than in the stationary case. Since nonstationary processes are not in statistical equilibrium, we cannot assume that they go infinitely into the past or that they start at $t = -\infty$. However, we can and shall assume that they start at some time point $t = -m$, say, where $-m$ is earlier than time $t = 1$, at which point we first observed the series. For convenience, we take $Y_t = 0$ for $t < -m$. The difference equation $Y_t - Y_{t-1} = W_t$ can be solved by summing both sides from $t = -m$ to $t = t$ to get the representation

$$Y_t = \sum_{j=-m}^t W_j \quad (5.2.3)$$

for the ARIMA($p, 1, q$) process.

The ARIMA($p, 2, q$) process can be dealt with similarly by summing twice to get the representations

$$\begin{aligned} Y_t &= \sum_{j=-m}^t \sum_{i=-m}^j W_i \\ &= \sum_{j=0}^{t+m} (j+1)W_{t-j} \end{aligned} \quad (5.2.4)$$

These representations have limited use but can be used to investigate the covariance properties of ARIMA models and also to express Y_t in terms of the white noise series $\{e_t\}$. We defer the calculations until we evaluate specific cases.

If the process contains no autoregressive terms, we call it an integrated moving average and abbreviate the name to IMA(d, q). If no moving average terms are present, we denote the model as ARI(p, d). We first consider in detail the important IMA(1,1) model.

The IMA(1,1) Model

The simple IMA(1,1) model satisfactorily represents numerous time series, especially those arising in economics and business. In difference equation form, the model is

$$Y_t = Y_{t-1} + e_t - \theta e_{t-1} \quad (5.2.5)$$

To write Y_t explicitly as a function of present and past noise values, we use Equation (5.2.3) and the fact that $W_t = e_t - \theta e_{t-1}$ in this case. After a little rearrangement, we can write

$$Y_t = e_t + (1-\theta)e_{t-1} + (1-\theta)e_{t-2} + \dots + (1-\theta)e_{-m} - \theta e_{-m-1} \quad (5.2.6)$$

Notice that in contrast to our stationary ARMA models, the weights on the white noise terms *do not die out* as we go into the past. Since we are assuming that $-m < 1$ and $0 < t$, we may usefully think of Y_t as mostly an equally weighted accumulation of a large number of white noise values.

From Equation (5.2.6), we can easily derive variances and correlations. We have

$$\text{Var}(Y_t) = [1 + \theta^2 + (1 - \theta)^2(t + m)]\sigma_e^2 \quad (5.2.7)$$

and

$$\begin{aligned} \text{Corr}(Y_t, Y_{t-k}) &= \frac{1 - \theta + \theta^2 + (1 - \theta)^2(t + m - k)}{[\text{Var}(Y_t)\text{Var}(Y_{t-k})]^{1/2}} \\ &\approx \sqrt{\frac{t + m - k}{t + m}} \\ &\approx 1 \quad \text{for large } m \text{ and moderate } k \end{aligned} \quad (5.2.8)$$

We see that as t increases, $\text{Var}(Y_t)$ increases and could be quite large. Also, the correlation between Y_t and Y_{t-k} will be strongly positive for many lags $k = 1, 2, \dots$.

The IMA(2,2) Model

The assumptions of Equation (5.1.11) led to an IMA(2,2) model. In difference equation form, we have

$$\nabla^2 Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}$$

or

$$Y_t = 2Y_{t-1} - Y_{t-2} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} \quad (5.2.9)$$

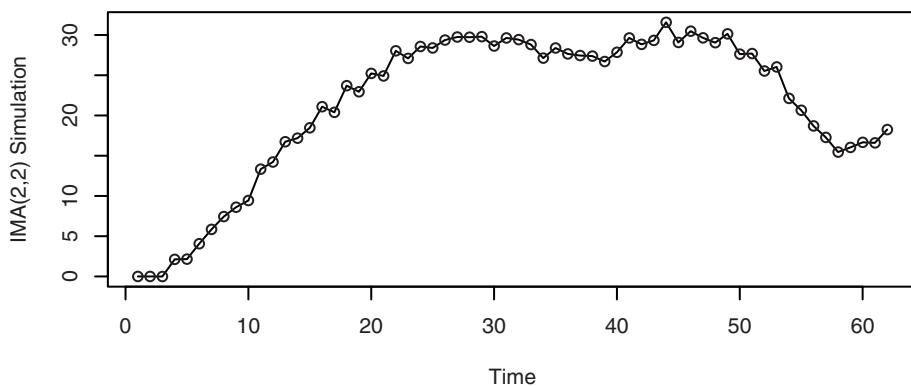
The representation of Equation (5.2.4) may be used to express Y_t in terms of e_t, e_{t-1}, \dots . After some tedious algebra, we find that

$$\begin{aligned} Y_t = e_t + \sum_{j=1}^{t+m} \psi_j e_{t-j} - [(t+m+1)\theta_1 + (t+m)\theta_2]e_{-m-1} \\ - (t+m+1)\theta_2 e_{-m-2} \end{aligned} \quad (5.2.10)$$

where $\psi_j = 1 + \theta_2 + (1 - \theta_1 - \theta_2)j$ for $j = 1, 2, 3, \dots, t+m$. Once more we see that the ψ -weights do not die out but form a linear function of j .

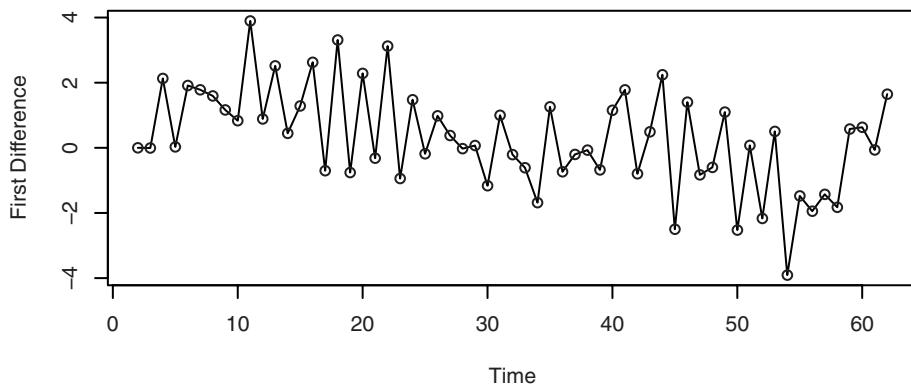
Again, variances and correlations for Y_t can be obtained from the representation given in Equation (5.2.10), but the calculations are tedious. We shall simply note that the variance of Y_t increases rapidly with t and again $\text{Corr}(Y_t, Y_{t-k})$ is nearly 1 for all moderate k .

The results of a simulation of an IMA(2,2) process are displayed in Exhibit 5.5. Notice the smooth change in the process values (and the unimportance of the zero-mean function). The increasing variance and the strong, positive neighboring correlations dominate the appearance of the time series plot.

Exhibit 5.5 Simulation of an IMA(2,2) Series with $\theta_1 = 1$ and $\theta_2 = -0.6$ 

```
> data(im22.s)
> plot(im22.s, ylab='IMA(2,2) Simulation', type='o')
```

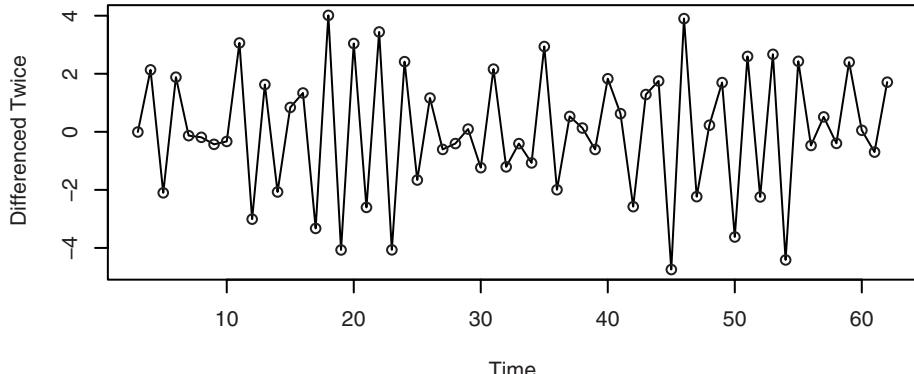
Exhibit 5.6 shows the time series plot of the first difference of the simulated series. This series is also nonstationary, as it is governed by an IMA(1,2) model.

Exhibit 5.6 First Difference of the Simulated IMA(2,2) Series

```
> plot(diff(im22.s), ylab='First Difference', type='o')
```

Finally, the second differences of the simulated IMA(2,2) series values are plotted in Exhibit 5.7. These values arise from a stationary MA(2) model with $\theta_1 = 1$ and $\theta_2 = -0.6$. From Equation (4.2.3) on page 63, the theoretical autocorrelations for this model are $\rho_1 = -0.678$ and $\rho_2 = 0.254$. These correlation values seem to be reflected in the appearance of the time series plot.

Exhibit 5.7 Second Difference of the Simulated IMA(2,2) Series



```
> plot(diff(ima22.s,difference=2),ylab='Differenced  
Twice',type='o')
```

The ARI(1,1) Model

The ARI(1,1) process will satisfy

$$Y_t - Y_{t-1} = \phi(Y_{t-1} - Y_{t-2}) + e_t \quad (5.2.11)$$

or

$$Y_t = (1 + \phi)Y_{t-1} - \phi Y_{t-2} + e_t \quad (5.2.12)$$

where $|\phi| < 1$.[†]

To find the ψ -weights in this case, we shall use a technique that will generalize to arbitrary ARIMA models. It can be shown that the ψ -weights can be obtained by equating like powers of x in the identity:

$$\begin{aligned} (1 - \phi_1 x - \phi_2 x^2 - \dots - \phi^p x^p)(1 - x)^d(1 + \psi_1 x + \psi_2 x^2 + \psi_3 x^3 + \dots) \\ = (1 - \theta_1 x - \theta_2 x^2 - \theta_3 x^3 - \dots - \theta_q x^q) \end{aligned} \quad (5.2.13)$$

In our case, this relationship reduces to

$$(1 - \phi x)(1 - x)(1 + \psi_1 x + \psi_2 x^2 + \psi_3 x^3 + \dots) = 1$$

or

$$[1 - (1 + \phi)x + \phi x^2](1 + \psi_1 x + \psi_2 x^2 + \psi_3 x^3 + \dots) = 1$$

Equating like powers of x on both sides, we obtain

[†] Notice that this looks like a special AR(2) model. However, one of the roots of the corresponding AR(2) characteristic polynomial is 1, and this is not allowed in stationary AR(2) models.

$$\begin{aligned} - (1 + \phi) + \psi_1 &= 0 \\ \phi - (1 + \phi)\psi_1 + \psi_2 &= 0 \end{aligned}$$

and, in general,

$$\psi_k = (1 + \phi)\psi_{k-1} - \phi\psi_{k-2} \quad \text{for } k \geq 2 \quad (5.2.14)$$

with $\psi_0 = 1$ and $\psi_1 = 1 + \phi$. This recursion with starting values allows us to compute as many ψ -weights as necessary. It can also be shown that in this case an explicit solution to the recursion is given as

$$\psi_k = \frac{1 - \phi^{k+1}}{1 - \phi} \quad \text{for } k \geq 1 \quad (5.2.15)$$

(It is easy, for example, to show that this expression satisfies Equation (5.2.14)).

5.3 Constant Terms in ARIMA Models

For an ARIMA(p, d, q) model, $\nabla^d Y_t = W_t$ is a stationary ARMA(p, q) process. Our standard assumption is that stationary models have a zero mean; that is, we are actually working with deviations from the constant mean. A nonzero constant mean, μ , in a stationary ARMA model $\{W_t\}$ can be accommodated in either of two ways. We can assume that

$$\begin{aligned} W_t - \mu &= \phi_1(W_{t-1} - \mu) + \phi_2(W_{t-2} - \mu) + \cdots + \phi_p(W_{t-p} - \mu) \\ &\quad + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \end{aligned}$$

Alternatively, we can introduce a constant term θ_0 into the model as follows:

$$\begin{aligned} W_t &= \theta_0 + \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} \\ &\quad + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \end{aligned}$$

Taking expected values on both sides of the latter expression, we find that

$$\mu = \theta_0 + (\phi_1 + \phi_2 + \cdots + \phi_p)\mu$$

so that

$$\mu = \frac{\theta_0}{1 - \phi_1 - \phi_2 - \cdots - \phi_p} \quad (5.3.16)$$

or, conversely, that

$$\theta_0 = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p) \quad (5.3.17)$$

Since the alternative representations are equivalent, we shall use whichever parameterization is convenient.

What will be the effect of a nonzero mean for W_t on the undifferenced series Y_t ? Consider the IMA(1,1) case with a constant term. We have

$$Y_t = Y_{t-1} + \theta_0 + e_t - \theta e_{t-1}$$

or

$$W_t = \theta_0 + e_t - \theta e_{t-1}$$

Either by substituting into Equation (5.2.3) on page 93 or by iterating into the past, we find that

$$\begin{aligned} Y_t = e_t + (1-\theta)e_{t-1} + (1-\theta)e_{t-2} + \cdots + (1-\theta)e_{-m} - \theta e_{-m-1} \\ + (t+m+1)\theta_0 \end{aligned} \quad (5.3.18)$$

Comparing this with Equation (5.2.6), we see that we have an added *linear deterministic time trend* $(t+m+1)\theta_0$ with slope θ_0 .

An equivalent representation of the process would then be

$$Y_t = Y'_t + \beta_0 + \beta_1 t$$

where Y'_t is an IMA(1,1) series with $E(\nabla Y'_t) = 0$ and $E(\nabla Y_t) = \beta_1$.

For a general ARIMA(p,d,q) model where $E(\nabla^d Y_t) \neq 0$, it can be argued that $Y_t = Y'_t + \mu_t$, where μ_t is a deterministic polynomial of degree d and Y'_t is ARIMA(p,d,q) with $E Y'_t = 0$. With $d=2$ and $\theta_0 \neq 0$, a quadratic trend would be implied.

5.4 Other Transformations

We have seen how differencing can be a useful transformation for achieving stationarity. However, the logarithm transformation is also a useful method in certain circumstances. We frequently encounter series where increased dispersion seems to be associated with higher levels of the series—the higher the level of the series, the more variation there is around that level and conversely.

Specifically, suppose that $Y_t > 0$ for all t and that

$$E(Y_t) = \mu_t \quad \text{and} \quad \sqrt{Var(Y_t)} = \mu_t \sigma \quad (5.4.1)$$

Then

$$E[\log(Y_t)] \approx \log(\mu_t) \quad \text{and} \quad Var(\log(Y_t)) \approx \sigma^2 \quad (5.4.2)$$

These results follow from taking expected values and variances of both sides of the (Taylor) expansion

$$\log(Y_t) \approx \log(\mu_t) + \frac{Y_t - \mu_t}{\mu_t}$$

In words, if the standard deviation of the series is proportional to the level of the series, then transforming to logarithms will produce a series with approximately constant variance over time. Also, if the level of the series is changing roughly exponentially, the

log-transformed series will exhibit a linear time trend. Thus, we might then want to take first differences. An alternative set of assumptions leading to differences of logged data follows.

Percentage Changes and Logarithms

Suppose Y_t tends to have relatively stable percentage changes from one time period to the next. Specifically, assume that

$$Y_t = (1 + X_t)Y_{t-1}$$

where $100X_t$ is the percentage change (possibly negative) from Y_{t-1} to Y_t . Then

$$\begin{aligned} \log(Y_t) - \log(Y_{t-1}) &= \log\left(\frac{Y_t}{Y_{t-1}}\right) \\ &= \log(1 + X_t) \end{aligned}$$

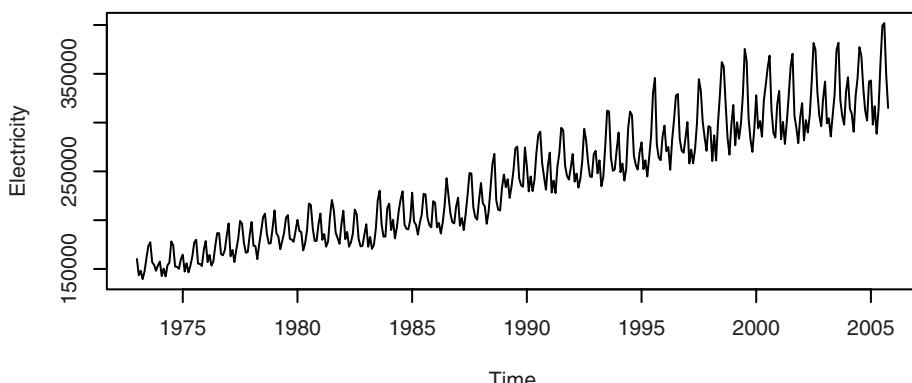
If X_t is restricted to, say, $|X_t| < 0.2$ (that is, the percentage changes are at most $\pm 20\%$), then, to a good approximation, $\log(1+X_t) \approx X_t$. Consequently,

$$\nabla[\log(Y_t)] \approx X_t \quad (5.4.3)$$

will be relatively stable and perhaps well-modeled by a stationary process. Notice that we take logs first and then compute first differences—the order does matter. In financial literature, the differences of the (natural) logarithms are usually called **returns**.

As an example, consider the time series shown in Exhibit 5.8. This series gives the total monthly electricity generated in the United States in millions of kilowatt-hours. The higher values display considerably more variation than the lower values.

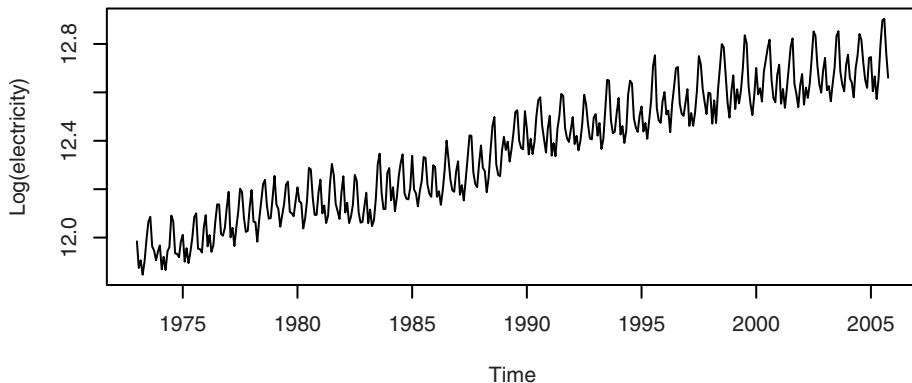
Exhibit 5.8 U.S. Electricity Generated by Month



```
> data(electricity); plot(electricity)
```

Exhibit 5.9 displays the time series plot of the logarithms of the electricity values. Notice how the amount of variation around the upward trend is now much more uniform across high and low values of the series.

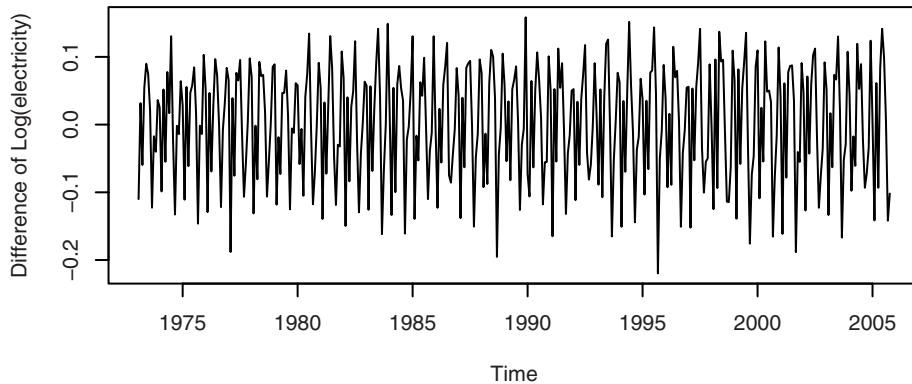
Exhibit 5.9 Time Series Plot of Logarithms of Electricity Values



```
> plot(log(electricity), ylab='Log(electricity)')
```

The differences of the logarithms of the electricity values are displayed in Exhibit 5.10. On the basis of this plot, we might well consider a stationary model as appropriate.

Exhibit 5.10 Difference of Logarithms for Electricity Time Series



```
> plot(diff(log(electricity)),
      ylab='Difference of Log(electricity)')
```

Power Transformations

A flexible family of transformations, the **power transformations**, was introduced by Box and Cox (1964). For a given value of the parameter λ , the transformation is defined by

$$g(x) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0 \\ \log x & \text{for } \lambda = 0 \end{cases} \quad (5.4.4)$$

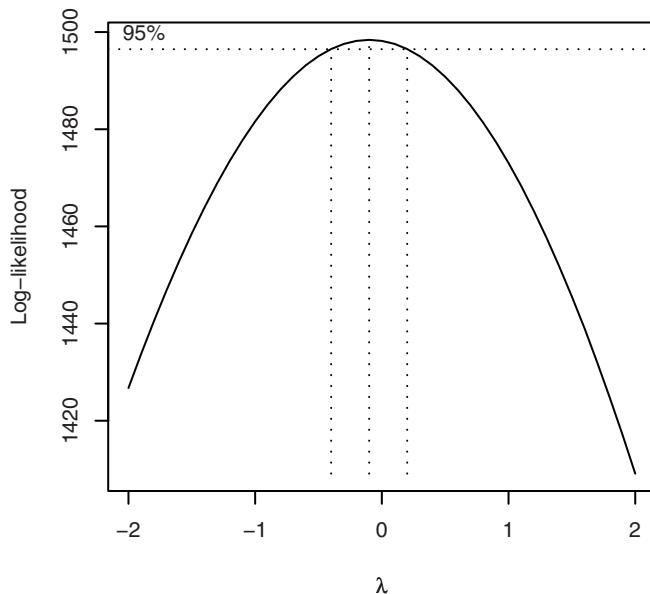
The term x^λ is the important part of the first expression, but subtracting 1 and dividing by λ makes $g(x)$ change smoothly as λ approaches zero. In fact, a calculus argument[†] shows that as $\lambda \rightarrow 0$, $(x^\lambda - 1)/\lambda \rightarrow \log(x)$. Notice that $\lambda = 1/2$ produces a square root transformation useful with Poisson-like data, and $\lambda = -1$ corresponds to a reciprocal transformation.

The power transformation applies only to positive data values. If some of the values are negative or zero, a positive constant may be added to all of the values to make them all positive before doing the power transformation. The shift is often determined subjectively. For example, for nonnegative catch data in biology, the occurrence of zeros is often dealt with by adding a constant equal to the smallest positive data value to all of the data values. An alternative approach consists of using transformations applicable to any data—positive or not. A drawback of this alternative approach is that interpretations of such transformations are often less straightforward than the interpretations of the power transformations. See Yeo and Johnson (2000) and the references contained therein.

We can consider λ as an additional parameter in the model to be estimated from the observed data. However, precise estimation of λ is usually not warranted. Evaluation of a range of transformations based on a grid of λ values, say $\pm 1, \pm 1/2, \pm 1/3, \pm 1/4$, and 0, will usually suffice and may have some intuitive meaning.

Software allows us to consider a range of lambda values and calculate a log-likelihood value for each lambda value based on a normal likelihood function. A plot of these values is shown in Exhibit 5.11 for the electricity data. The 95% confidence interval for λ contains the value of $\lambda = 0$ quite near its center and strongly suggests a logarithmic transformation ($\lambda = 0$) for these data.

[†] Exercise (5.17) asks you to verify this.

Exhibit 5.11 Log-likelihood versus Lambda

```
> BoxCox.ar(electricity)
```

5.5 Summary

This chapter introduced the concept of differencing to induce stationarity on certain nonstationary processes. This led to the important integrated autoregressive moving average models (ARIMA). The properties of these models were then thoroughly explored. Other transformations, namely percentage changes and logarithms, were then considered. More generally, power transformations or Box-Cox transformations were introduced as useful transformations to stationarity and often normality.

EXERCISES

- 5.1** Identify the following as specific ARIMA models. That is, what are p , d , and q and what are the values of the parameters (the ϕ 's and θ 's)?
- $Y_t = Y_{t-1} - 0.25Y_{t-2} + e_t - 0.1e_{t-1}$.
 - $Y_t = 2Y_{t-1} - Y_{t-2} + e_t$.
 - $Y_t = 0.5Y_{t-1} - 0.5Y_{t-2} + e_t - 0.5e_{t-1} + 0.25e_{t-2}$.
- 5.2** For each of the ARIMA models below, give the values for $E(\nabla Y_t)$ and $Var(\nabla Y_t)$.
- $Y_t = 3 + Y_{t-1} + e_t - 0.75e_{t-1}$.
 - $Y_t = 10 + 1.25Y_{t-1} - 0.25Y_{t-2} + e_t - 0.1e_{t-1}$.
 - $Y_t = 5 + 2Y_{t-1} - 1.7Y_{t-2} + 0.7Y_{t-3} + e_t - 0.5e_{t-1} + 0.25e_{t-2}$.
- 5.3** Suppose that $\{Y_t\}$ is generated according to $Y_t = e_t + ce_{t-1} + ce_{t-2} + ce_{t-3} + \dots + ce_0$ for $t > 0$.
- Find the mean and covariance functions for $\{Y_t\}$. Is $\{Y_t\}$ stationary?
 - Find the mean and covariance functions for $\{\nabla Y_t\}$. Is $\{\nabla Y_t\}$ stationary?
 - Identify $\{Y_t\}$ as a specific ARIMA process.
- 5.4** Suppose that $Y_t = A + Bt + X_t$, where $\{X_t\}$ is a random walk. First suppose that A and B are constants.
- Is $\{Y_t\}$ stationary?
 - Is $\{\nabla Y_t\}$ stationary?
- Now suppose that A and B are random variables that are independent of the random walk $\{X_t\}$.
- Is $\{Y_t\}$ stationary?
 - Is $\{\nabla Y_t\}$ stationary?
- 5.5** Using the simulated white noise values in Exhibit 5.2, on page 88, verify the values shown for the explosive process Y_t .
- 5.6** Consider a stationary process $\{Y_t\}$. Show that if $\rho_1 < \frac{1}{2}$, ∇Y_t has a larger variance than does Y_t .
- 5.7** Consider two models:
- A: $Y_t = 0.9Y_{t-1} + 0.09Y_{t-2} + e_t$
- B: $Y_t = Y_{t-1} + e_t - 0.1e_{t-1}$.
- Identify each as a specific ARIMA model. That is, what are p , d , and q and what are the values of the parameters, ϕ 's and θ 's?
 - In what ways are the two models different?
 - In what ways are the two models similar? (Compare ψ -weights and π -weights.)

- 5.8** Consider a nonstationary “AR(1)” process defined as a solution to Equation (5.1.2) on page 88, with $|\phi| > 1$.
- Derive an equation similar to Equation (5.1.3) on page 88, for this more general case. Use $Y_0 = 0$ as an initial condition.
 - Derive an equation similar to Equation (5.1.4) on page 89, for this more general case.
 - Derive an equation similar to Equation (5.1.5) on page 89, for this more general case.
 - Is it true that for any $|\phi| > 1$, $\text{Corr}(Y_t, Y_{t-k}) \approx 1$ for large t and moderate k ?
- 5.9** Verify Equation (5.1.10) on page 90.
- 5.10** Nonstationary ARIMA series can be simulated by first simulating the corresponding stationary ARMA series and then “integrating” it (really partially summing it). Use statistical software to simulate a variety of IMA(1,1) and IMA(2,2) series with a variety of parameter values. Note any stochastic “trends” in the simulated series.
- 5.11** The data file `winnebago` contains monthly unit sales of recreational vehicles (RVs) from Winnebago, Inc., from November 1966 through February 1972.
- Display and interpret the time series plot for these data.
 - Now take natural logarithms of the monthly sales figures and display the time series plot of the transformed values. Describe the effect of the logarithms on the behavior of the series.
 - Calculate the fractional relative changes, $(Y_t - Y_{t-1})/Y_{t-1}$, and compare them with the differences of (natural) logarithms, $\nabla \log(Y_t) = \log(Y_t) - \log(Y_{t-1})$. How do they compare for smaller values and for larger values?
- 5.12** The data file `SP` contains quarterly Standard & Poor’s Composite Index stock price values from the first quarter of 1936 through the fourth quarter of 1977.
- Display and interpret the time series plot for these data.
 - Now take natural logarithms of the quarterly values and display and the time series plot of the transformed values. Describe the effect of the logarithms on the behavior of the series.
 - Calculate the (fractional) relative changes, $(Y_t - Y_{t-1})/Y_{t-1}$, and compare them to the differences of (natural) logarithms, $\nabla \log(Y_t)$. How do they compare for smaller values and for larger values?
- 5.13** The data file `airpass` contains monthly U.S. air passenger miles flown from January 1949 through December 1960. This is a classic time series analyzed in Box and Jenkins (1976).
- Display and interpret the time series plot for these data.
 - Now take natural logarithms of the monthly values and display and the time series plot of the transformed values. Describe the effect of the logarithms on the behavior of the series.
 - Calculate the (fractional) relative changes, $(Y_t - Y_{t-1})/Y_{t-1}$, and compare them to the differences of (natural) logarithms, $\nabla \log(Y_t)$. How do they compare for smaller values and for larger values?

- 5.14** Consider the annual rainfall data for Los Angeles shown in Exhibit 1.1, on page 2. The quantile-quantile normal plot of these data, shown in Exhibit 3.17, on page 50, convinced us that the data were not normal. The data are in the file larain.
- (a) Use software to produce a plot similar to Exhibit 5.11, on page 102, and determine the “best” value of λ for a power transformation of the data.
 - (b) Display a quantile-quantile plot of the transformed data. Are they more normal?
 - (c) Produce a time series plot of the transformed values.
 - (d) Use the transformed values to display a plot of Y_t versus Y_{t-1} as in Exhibit 1.2, on page 2. Should we expect the transformation to change the dependence or lack of dependence in the series?
- 5.15** Quarterly earnings per share for the Johnson & Johnson Company are given in the data file named JJ. The data cover the years from 1960 through 1980.
- (a) Display a time series plot of the data. Interpret the interesting features in the plot.
 - (b) Use software to produce a plot similar to Exhibit 5.11, on page 102, and determine the “best” value of λ for a power transformation of these data.
 - (c) Display a time series plot of the transformed values. Does this plot suggest that a stationary model might be appropriate?
 - (d) Display a time series plot of the differences of the transformed values. Does this plot suggest that a stationary model might be appropriate for the differences?
- 5.16** The file named gold contains the daily price of gold (in dollars per troy ounce) for the 252 trading days of year 2005.
- (a) Display the time series plot of these data. Interpret the plot.
 - (b) Display the time series plot of the differences of the logarithms of these data. Interpret this plot.
 - (c) Calculate and display the sample ACF for the differences of the logarithms of these data and argue that the logarithms appear to follow a random walk model.
 - (d) Display the differences of logs in a histogram and interpret.
 - (e) Display the differences of logs in a quantile-quantile normal plot and interpret.
- 5.17** Use calculus to show that, for any fixed $x > 0$, as $\lambda \rightarrow 0$, $(x^\lambda - 1)/\lambda \rightarrow \log x$.

Appendix D: The Backshift Operator

Many other books and much of the time series literature use what is called the **backshift operator** to express and manipulate ARIMA models. The backshift operator, denoted B , operates on the time index of a series and shifts time back one time unit to form a new series.[†] In particular,

$$BY_t = Y_{t-1}$$

The backshift operator is linear since for any constants a , b , and c and series Y_t and X_t , it is easy to see that

$$B(aY_t + bX_t + c) = aBY_t + bBX_t + c$$

Consider now the MA(1) model. In terms of B , we can write

$$\begin{aligned} Y_t &= e_t - \theta e_{t-1} = e_t - \theta B e_t = (1 - \theta B)e_t \\ &= \theta(B)e_t \end{aligned}$$

where $\theta(B)$ is the MA characteristic polynomial “evaluated” at B .

Since BY_t is itself a time series, it is meaningful to consider BBY_t . But clearly $BBY_t = BY_{t-1} = Y_{t-2}$, and we can write

$$B^2Y_t = Y_{t-2}$$

More generally, we have

$$B^m Y_t = Y_{t-m}$$

for any positive integer m . For a general MA(q) model, we can then write

$$\begin{aligned} Y_t &= e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \\ &= e_t - \theta_1 Be_t - \theta_2 B^2 e_t - \cdots - \theta_q B^q e_t \\ &= (1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q)e_t \end{aligned}$$

or

$$Y_t = \theta(B)e_t$$

where, again, $\theta(B)$ is the MA characteristic polynomial evaluated at B .

For autoregressive models AR(p), we first move all of the terms involving Y to the left-hand side

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p} = e_t$$

and then write

$$Y_t - \phi_1 BY_t - \phi_2 B^2 Y_t - \cdots - \phi_p B^p Y_t = e_t$$

or

[†] Sometimes B is called a **Lag operator**.

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) Y_t = e_t$$

which can be expressed as

$$\phi(B)Y_t = e_t$$

where $\phi(B)$ is the AR characteristic polynomial evaluated at B .

Combining the two, the general ARMA(p, q) model may be written compactly as

$$\phi(B)Y_t = \theta(B)e_t$$

Differencing can also be conveniently expressed in terms of B . We have

$$\begin{aligned} \nabla Y_t &= Y_t - Y_{t-1} = Y_t - BY_t \\ &= (1 - B)Y_t \end{aligned}$$

with second differences given by

$$\nabla^2 Y_t = (1 - B)^2 Y_t$$

Effectively, $\nabla = 1 - B$ and $\nabla^2 = (1 - B)^2$.

The general ARIMA(p, d, q) model is expressed concisely as

$$\phi(B)(1 - B)^d Y_t = \theta(B)e_t$$

In the literature, one must carefully distinguish from the context the use of B as a backshift operator and its use as an ordinary real (or complex) variable. For example, the stationarity condition is frequently given by stating that the roots of $\phi(B) = 0$ must be greater than 1 in absolute value or, equivalently, must lie outside the unit circle in the complex plane. Here B is to be treated as a dummy variable in an equation rather than as the backshift operator.

CHAPTER 6

MODEL SPECIFICATION

We have developed a large class of parametric models for both stationary and nonstationary time series—the ARIMA models. We now begin our study and implementation of *statistical inference* for such models. The subjects of the next three chapters, respectively, are:

1. how to choose appropriate values for p , d , and q for a given series;
2. how to estimate the parameters of a specific ARIMA(p, d, q) model;
3. how to check on the appropriateness of the fitted model and improve it if needed.

Our overall strategy will first be to decide on reasonable—but tentative—values for p , d , and q . Having done so, we shall estimate the ϕ 's, θ 's, and σ_e for that model in the most efficient way. Finally, we shall look critically at the fitted model thus obtained to check its adequacy, in much the same way that we did in Section 3.6 on page 42. If the model appears inadequate in some way, we consider the nature of the inadequacy to help us select another model. We proceed to estimate that new model and check it for adequacy.

With a few iterations of this model-building strategy, we hope to arrive at the best possible model for a given series. The book by George E. P. Box and G. M. Jenkins (1976) so popularized this technique that many authors call the procedure the “Box-Jenkins method.” We begin by continuing our investigation of the properties of the sample autocorrelation function.

6.1 Properties of the Sample Autocorrelation Function

Recall from page 46 the definition of the sample or estimated autocorrelation function. For the observed series Y_1, Y_2, \dots, Y_n , we have

$$r_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \quad \text{for } k = 1, 2, \dots \quad (6.1.1)$$

Our goal is to recognize, to the extent possible, patterns in r_k that are characteristic of the known patterns in ρ_k for common ARMA models. For example, we know that $\rho_k = 0$ for $k > q$ in an MA(q) model. However, as the r_k are only estimates of the ρ_k , we

need to investigate their sampling properties to facilitate the comparison of estimated correlations with theoretical correlations.

From the definition of r_k , a ratio of quadratic functions of possibly dependent variables, it should be apparent that the sampling properties of r_k will *not* be obtained easily. Even the expected value of r_k is difficult to determine—recall that the expected value of a ratio is *not* the ratio of the respective expected values. We shall be content to accept a general large-sample result and consider its implications in special cases. Bartlett (1946) carried out the original work. We shall take a more general result from Anderson (1971). A recent discussion of these results may be found in Shumway and Stoffer (2006, p. 519).

We suppose that

$$Y_t = \mu + \sum_{j=0}^{\infty} \psi_j e_{t-j}$$

where the e_t are independent and identically distributed with zero means and finite, non-zero, common variances. We assume further that

$$\sum_{j=0}^{\infty} |\psi_j| < \infty \quad \text{and} \quad \sum_{j=0}^{\infty} j\psi_j^2 < \infty$$

(These will be satisfied by any stationary ARMA model.)

Then, for any fixed m , the joint distribution of

$$\sqrt{n}(r_1 - \rho_1), \sqrt{n}(r_2 - \rho_2), \dots, \sqrt{n}(r_m - \rho_m)$$

approaches, as $n \rightarrow \infty$, a joint normal distribution with zero means, variances c_{jj} , and covariances c_{ij} , where

$$c_{ij} = \sum_{k=-\infty}^{\infty} (\rho_{k+i}\rho_{k+j} + \rho_{k-i}\rho_{k+j} - 2\rho_i\rho_k\rho_{k+j} - 2\rho_j\rho_k\rho_{k+i} + 2\rho_i\rho_j\rho_k^2) \quad (6.1.2)$$

For large n , we would say that r_k is approximately normally distributed with mean ρ_k and variance c_{kk}/n . Furthermore, $\text{Corr}(r_k, r_j) \approx c_{kj}/\sqrt{c_{kk}c_{jj}}$. Notice that the approximate variance of r_k is inversely proportional to the sample size, but $\text{Corr}(r_k, r_j)$ is approximately *constant* for large n .

Since Equation (6.1.2) is clearly difficult to interpret in its present generality, we shall consider some important special cases and simplifications. Suppose first that $\{Y_t\}$ is white noise. Then Equation (6.1.2) reduces considerably, and we obtain

$$\text{Var}(r_k) \approx \frac{1}{n} \quad \text{and} \quad \text{Corr}(r_k, r_j) \approx 0 \quad \text{for } k \neq j \quad (6.1.3)$$

Next suppose that $\{Y_t\}$ is generated by an AR(1) process with $\rho_k = \phi^k$ for $k > 0$. Then, after considerable algebra and summing several geometric series, Equation (6.1.2) with $i = j$ yields

$$\text{Var}(r_k) \approx \frac{1}{n} \left[\frac{(1 + \phi^2)(1 - \phi^{2k})}{1 - \phi^2} - 2k\phi^{2k} \right] \quad (6.1.4)$$

In particular,

$$\text{Var}(r_1) \approx \frac{1 - \phi^2}{n} \quad (6.1.5)$$

Notice that the closer ϕ is to ± 1 , the more precise our estimate of ρ_1 ($= \phi$) becomes.

For large lags, the terms in Equation (6.1.4) involving ϕ^k may be ignored, and we have

$$\text{Var}(r_k) \approx \frac{1}{n} \left[\frac{1 + \phi^2}{1 - \phi^2} \right] \text{ for large } k \quad (6.1.6)$$

Notice that here, in contrast to Equation (6.1.5), values of ϕ close to ± 1 imply large variances for r_k . Thus we should not expect nearly as precise estimates of $\rho_k = \phi^k \approx 0$ for large k as we do of $\rho_k = \phi^k$ for small k .

For the AR(1) model, Equation (6.1.2) can also be simplified (after much algebra) for general $0 < i < j$ as

$$c_{ij} = \frac{(\phi^{j-i} - \phi^{j+i})(1 + \phi^2)}{1 - \phi^2} + (j-i)\phi^{j-i} - (j+i)\phi^{j+i} \quad (6.1.7)$$

In particular, we find

$$\text{Corr}(r_1, r_2) \approx 2\phi \sqrt{\frac{1 - \phi^2}{1 + 2\phi^2 - 3\phi^4}} \quad (6.1.8)$$

Based on Equations (6.1.4) through (6.1.8), Exhibit 6.1 gives approximate standard deviations and correlations for several lags and a few values of ϕ in AR(1) models.

Exhibit 6.1 Large Sample Results for Selected r_k from an AR(1) Model

ϕ	$\sqrt{\text{Var}(r_1)}$	$\sqrt{\text{Var}(r_2)}$	$\text{Corr}(r_1, r_2)$	$\sqrt{\text{Var}(r_{10})}$
± 0.9	$0.44/\sqrt{n}$	$0.807/\sqrt{n}$	± 0.97	$2.44/\sqrt{n}$
± 0.7	$0.71/\sqrt{n}$	$1.12/\sqrt{n}$	± 0.89	$1.70/\sqrt{n}$
± 0.4	$0.92/\sqrt{n}$	$1.11/\sqrt{n}$	± 0.66	$1.18/\sqrt{n}$
± 0.2	$0.98/\sqrt{n}$	$1.04/\sqrt{n}$	± 0.38	$1.04/\sqrt{n}$

For the MA(1) case, Equation (6.1.2) simplifies as follows:

$$c_{11} = 1 - 3\rho_1^2 + 4\rho_1^4 \quad \text{and} \quad c_{kk} = 1 + 2\rho_1^2 \quad \text{for } k > 1 \quad (6.1.9)$$

Furthermore,

$$c_{12} = 2\rho_1(1 - \rho_1^2) \quad (6.1.10)$$

Based on these expressions, Exhibit 6.2 lists large-sample standard deviations and correlations for the sample autocorrelations for several lags and several θ -values. Notice again that the sample autocorrelations can be highly correlated and that the standard deviation of r_k is larger for $k > 1$ than for $k = 1$.

Exhibit 6.2 Large-Sample Results for Selected r_k from an MA(1) Model

θ	$\sqrt{Var(r_1)}$	$\sqrt{Var(r_k)}$ for $k > 1$	$Corr(r_1, r_2)$
± 0.9	$0.71/\sqrt{n}$	$1.22/\sqrt{n}$	∓ 0.86
± 0.7	$0.73/\sqrt{n}$	$1.20/\sqrt{n}$	∓ 0.84
± 0.5	$0.79/\sqrt{n}$	$1.15/\sqrt{n}$	∓ 0.74
± 0.4	$0.89/\sqrt{n}$	$1.11/\sqrt{n}$	∓ 0.53

For a general MA(q) process and $i = j = k$, Equation (6.1.2) reduces to

$$c_{kk} = 1 + 2 \sum_{j=1}^q \rho_j^2 \text{ for } k > q$$

so that

$$Var(r_k) = \frac{1}{n} \left[1 + 2 \sum_{j=1}^q \rho_j^2 \right] \text{ for } k > q \quad (6.1.11)$$

For an observed time series, we can replace ρ 's by r 's, take the square root, and obtain an estimated standard deviation of r_k , that is, the **standard error** of r_k for large lags. A test of the hypothesis that the series is MA(q) could be carried out by comparing r_k to plus and minus two standard errors. We would reject the null hypothesis if and only if r_k lies outside these bounds. In general, we should not expect the sample autocorrelation to mimic the true autocorrelation in great detail. Thus, we should not be surprised to see ripples or “trends” in r_k that have no counterparts in the ρ_k .

6.2 The Partial and Extended Autocorrelation Functions

Since for MA(q) models the autocorrelation function is zero for lags beyond q , the sample autocorrelation is a good indicator of the order of the process. However, the autocorrelations of an AR(p) model do not become zero after a certain number of lags—they die off rather than cut off. So a different function is needed to help determine the order of autoregressive models. Such a function may be defined as the correlation between Y_t and Y_{t-k} after removing the effect of the intervening variables $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots, Y_{t-k+1}$. This coefficient is called the partial autocorrelation at lag k and will be denoted by ϕ_{kk} . (The reason for the seemingly redundant double subscript on ϕ_{kk} will become apparent later on in this section.)

There are several ways to make this definition precise. If $\{Y_t\}$ is a normally distributed time series, we can let

$$\phi_{kk} = Corr(Y_t, Y_{t-k}|Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}) \quad (6.2.1)$$

That is, ϕ_{kk} is the correlation in the bivariate distribution of Y_t and Y_{t-k} conditional on $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}$.

An alternative approach, not based on normality, can be developed in the following way. Consider predicting Y_t based on a linear function of the intervening variables $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}$, say, $\beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_{k-1} Y_{t-k+1}$, with the β 's chosen to minimize the mean square error of prediction. If we assume that the β 's have been so chosen and then think backward in time, it follows from stationarity that the best “predictor” of Y_{t-k} based on the same $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}$ will be $\beta_1 Y_{t-k+1} + \beta_2 Y_{t-k+2} + \dots + \beta_{k-1} Y_{t-1}$. The **partial autocorrelation function** at lag k is then defined to be the correlation between the prediction errors; that is,

$$\begin{aligned}\phi_{kk} = \text{Corr}(Y_t - \beta_1 Y_{t-1} - \beta_2 Y_{t-2} - \dots - \beta_{k-1} Y_{t-k}, \\ Y_{t-k} - \beta_1 Y_{t-k+1} - \beta_2 Y_{t-k+2} - \dots - \beta_{k-1} Y_{t-1})\end{aligned}\quad (6.2.2)$$

(For normally distributed series, it can be shown that the two definitions coincide.) By convention, we take $\phi_{11} = 1$.

As an example, consider ϕ_{22} . It is shown in Appendix F on page 218 that the best linear prediction of Y_t based on Y_{t-1} alone is just $\rho_1 Y_{t-1}$. Thus, according to Equation (6.2.2), we will obtain ϕ_{22} by computing

$$\text{Cov}(Y_t - \rho_1 Y_{t-1}, Y_{t-2} - \rho_1 Y_{t-1}) = \gamma_0(\rho_2 - \rho_1^2 - \rho_1^2 + \rho_1^2) = \gamma_0(1 - \rho_1^2)$$

Since

$$\begin{aligned}\text{Var}(Y_t - \rho_1 Y_{t-1}) &= \text{Var}(Y_{t-2} - \rho_1 Y_{t-1}) \\ &= \gamma_0(1 + \rho_1^2 - 2\rho_1^2) \\ &= \gamma_0(1 - \rho_1^2)\end{aligned}$$

we have that, for any stationary process, the lag 2 partial autocorrelation can be expressed as

$$\phi_{22} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2} \quad (6.2.3)$$

Consider now an AR(1) model. Recall that $\rho_k = \phi^k$ so that

$$\phi_{22} = \frac{\phi^2 - \phi^2}{1 - \phi^2} = 0$$

We shall soon see that for the AR(1) case, $\phi_{kk} = 0$ for all $k > 1$. Thus the partial autocorrelation is nonzero for lag 1, the order of the AR(1) process, but is zero for all lags greater than 1. We shall show this to be generally the case for AR(p) models. Sometimes we say that the partial autocorrelation function for an AR(p) process *cuts off* after the lag exceeds the order of the process.

Consider a general AR(p) case. It will be shown in Chapter 9 that the best linear predictor of Y_t based on a linear function of the variables $Y_{t-1}, Y_{t-2}, \dots, Y_p, \dots, Y_{t-k+1}$ for $k > p$ is $\phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p}$. Also, the best linear predictor of Y_{t-k} is some function of $Y_{t-1}, Y_{t-2}, \dots, Y_p, \dots, Y_{t-k+1}$, call it $h(Y_{t-1}, Y_{t-2}, \dots, Y_p, \dots, Y_{t-k+1})$. So the covariance between the two prediction errors is

$$\begin{aligned}
& \text{Cov}(Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p}, \\
& \quad Y_{t-k} - h(Y_{t-k+1}, Y_{t-k+2}, \dots, Y_{t-1})) \\
& = \text{Cov}(e_t, Y_{t-k} - h(Y_{t-k+1}, Y_{t-k+2}, \dots, Y_{t-1})) \\
& = 0 \text{ since } e_t \text{ is independent of } Y_{t-k}, Y_{t-k+1}, Y_{t-k+2}, \dots, Y_{t-1}
\end{aligned}$$

Thus we have established the key fact that, for an AR(p) model,

$$\phi_{kk} = 0 \text{ for } k > p \quad (6.2.4)$$

For an MA(1) model, Equation (6.2.3) quickly yields

$$\phi_{22} = \frac{-\theta^2}{1 + \theta^2 + \theta^4} \quad (6.2.5)$$

Furthermore, for the MA(1) case, it may be shown that

$$\phi_{kk} = -\frac{\theta^k(1 - \theta^2)}{1 - \theta^{2(k+1)}} \text{ for } k \geq 1 \quad (6.2.6)$$

Notice that the partial autocorrelation of an MA(1) model never equals zero but essentially decays to zero exponentially fast as the lag increases—rather like the autocorrelation function of the AR(1) process. More generally, it can be shown that the partial autocorrelation of an MA(q) model behaves very much like the *autocorrelation* of an AR(q) model.

A general method for finding the partial autocorrelation function for any stationary process with autocorrelation function ρ_k is as follows (see Anderson 1971, pp. 187–188, for example). For a given lag k , it can be shown that the ϕ_{kk} satisfy the Yule-Walker equations (which first appeared in Chapter 4 on page 79):

$$\rho_j = \phi_{k1}\rho_{j-1} + \phi_{k2}\rho_{j-2} + \phi_{k3}\rho_{j-3} + \cdots + \phi_{kk}\rho_{j-k} \quad \text{for } j = 1, 2, \dots, k \quad (6.2.7)$$

More explicitly, we can write these k linear equations as

$$\left. \begin{array}{l}
\phi_{k1} + \rho_1\phi_{k2} + \rho_2\phi_{k3} + \cdots + \rho_{k-1}\phi_{kk} = \rho_1 \\
\rho_1\phi_{k1} + \phi_{k2} + \rho_1\phi_{k3} + \cdots + \rho_{k-2}\phi_{kk} = \rho_2 \\
\vdots \\
\rho_{k-1}\phi_{k1} + \rho_{k-2}\phi_{k2} + \rho_{k-3}\phi_{k3} + \cdots + \phi_{kk} = \rho_k
\end{array} \right\} \quad (6.2.8)$$

Here we are treating $\rho_1, \rho_2, \dots, \rho_k$ as given and wish to solve for $\phi_{k1}, \phi_{k2}, \dots, \phi_{kk}$ (discarding all but ϕ_{kk}).

These equations yield ϕ_{kk} for any stationary process. However, if the process is in fact AR(p), then since for $k = p$ Equations (6.2.8) are just the Yule-Walker equations (page 79), which the AR(p) model is known to satisfy, we must have $\phi_{pp} = \phi_p$. In addition, as we have already seen by an alternative derivation, $\phi_{kk} = 0$ for $k > p$. Thus the partial autocorrelation effectively displays the correct order p of an autoregressive process as the highest lag k before ϕ_{kk} becomes zero.

The Sample Partial Autocorrelation Function

For an observed time series, we need to be able to estimate the partial autocorrelation function at a variety of lags. Given the relationships in Equations (6.2.8), an obvious method is to estimate the ρ 's with sample autocorrelations, the corresponding r 's, and then solve the resulting linear equations for $k = 1, 2, 3, \dots$ to get estimates of ϕ_{kk} . We call the estimated function the **sample partial autocorrelation function** (sample PACF) and denote it by $\hat{\phi}_{kk}$.

Levinson (1947) and Durbin (1960) gave an efficient method for obtaining the solutions to Equations (6.2.8) for either theoretical or sample partial autocorrelations. They showed independently that Equations (6.2.8) can be solved recursively as follows:

$$\hat{\phi}_{kk} = \frac{\rho_k - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_{k-j}}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j} \rho_j} \quad (6.2.9)$$

where

$$\phi_{k,j} = \phi_{k-1,j} - \phi_{kk} \phi_{k-1,k-j} \quad \text{for } j = 1, 2, \dots, k-1$$

For example, using $\phi_{11} = \rho_1$ to get started, we have

$$\hat{\phi}_{22} = \frac{\rho_2 - \phi_{11}\rho_1}{1 - \phi_{11}\rho_1} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}$$

(as before) with $\phi_{21} = \phi_{11} - \hat{\phi}_{22}\phi_{11}$, which is needed for the next step.

Then

$$\hat{\phi}_{33} = \frac{\rho_3 - \phi_{21}\rho_2 - \phi_{22}\rho_1}{1 - \phi_{21}\rho_1 - \phi_{22}\rho_2}$$

We may thus calculate numerically as many values for $\hat{\phi}_{kk}$ as desired. As stated, these recursive equations give us the theoretical partial autocorrelations, but by replacing ρ 's with r 's, we obtain the estimated or sample partial autocorrelations.

To assess the possible magnitude of the sample partial autocorrelations, Quenouille (1949) has shown that, under the hypothesis that an AR(p) model is correct, the sample partial autocorrelations at lags greater than p are approximately normally distributed with zero means and variances $1/n$. Thus, for $k > p$, $\pm 2/\sqrt{n}$ can be used as critical limits on $\hat{\phi}_{kk}$ to test the null hypothesis that an AR(p) model is correct.

Mixed Models and the Extended Autocorrelation Function

Exhibit 6.3 summarizes the behavior of the autocorrelation and partial autocorrelation functions that is useful in specifying models.

Exhibit 6.3 General Behavior of the ACF and PACF for ARMA Models

	AR(p)	MA(q)	ARMA(p, q), $p > 0$, and $q > 0$
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

The Extended Autocorrelation Function

The sample ACF and PACF provide effective tools for identifying pure AR(p) or MA(q) models. However, for a mixed ARMA model, its theoretical ACF and PACF have infinitely many nonzero values, making it difficult to identify mixed models from the sample ACF and PACF. Many graphical tools have been proposed to make it easier to identify the ARMA orders, for example, the corner method (Becuin et al., 1980), the extended autocorrelation (EACF) method (Tsay and Tiao, 1984), and the smallest canonical correlation (SCAN) method (Tsay and Tiao, 1985), among others. We shall outline the EACF method, which seems to have good sampling properties for moderately large sample sizes according to a comparative simulation study done by W. S. Chan (1999).

The EACF method uses the fact that if the AR part of a mixed ARMA model is known, “filtering out” the autoregression from the observed time series results in a pure MA process that enjoys the cutoff property in its ACF. The AR coefficients may be estimated by a finite sequence of regressions. We illustrate the procedure for the case where the true model is an ARMA(1,1) model:

$$Y_t = \phi Y_{t-1} + e_t - \theta e_{t-1}$$

In this case, a simple linear regression of Y_t on Y_{t-1} results in an inconsistent estimator of ϕ , even with infinitely many data. Indeed, the theoretical regression coefficient equals $\rho_1 = (\phi - \theta)(1 - \phi\theta)/(1 - 2\phi\theta + \theta^2)$, not ϕ . But the residuals from this regression do contain information about the error process $\{e_t\}$. A second multiple regression is performed that consists of regressing Y_t on Y_{t-1} and on the lag 1 of the residuals from the first regression. The coefficient of Y_{t-1} in the second regression, denoted by $\tilde{\phi}$, turns out to be a consistent estimator of ϕ . Define $W_t = Y_t - \tilde{\phi}Y_{t-1}$, which is then approximately an MA(1) process. For an ARMA(1,2) model, a third regression that regresses Y_t on its lag 1, the lag 1 of the residuals from the second regression, and the lag 2 of the residuals from the first regression leads to the coefficient of Y_{t-1} being a consistent estimator of ϕ . Similarly, the AR coefficients of an ARMA(p, q) model can be consistently estimated via a sequence of q regressions.

As the AR and MA orders are unknown, an iterative procedure is required. Let

$$W_{t,k,j} = Y_t - \tilde{\phi}_1 Y_{t-1} - \cdots - \tilde{\phi}_k Y_{t-k} \quad (6.2.10)$$

be the autoregressive residuals defined with the AR coefficients estimated iteratively assuming the AR order is k and the MA order is j . The sample autocorrelations of $W_{t,k,j}$ are referred to as the extended sample autocorrelations. For $k = p$ and $j \geq q$, $\{W_{t,k,j}\}$ is approximately an MA(q) model, so that its theoretical autocorrelations of lag $q+1$ or

higher are equal to zero. For $k > p$, an overfitting problem occurs, and this increases the MA order for the W process by the minimum of $k - p$ and $j - q$. Tsay and Tiao (1984) suggested summarizing the information in the sample EACF by a table with the element in the k th row and j th column equal to the symbol X if the lag $j + 1$ sample correlation of $W_{t,k,j}$ is significantly different from 0 (that is, if its magnitude is greater than $1.96/\sqrt{n-j-k}$ since the sample autocorrelation is asymptotically $N(0,1/(n-k-j))$) if the W 's are approximately an $MA(j)$ process) and 0 otherwise. In such a table, an $MA(p,q)$ process will have a theoretical pattern of a triangle of zeroes, with the upper left-hand vertex corresponding to the ARMA orders. Exhibit 6.4 displays the schematic pattern for an $ARMA(1,1)$ model. The upper left-hand vertex of the triangle of zeros is marked with the symbol 0^* and is located in the $p = 1$ row and $q = 1$ column—an indication of an $ARMA(1,1)$ model.

Exhibit 6.4 Theoretical Extended ACF (EACF) for an ARMA(1,1) Model

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	0^*	0	0	0	0	0	0	0	0	0	0	0	0
2	x	x	0	0	0	0	0	0	0	0	0	0	0	0
3	x	x	x	0	0	0	0	0	0	0	0	0	0	0
4	x	x	x	x	0	0	0	0	0	0	0	0	0	0
5	x	x	x	x	x	0	0	0	0	0	0	0	0	0
6	x	x	x	x	x	x	0	0	0	0	0	0	0	0
7	x	x	x	x	x	x	x	0	0	0	0	0	0	0

Of course, the *sample* EACF will never be this clear-cut. Displays like Exhibit 6.4 will contain $8 \times 14 = 112$ different estimated correlations, and some will be statistically significantly different from zero by chance (see Exhibit 6.17 on page 124, for an example). We will illustrate the use of the EACF in the next two sections and throughout the remainder of the book.

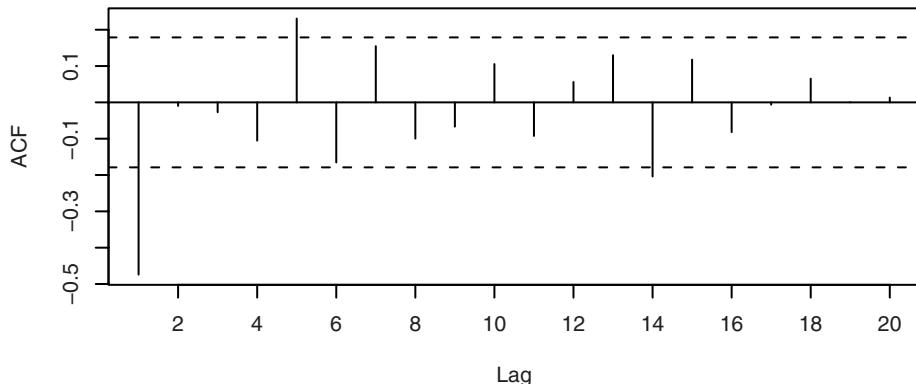
6.3 Specification of Some Simulated Time Series

To illustrate the theory of Sections 6.1 and 6.2, we shall consider the sample autocorrelation and sample partial correlation of some simulated time series.

Exhibit 6.5 displays a graph of the sample autocorrelation out to lag 20 for the simulated time series that we first saw in Exhibit 4.5 on page 61. This series, of length 120, was generated from an $MA(1)$ model with $\theta = 0.9$. From Exhibit 4.1 on page 58, the theoretical autocorrelation at lag 1 is -0.4972 . The estimated or sample value shown at lag 1 on the graph is -0.474 . Using Exhibit 6.2 on page 112, the approximate standard error

of this estimate is $0.71/\sqrt{n} = 0.71/\sqrt{120} = 0.065$, so the estimate is well within two standard errors of the true value.

Exhibit 6.5 Sample Autocorrelation of an MA(1) Process with $\theta = 0.9$

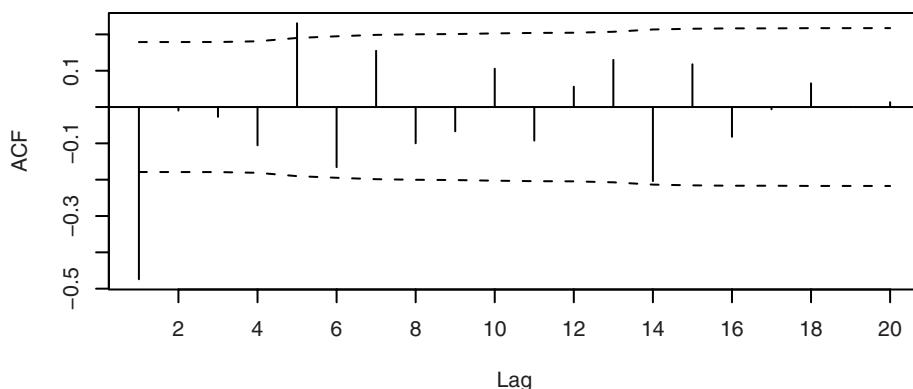


```
> data(ma1.1.s)
> win.graph(width=4.875, height=3, pointsize=8)
> acf(ma1.1.s, xaxp=c(0,20,10))
```

The dashed horizontal lines in Exhibit 6.5, plotted at $\pm 2/\sqrt{n} = \pm 0.1826$, are intended to give critical values for testing whether or not the autocorrelation coefficients are significantly different from zero. These limits are based on the approximate large sample standard error that applies to a white noise process, namely $1/\sqrt{n}$. Notice that the sample ACF values exceed these rough critical values at lags 1, 5, and 14. Of course, the true autocorrelations at lags 5 and 14 are both zero.

Exhibit 6.6 displays the same sample ACF but with critical bounds based on plus and minus two of the more complex standard errors implied by Equation (6.1.11) on page 112. In using Equation (6.1.11), we replace ρ 's by r 's, let q equal 1, 2, 3, ... successively, and take the square root to obtain these standard errors.

Exhibit 6.6 Alternative Bounds for the Sample ACF for the MA(1) Process

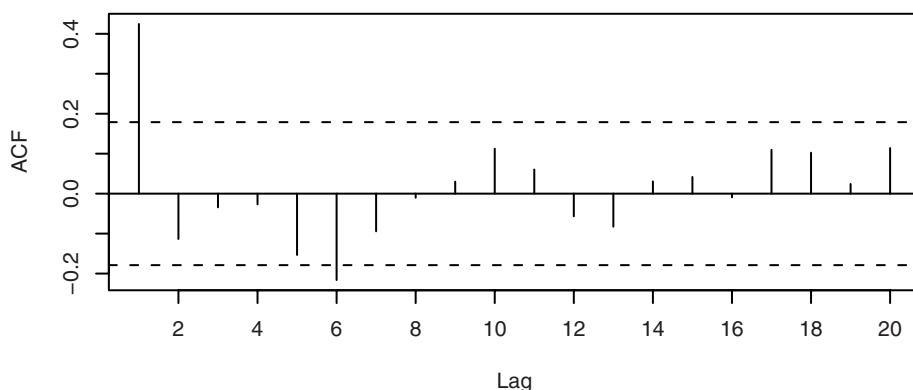


```
> acf(ma1.1.s, ci.type='ma', xaxp=c(0, 20, 10))
```

Now the sample ACF value at lag 14 is insignificant and the one at lag 5 is just barely significant. The lag 1 autocorrelation is still highly significant, and the information given in these two plots taken together leads us to consider an MA(1) model for this series. Remember that the model is tentative at this point and we would certainly want to consider other “nearby” alternative models when we carry out model diagnostics.

As a second example, Exhibit 6.7 shows the sample ACF for the series shown in Exhibit 4.2 on page 59, generated by an MA(1) model with $\theta = -0.9$. The critical values based on the very approximate standard errors point to an MA(1) model for this series also.

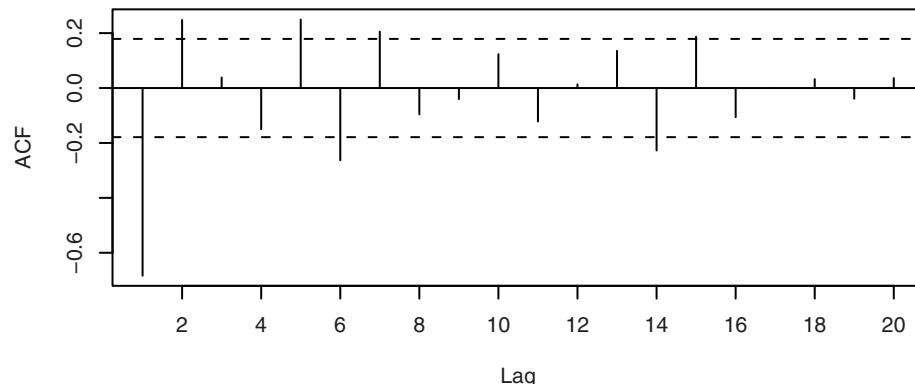
Exhibit 6.7 Sample Autocorrelation for an MA(1) Process with $\theta = -0.9$



```
> data(ma1.2.s); acf(ma1.2.s, xaxp=c(0, 20, 10))
```

For our third example, we use the data shown in Exhibit 4.8 on page 63, which were simulated from an MA(2) model with $\theta_1 = 1$ and $\theta_2 = -0.6$. The sample ACF displays significance at lags 1, 2, 5, 6, 7, and 14 when we use the simple standard error bounds.

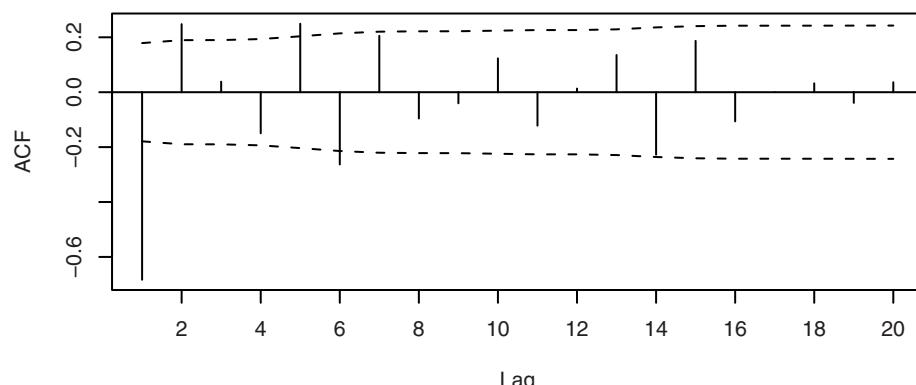
Exhibit 6.8 Sample ACF for an MA(2) Process with $\theta_1 = 1$ and $\theta_2 = -0.6$



```
> data(ma2.s); acf(ma2.s,xaxp=c(0,20,10))
```

Exhibit 6.9 displays the sample ACF with the more sophisticated standard error bounds. Now the lag 2 ACF is no longer significant, and it appears that an MA(1) may be applicable. We will have to wait until we get further along in the model-building process to see that the MA(2) model—the correct one—is the most appropriate model for these data.

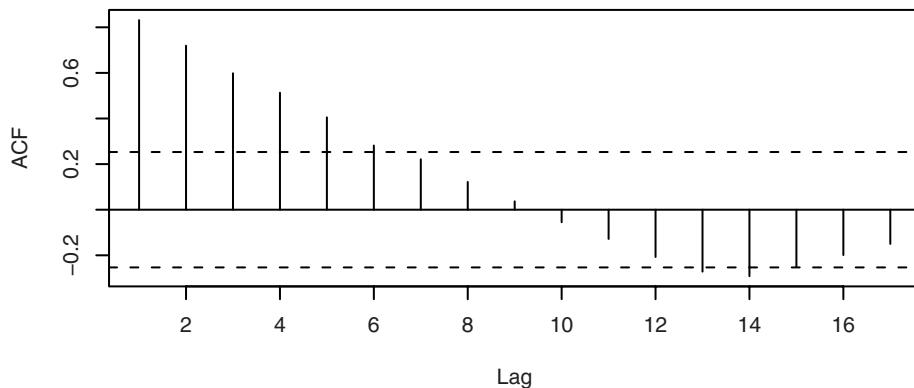
Exhibit 6.9 Alternative Bounds for the Sample ACF for the MA(2) Process



```
> acf(ma2.s,ci.type='ma',xaxp=c(0,20,10))
```

How do these techniques work for autoregressive models? Exhibit 6.10 gives the sample ACF for the simulated AR(1) process we saw in Exhibit 4.13 on page 68. The positive sample ACF values at lags 1, 2, and 3 reflect the strength of the lagged relationships that we saw earlier in Exhibits 4.14, 4.15, and 4.16. However, notice that the sample ACF decreases more linearly than exponentially as theory suggests. Also contrary to theory, the sample ACF goes negative at lag 10 and remains so for many lags.

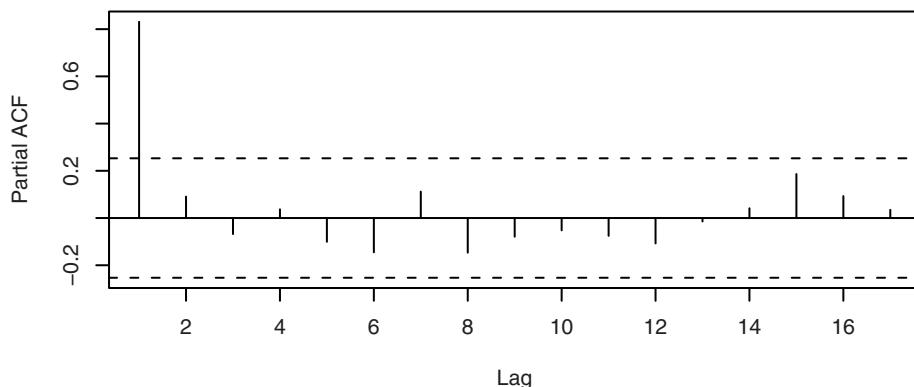
Exhibit 6.10 Sample ACF for an AR(1) Process with $\phi = 0.9$



```
> data(ar1.s); acf(ar1.s,xaxp=c(0,20,10))
```

The sample partial autocorrelation (PACF) shown in Exhibit 6.11, gives a much clearer picture about the nature of the generating model. Based on this graph, we would certainly entertain an AR(1) model for this time series.

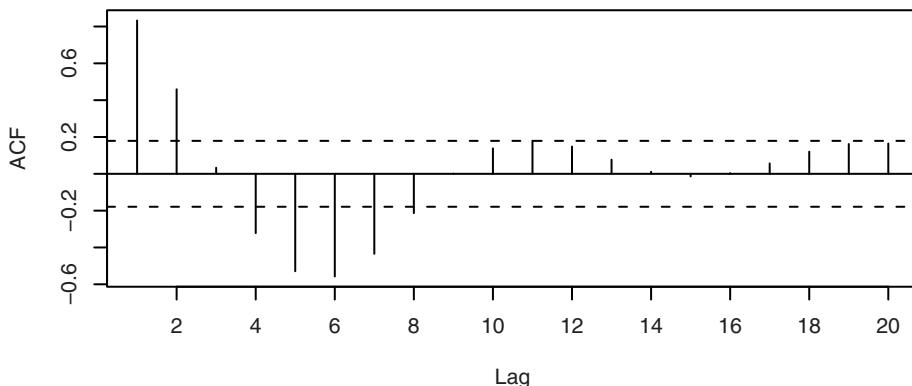
Exhibit 6.11 Sample Partial ACF for an AR(1) Process with $\phi = 0.9$



```
> pacf(ar1.s,xaxp=c(0,20,10))
```

Exhibit 6.12 displays the sample ACF for our AR(2) time series. The time series plot for this series was shown in Exhibit 4.19 on page 74. The sample ACF does look somewhat like the damped wave that Equation (4.3.17) on page 73, and Exhibit 4.18 suggest. However, the sample ACF does not damp down nearly as quickly as theory predicts.

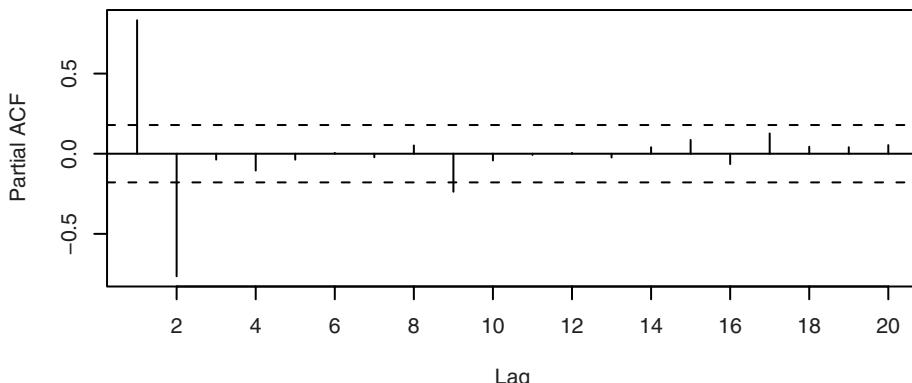
Exhibit 6.12 Sample ACF for an AR(2) Process with $\phi_1 = 1.5$ and $\phi_2 = -0.75$



```
> acf(ar2.s,xaxp=c(0,20,10))
```

The sample PACF in Exhibit 6.13 gives a strong indication that we should consider an AR(2) model for these data. The seemingly significant sample PACF at lag 9 would need to be investigated further during model diagnostics.

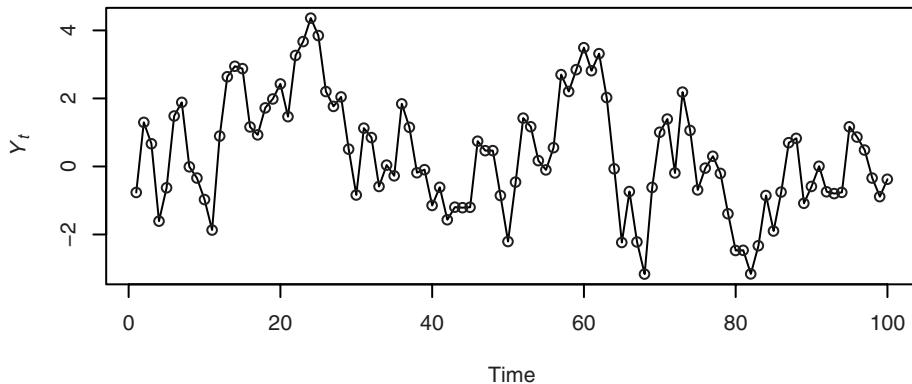
Exhibit 6.13 Sample PACF for an AR(2) Process with $\phi_1 = 1.5$ and $\phi_2 = -0.75$



```
> pacf(ar2.s, xaxp=c(0, 20, 10))
```

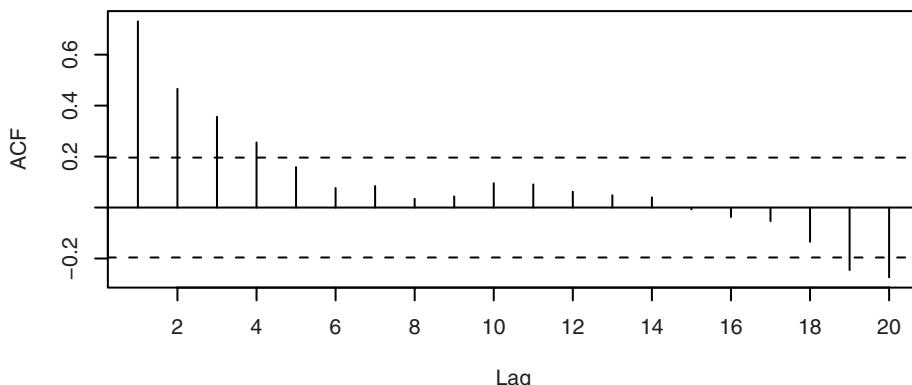
As a final example, we simulated 100 values of a mixed ARMA(1,1) model with $\phi = 0.6$ and $\theta = -0.3$. The time series plot is shown in Exhibit 6.14 and the sample ACF and PACFs are shown in Exhibit 6.15 and Exhibit 6.16, respectively. These seem to indicate that an AR(1) model should be specified.

Exhibit 6.14 Simulated ARMA(1,1) Series with $\phi = 0.6$ and $\theta = -0.3$.



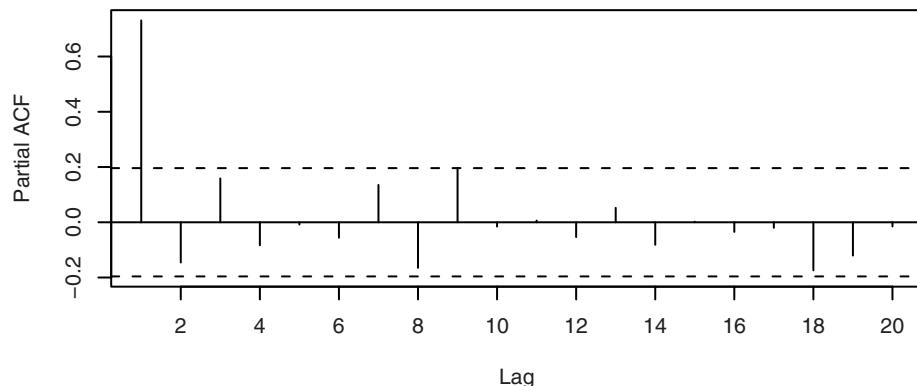
```
> data(armall.s)
> plot(armall.s, type='o', ylab=expression(Y[t]))
```

Exhibit 6.15 Sample ACF for Simulated ARMA(1,1) Series



```
> acf(armall.s, xaxp=c(0, 20, 10))
```

Exhibit 6.16 Sample PACF for Simulated ARMA(1,1) Series



```
> pacf(armall.s, xaxp=c(0,20,10))
```

However, the triangular region of zeros shown in the sample EACF in Exhibit 6.17 indicates quite clearly that a mixed model with $q = 1$ and with $p = 1$ or 2 would be more appropriate. We will illustrate further uses of the EACF when we specify some real series in Section 6.6.

Exhibit 6.17 Sample EACF for Simulated ARMA(1,1) Series

AR / MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	o	o	o	o	o	o	o	o	o	o
1	x	o	o	o	o	o	o	o	o	o	o	o	o	o
2	x	o	o	o	o	o	o	o	o	o	o	o	o	o
3	x	x	o	o	o	o	o	o	o	o	o	o	o	o
4	x	o	x	o	o	o	o	o	o	o	o	o	o	o
5	x	o	o	o	o	o	o	o	o	o	o	o	o	o
6	x	o	o	o	x	o	o	o	o	o	o	o	o	o
7	x	o	o	o	x	o	o	o	o	o	o	o	o	o

```
> eacf(armall.s)
```

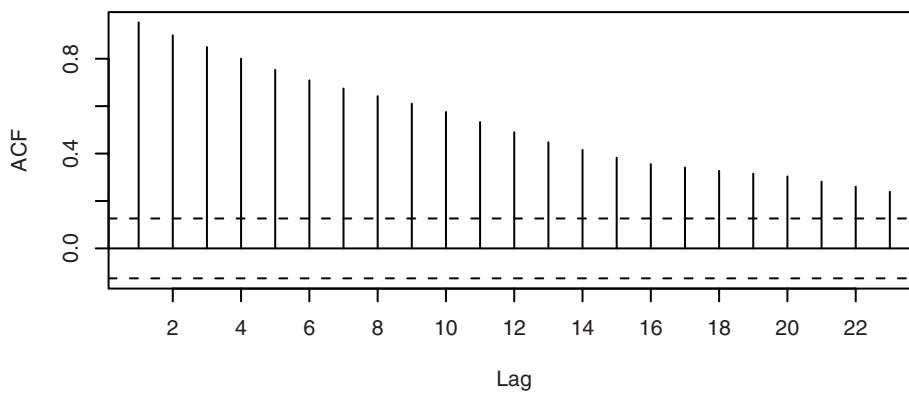
6.4 Nonstationarity

As indicated in Chapter 5, many series exhibit nonstationarity that can be explained by integrated ARMA models. The nonstationarity will frequently be apparent in the time series plot of the series. A review of Exhibits 5.1, 5.5, and 5.8 is recommended here.

The sample ACF computed for nonstationary series will also usually indicate the nonstationarity. The definition of the sample autocorrelation function implicitly *assumes* stationarity; for example, we use lagged products of deviations from the overall mean, and the denominator assumes a constant variance over time. Thus it is not at all clear what the sample ACF is estimating for a nonstationary process. Nevertheless, for nonstationary series, the sample ACF typically fails to die out rapidly as the lags increase. This is due to the tendency for nonstationary series to drift slowly, either up or down, with apparent “trends.” The values of r_k need not be large even for low lags, but often they are.

Consider the oil price time series shown in Exhibit 5.1 on page 88. The sample ACF for the logarithms of these data is displayed in Exhibit 6.18. All values shown are “significantly far from zero,” and the only pattern is perhaps a linear decrease with increasing lag. The sample PACF (not shown) is also indeterminate.

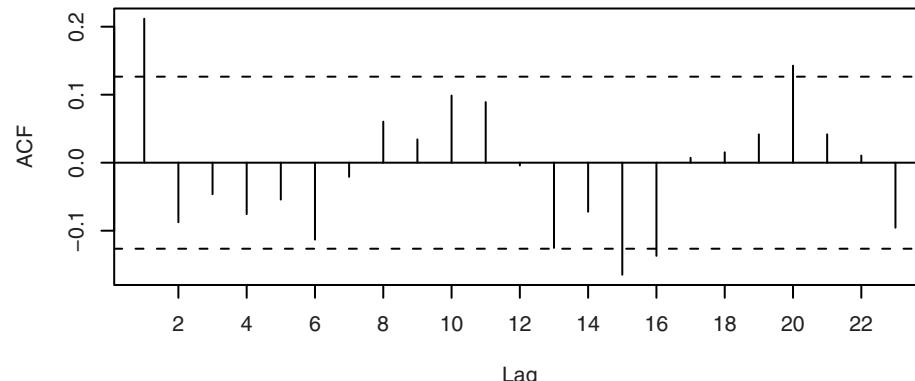
Exhibit 6.18 Sample ACF for the Oil Price Time Series



```
> data(oil.price)
> acf(as.vector(oil.price), xaxp=c(0, 24, 12))
```

The sample ACF computed on the first differences of the logs of the oil price series is shown in Exhibit 6.19. Now the pattern emerges much more clearly—after differencing, a moving average model of order 1 seems appropriate. The model for the original oil price series would then be a nonstationary IMA(1,1) model. (The “significant” ACF at lags 15, 16, and 20 are ignored for now.)

Exhibit 6.19 Sample ACF for the Difference of the Log Oil Price Series



```
> acf(diff(as.vector(log(oil.price))), xaxp=c(0, 24, 12))
```

If the first difference of a series and its sample ACF do not appear to support a stationary ARMA model, then we take another difference and again compute the sample ACF and PACF to look for characteristics of a stationary ARMA process. Usually one or at most two differences, perhaps combined with a logarithm or other transformation, will accomplish this reduction to stationarity. Additional properties of the sample ACF computed on nonstationary data are given in Wichern (1973), Roy (1977), and Hasza (1980). See also Box, Jenkins, and Reinsel (1994, p. 218).

Overdifferencing

From Exercise 2.6 on page 20, we know that the difference of any stationary time series is also stationary. However, overdifferencing introduces unnecessary correlations into a series and will complicate the modeling process.

For example, suppose our observed series, $\{Y_t\}$, is in fact a random walk so that one difference would lead to a very simple white noise model

$$\nabla Y_t = Y_t - Y_{t-1} = e_t$$

However, if we difference once more (that is, overdifference) we have

$$\nabla^2 Y_t = e_t - e_{t-1}$$

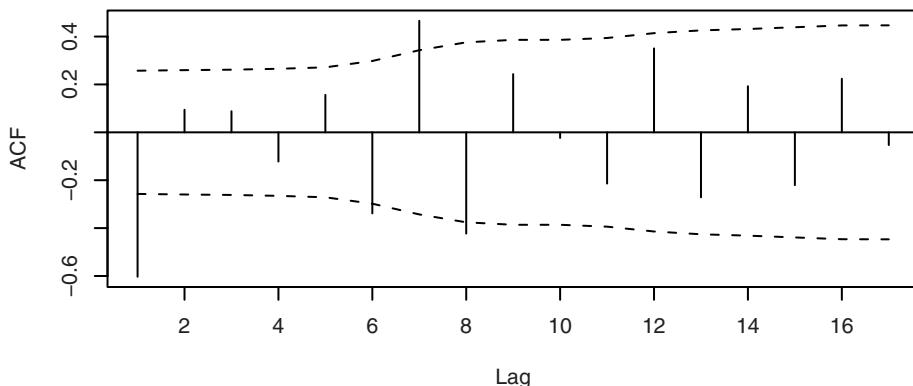
which is an MA(1) model but with $\theta = 1$. If we take two differences in this situation we unnecessarily have to estimate the unknown value of θ . Specifying an IMA(2,1) model would not be appropriate here. The random walk model, which can be thought of as IMA(1,1) with $\theta = 0$, is the correct model.[†] Overdifferencing also creates a noninvert-

[†] The random walk model can also be thought of as an ARI(1,1) with $\phi = 0$ or as a nonstationary AR(1) with $\phi = 1$.

ible model—see Section 4.5 on page 79.[†] Noninvertible models also create serious problems when we attempt to estimate their parameters—see Chapter 7.

To illustrate overdifferencing, consider the random walk shown in Exhibit 2.1 on page 14. Taking one difference should lead to white noise—a very simple model. If we mistakenly take two differences (that is, overdifference) and compute the sample ACF, we obtain the graph shown in Exhibit 6.20. Based on this plot, we would likely specify at least an IMA(2,1) model for the original series and then estimate the unnecessary MA parameter. We also have a significant sample ACF value at lag 7 to think about and deal with.

Exhibit 6.20 Sample ACF of Overdifferenced Random Walk

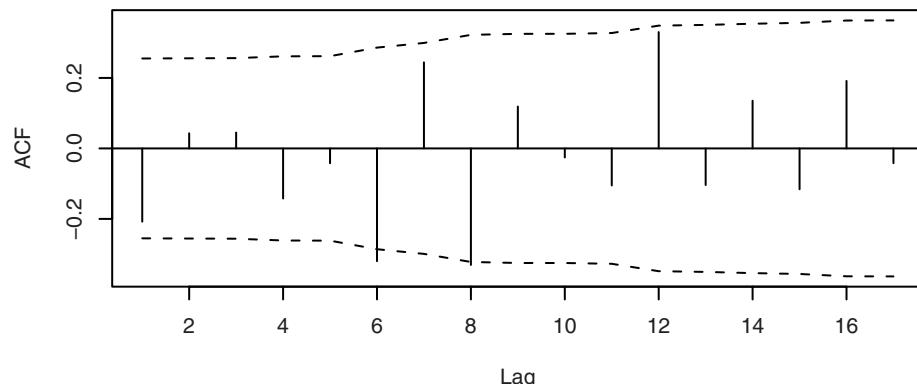


```
> data(rwalk)
> acf(diff(rwalk,difference=2),ci.type='ma', xaxp=c(0,18,9))
```

In contrast, Exhibit 6.21 displays the sample ACF of the *first* difference of the random walk series. Viewing this graph, we would likely want to consider the correct model—the first difference looks very much like white noise.

[†] In backshift notation, if the correct model is $\phi(B)(1-B)Y_t = \theta(B)e_t$, overdifferencing leads to $\phi(B)(1-B)^2Y_t = \theta(B)(1-B)e_t = \theta'(B)e_t$, say, where $\theta'(B) = (1-B)\theta(B)$ and the “forbidden” root in $\theta'(B)$ at $B = 1$ is obvious.

Exhibit 6.21 Sample ACF of Correctly Differenced Random Walk



```
> acf(diff(rwalk), ci.type='ma', xaxp=c(0,18,9))
```

To avoid overdifferencing, we recommend looking carefully at each difference in succession and keeping the principle of parsimony always in mind—*models should be simple, but not too simple*.

The Dickey-Fuller Unit-Root Test

While the approximate linear decay of the sample ACF is often taken as a symptom that the underlying time series is nonstationary and requires differencing, it is also useful to quantify the evidence of nonstationarity in the data-generating mechanism. This can be done via hypothesis testing. Consider the model

$$Y_t = \alpha Y_{t-1} + X_t \text{ for } t = 1, 2, \dots$$

where $\{X_t\}$ is a stationary process. The process $\{Y_t\}$ is nonstationary if the coefficient $\alpha = 1$, but it is stationary if $|\alpha| < 1$. Suppose that $\{X_t\}$ is an AR(k) process: $X_t = \phi_1 X_{t-1} + \dots + \phi_k X_{t-k} + e_t$. Under the null hypothesis that $\alpha = 1$, $X_t = Y_t - Y_{t-1}$. Letting $a = \alpha - 1$, we have

$$\begin{aligned} Y_t - Y_{t-1} &= (\alpha - 1)Y_{t-1} + X_t \\ &= aY_{t-1} + \phi_1 X_{t-1} + \dots + \phi_k X_{t-k} + e_t \\ &= aY_{t-1} + \phi_1(Y_{t-1} - Y_{t-2}) + \dots + \phi_k(Y_{t-k} - Y_{t-k-1}) + e_t \end{aligned} \tag{6.4.1}$$

where $a = 0$ under the hypothesis that Y_t is difference nonstationary. On the other hand, if $\{Y_t\}$ is stationary so that $-1 < \alpha < 1$, then it can be verified that Y_t still satisfies an equation similar to the equation above but with different coefficients; for example, $a = (1 - \phi_1 - \dots - \phi_k)(1 - \alpha) < 0$. Indeed, $\{Y_t\}$ is then an AR($k+1$) process whose AR characteristic equation is given by $\Phi(x)(1 - \alpha x) = 0$, where $\Phi(x) = 1 - \phi_1 x - \dots - \phi_k x^k$. So, the null hypothesis corresponds to the case where the AR characteristic polynomial has a unit root and the alternative hypothesis states that it has no unit roots. *Consequently, the*

test for differencing amounts to testing for a unit root in the AR characteristic polynomial of $\{Y_t\}$.

By the analysis above, the null hypothesis that $\alpha = 1$ (equivalently $a = 0$) can be tested by regressing the first difference of the observed time series on lag 1 of the observed series and on the past k lags of the first difference of the observed series. We then test whether the coefficient $a = 0$ —the null hypothesis being that the process is difference nonstationary. That is, the process is nonstationary but becomes stationary after first differencing. The alternative hypothesis is that $a < 0$ and hence $\{Y_t\}$ is stationary. The augmented Dickey-Fuller (ADF) test statistic is the t -statistic of the estimated coefficient of a from the method of least squares regression. However, the ADF test statistic is not approximately t -distributed under the null hypothesis; instead, it has a certain non-standard large-sample distribution under the null hypothesis of a unit root. Fortunately, percentage points of this limit (null) distribution have been tabulated; see Fuller (1996).

In practice, even after first differencing, the process may not be a finite-order AR process, but it may be closely approximated by some AR process with the AR order increasing with the sample size. Said and Dickey (1984) (see also Chang and Park, 2002) showed that with the AR order increasing with the sample size, the ADF test has the same large-sample null distribution as the case where the first difference of the time series is a finite-order AR process. Often, the approximating AR order can be first estimated based on some information criteria (for example, AIC or BIC) before carrying out the ADF test. See Section 6.5 on page 130 for more information on the AIC and BIC criteria.

In some cases, the process may be trend nonstationary in the sense that it has a deterministic trend (for example, some linear trend) but otherwise is stationary. A unit-root test may be conducted with the aim of discerning difference stationarity from trend stationarity. This can be done by carrying out the ADF test with the detrended data. Equivalently, this can be implemented by regressing the first difference on the covariates defining the trend, the lag 1 of the original data, and the past lags of the first difference of the original data. The t -statistic based on the coefficient estimate of the lag 1 of the original data furnishes the ADF test statistic, which has another nonstandard large-sample null distribution. See Phillips and Xiao (1998) for a survey of unit root testing.

We now illustrate the ADF test with the simulated random walk shown in Exhibit 2.1 on page 14. First, we consider testing the null hypothesis of unit root versus the alternative hypothesis that the time series is stationary with unknown mean. Hence, the regression defined by Equation (6.4.1) is augmented with an intercept to allow for the possibly nonzero mean under the alternative hypothesis. (For the alternative hypothesis that the process is a stationary process of zero mean, the ADF test statistic can be obtained by running the unaugmented regression defined by Equation (6.4.1).) To carry out the test, it remains to determine k .[†] According to the AIC with the first difference of the data, k is found to be 8, in which case the ADF test statistic becomes -0.601 , with the p -value being greater than 0.1 .[‡] On the other hand, setting $k = 0$ (the true order) leads

[†] R code: `ar(diff(rwalk))`

to the ADF statistic -1.738 , with p -value still greater than 0.1 .[†] Thus, there is strong evidence supporting the unit-root hypothesis. Second, recall that the simulated random walk appears to have a linear trend. Hence, linear trend plus stationary error forms another reasonable alternative to the null hypothesis of unit root (difference nonstationarity). For this test, we include both an intercept term and the covariate time in the regression defined by Equation (6.4.1). With $k = 8$, the ADF test statistic equals -2.289 with p -value greater than 0.1 ; that is, we do not reject the null hypothesis of unit root. On the other hand, setting $k = 0$, the true order that is unknown in practice, the ADF test statistic becomes -3.49 with p -value equal to 0.0501 .[‡] Hence, there is weak evidence that the process is linear-trend nonstationary; that is, the process equals linear time trend plus stationary error, contrary to the truth that the process is a random walk, being difference nonstationary! This example shows that with a small sample size, it may be hard to differentiate between trend nonstationarity and difference nonstationarity.

6.5 Other Specification Methods

A number of other approaches to model specification have been proposed since Box and Jenkins' seminal work. One of the most studied is **Akaike's (1973) Information Criterion (AIC)**. This criterion says to select the model that minimizes

$$\text{AIC} = -2\log(\text{maximum likelihood}) + 2k \quad (6.5.1)$$

where $k = p + q + 1$ if the model contains an intercept or constant term and $k = p + q$ otherwise. Maximum likelihood estimation is discussed in Chapter 7. The addition of the term $2(p + q + 1)$ or $2(p + q)$ serves as a “penalty function” to help ensure selection of parsimonious models and to avoid choosing models with too many parameters.

The AIC is an estimator of the average Kullback-Leibler divergence of the estimated model from the true model. Let $p(y_1, y_2, \dots, y_n)$ be the true pdf of Y_1, Y_2, \dots, Y_n , and $q_\theta(y_1, y_2, \dots, y_n)$ be the corresponding pdf under the model with parameter θ . The Kullback-Leibler divergence of q_θ from p is defined by the formula

$$D(p, q_\theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} p(y_1, y_2, \dots, y_n) \log \left[\frac{p(y_1, y_2, \dots, y_n)}{q_\theta(y_1, y_2, \dots, y_n)} \right] dy_1 dy_2 \dots dy_n$$

The AIC estimates $E[D(p, q_{\hat{\theta}})]$, where $\hat{\theta}$ is the maximum likelihood estimator of the vector parameter θ . However, the AIC is a biased estimator, and the bias can be appreciable for large parameter per data ratios. Hurvich and Tsai (1989) showed that the bias can be approximately eliminated by adding another nonstochastic penalty term to the AIC, resulting in the corrected AIC, denoted by AIC_C and defined by the formula

[†] R code: library(uroot); ADF.test(rwalk, selectlags=list
(mode=c(1,2,3,4,5,6,7,8), Pmax=8), itsd=c(1,0,0))

[‡] ADF.test(rwalk, selectlags=list(mode=c(1,2,3,4,5,6,7,8), Pmax=8),
itsd=c(1,1,0))

^{*} ADF.test(rwalk, selectlags=list(Pmax=0), itsd=c(1,1,0))

$$\text{AIC}_c = \text{AIC} + \frac{2(k+1)(k+2)}{n-k-2} \quad (6.5.2)$$

Here n is the (effective) sample size and again k is the total number of parameters as above excluding the noise variance. Simulation results by Hurvich and Tsai (1989) suggest that for cases with k/n greater than 10%, the AIC_c outperforms many other model selection criteria, including both the AIC and BIC.

Another approach to determining the ARMA orders is to select a model that minimizes the Schwarz **Bayesian Information Criterion (BIC)** defined as

$$\text{BIC} = -2\log(\text{maximum likelihood}) + k\log(n) \quad (6.5.3)$$

If the true process follows an $\text{ARMA}(p,q)$ model, then it is known that the orders specified by minimizing the BIC are consistent; that is, they approach the true orders as the sample size increases. However, if the true process is not a finite-order ARMA process, then minimizing AIC among an increasingly large class of ARMA models enjoys the appealing property that it will lead to an optimal ARMA model that is closest to the true process among the class of models under study.[†]

Regardless of whether we use the AIC or BIC, the methods require carrying out maximum likelihood estimation. However, maximum likelihood estimation for an ARMA model is prone to numerical problems due to multimodality of the likelihood function and the problem of overfitting when the AR and MA orders exceed the true orders. Hannan and Rissanen (1982) proposed an interesting and practical solution to this problem. Their procedure consists of first fitting a high-order AR process with the order determined by minimizing the AIC. The second step uses the residuals from the first step as proxies for the unobservable error terms. Thus, an $\text{ARMA}(k,j)$ model can be approximately estimated by regressing the time series on its own lags 1 to k together with the lags 1 to j of the residuals from the high order autoregression; the BIC of this autoregressive model is an estimate of the BIC obtained with maximum likelihood estimation. Hannan and Rissanen (1982) demonstrated that minimizing the approximate BIC still leads to consistent estimation of the ARMA orders.

Order determination is related to the problem of finding the subset of nonzero coefficients of an ARMA model with sufficiently high ARMA orders. A subset $\text{ARMA}(p,q)$ model is an $\text{ARMA}(p,q)$ model with a subset of its coefficients known to be zero. For example, the model

$$Y_t = 0.8Y_{t-12} + e_t + 0.7e_{t-12} \quad (6.5.4)$$

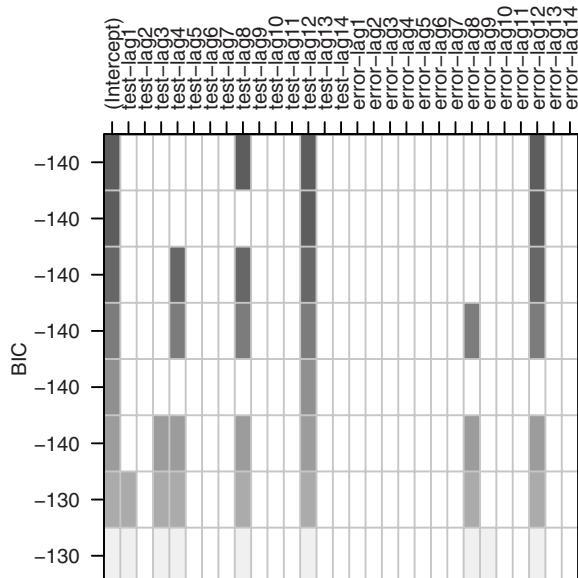
is a subset $\text{ARMA}(12,12)$ model useful for modeling some monthly seasonal time series. For ARMA models of very high orders, such as the preceding $\text{ARMA}(12,12)$ model, finding a subset ARMA model that adequately approximates the underlying process is more important from a practical standpoint than simply determining the ARMA orders. The method of Hannan and Rissanen (1982) for estimating the ARMA orders can be extended to solving the problem of finding an optimal subset ARMA model.

[†] Closeness is measured in terms of the Kullback-Leibler divergence—a measure of disparity between models. See Shibata (1976) and the discussion in Stenseth et al. (2004).

Indeed, several model selection criteria (including AIC and BIC) of the subset ARMA(p, q) models (2^{p+q} of them!) can be approximately, exhaustively, and quickly computed by the method of regression by leaps and bounds (Furnival and Wilson, 1974) applied to the subset regression of Y_t on its own lags and on lags of the residuals from a high-order autoregression of $\{Y_t\}$.

It is prudent to examine a few best subset ARMA models (in terms of, for example, BIC) in order to arrive at some helpful tentative models for further study. The pattern of which lags of the observed time series and which of the error process enter into the various best subset models can be summarized succinctly in a display like that shown in Exhibit 6.22. This table is based on a simulation of the ARMA(12,12) model shown in Equation (6.5.4). Each row in the exhibit corresponds to a subset ARMA model where the cells of the variables selected for the model are shaded. The models are sorted according to their BIC, with better models (lower BIC) placed in higher rows and with darker shades. The top row tells us that the subset ARMA(14,14) model with the smallest BIC contains only lags 8 and 12 of the observed time series and lag 12 of the error process. The next best model contains lag 12 of the time series and lag 8 of the errors, while the third best model contains lags 4, 8, and 12 of the time series and lag 12 of the errors. In our simulated time series, the second best model is the true subset model. However, the BIC values for these three models are all very similar, and all three (plus the fourth best model) are worthy of further study. However, lag 12 of the time series and that of the errors are the two variables most frequently found in the various subset models summarized in the exhibit, suggesting that perhaps they are the more important variables, as we know they are!

Exhibit 6.22 Best Subset ARMA Selection Based on BIC



```
> set.seed(92397)
> test=arima.sim(model=list(ar=c(rep(0,11),.8),
+   ma=c(rep(0,11),0.7)),n=120)
> res=armasubsets(y=test,nar=14,nma=14,y.name='test',
+   ar.method='ols')
> plot(res)
```

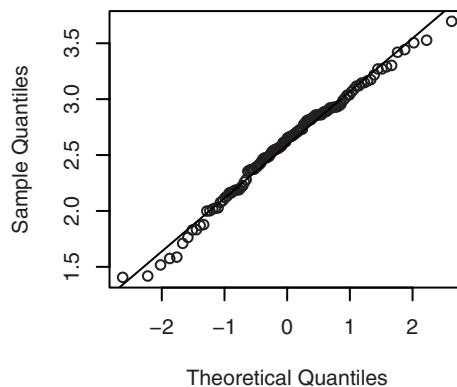
6.6 Specification of Some Actual Time Series

Consider now specification of models for some of the actual time series that we saw in earlier chapters.

The Los Angeles Annual Rainfall Series

Annual total rainfall amounts for Los Angeles were shown in Exhibit 1.1 on page 2. In Chapter 3, we noted in Exhibit 3.17 on page 50, that rainfall amounts were not normally distributed. As is shown in Exhibit 6.23, taking logarithms improves the normality dramatically.

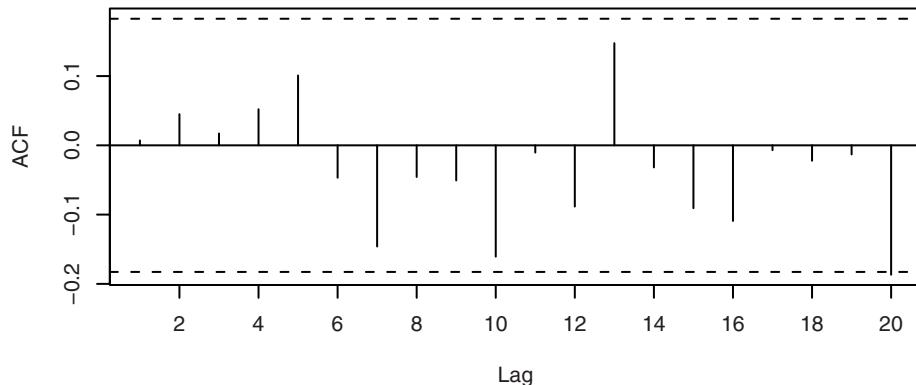
Exhibit 6.23 QQ Normal Plot of the Logarithms of LA Annual Rainfall



```
> data(larain); win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(log(larain)); qqline(log(larain))
```

Exhibit 6.24 displays the sample autocorrelations for the logarithms of the annual rainfall series.

Exhibit 6.24 Sample ACF of the Logarithms of LA Annual Rainfall

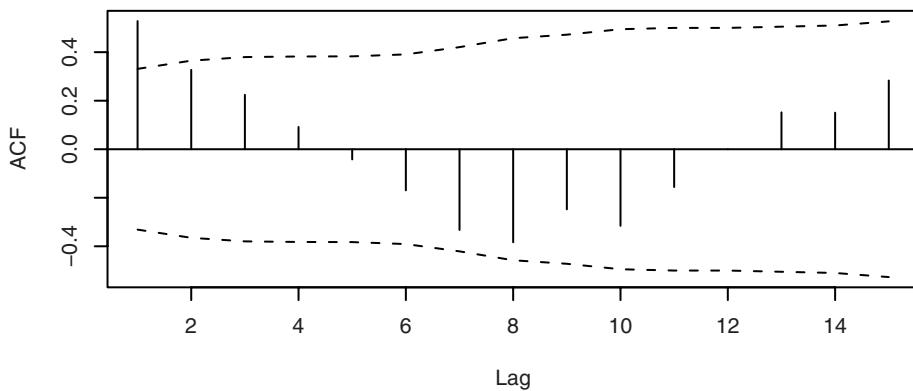


```
> win.graph(width=4.875, height=3, pointsize=8)
> acf(log(larain), xaxp=c(0, 20, 10))
```

The log transformation has improved the normality, but there is no discernable dependence in this time series. We could model the logarithm of annual rainfall amount as independent, normal random variables with mean 2.58 and standard deviation 0.478. Both these values are in units of log(inches).

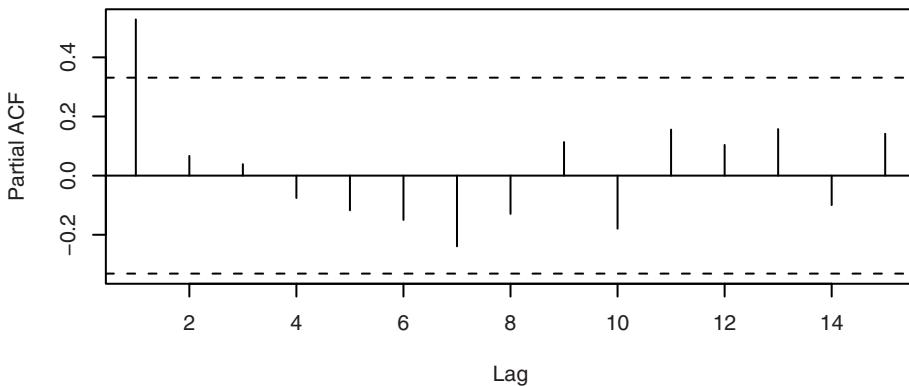
The Chemical Process Color Property Series

The industrial chemical process color property displayed in Exhibit 1.3 on page 3, shows more promise of interesting time series modeling—especially in light of the dependence of successive batches shown in Exhibit 1.4 on page 4. The sample ACF plotted in Exhibit 6.25 might at first glance suggest an MA(1) model, as only the lag 1 autocorrelation is significantly different from zero.

Exhibit 6.25 Sample ACF for the Color Property Series

```
> data(color); acf(color, ci.type='ma')
```

However, the damped sine wave appearance of the plot encourages us to look further at the sample partial autocorrelation. Exhibit 6.26 displays that plot, and now we see clearly that an AR(1) model is worthy of first consideration. As always, our specified models are tentative and subject to modification during the model diagnostics stage of model building.

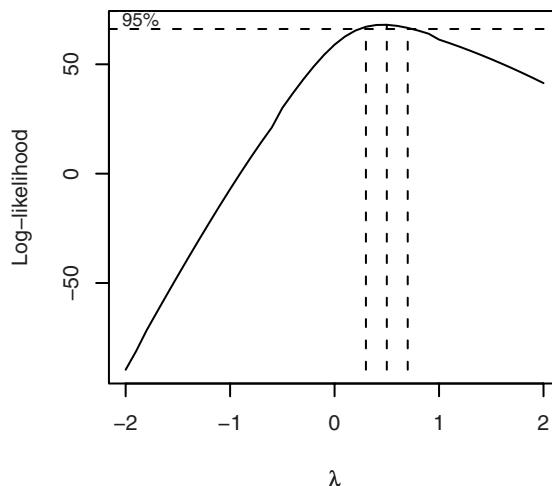
Exhibit 6.26 Sample Partial ACF for the Color Property Series

```
> pacf(color)
```

The Annual Abundance of Canadian Hare Series

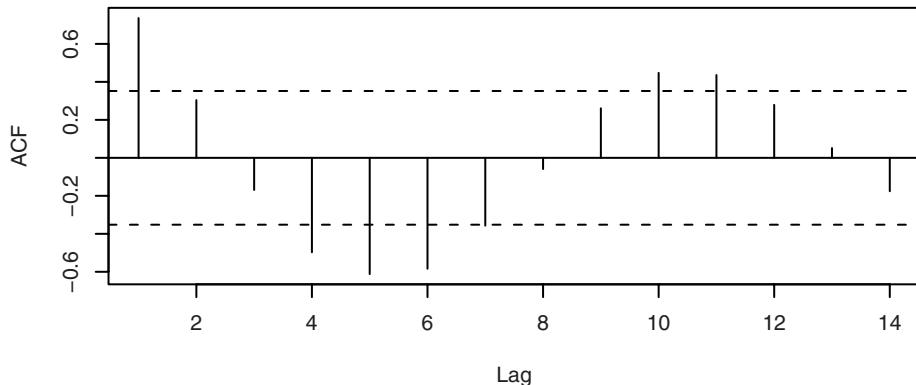
The time series of annual abundance of hare of the Hudson Bay in Canada was displayed in Exhibit 1.5 on page 5, and the year-to-year dependence was demonstrated in Exhibit 1.6. It has been suggested in the literature that a transformation might be used to produce a good model for these data. Exhibit 6.27 displays the log-likelihood as a function of the power parameter, λ . The maximum occurs at $\lambda = 0.4$, but a square root transformation with $\lambda = 0.5$ is well within the confidence interval for λ . We will take the square root of the abundance values for all further analyses.

Exhibit 6.27 Box-Cox Power Transformation Results for Hare Abundance



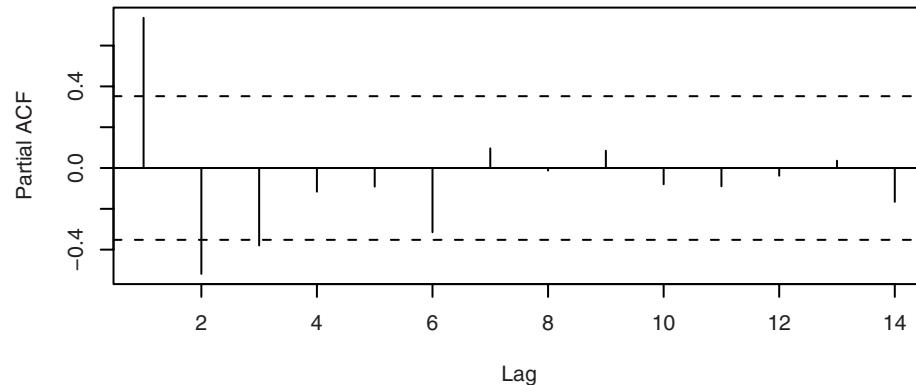
```
> win.graph(width=3,height=3,pointsize=8)
> data(hare); BoxCox.ar(hare)
```

Exhibit 6.28 shows the sample ACF for this transformed series. The fairly strong lag 1 autocorrelation dominates but, again, there is a strong indication of damped oscillatory behavior.

Exhibit 6.28 Sample ACF for Square Root of Hare Abundance

```
> acf(hare^.5)
```

The sample partial autocorrelation for the transformed series is shown in Exhibit 6.29. It gives strong evidence to support an AR(2) or possibly an AR(3) model for these data.

Exhibit 6.29 Sample Partial ACF for Square Root of Hare Abundance

```
> pacf(hare^.5)
```

The Oil Price Series

In Chapter 5, we began to look at the monthly oil price time series and argued graphically that the difference of the logarithms could be considered stationary—see Exhibit 5.1 on page 88. Software implementation of the Augmented Dickey-Fuller unit-root test applied to the logs of the original prices leads to a test statistic of -1.1119 and a p -value of 0.9189 . With stationarity as the alternative hypothesis, this provides strong evidence of nonstationarity and the appropriateness of taking a difference of the logs. For this test, the software chose a value of $k = 6$ in Equation (6.4.1) on page 128 based on large-sample theory.

Exhibit 6.30 shows the summary EACF table for the differences of the logarithms of the oil price data. This table suggests an ARMA model with $p = 0$ and $q = 1$.

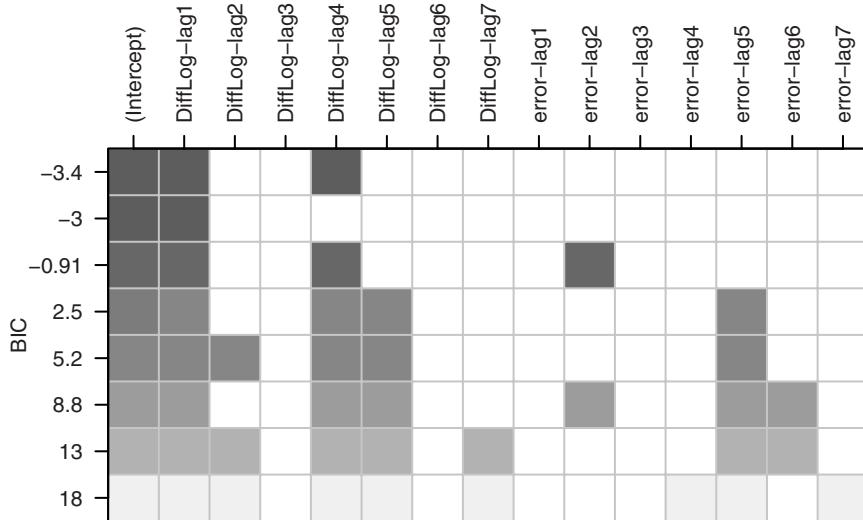
Exhibit 6.30 Extended ACF for Difference of Logarithms of Oil Price Series

AR / MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	o	o	o	o	o	o	o	o	o	o	o	o	o
1	x	x	o	o	o	o	o	o	o	o	x	o	o	o
2	o	x	o	o	o	o	o	o	o	o	o	o	o	o
3	o	x	o	o	o	o	o	o	o	o	o	o	o	o
4	o	x	x	o	o	o	o	o	o	o	o	o	o	o
5	o	x	o	x	o	o	o	o	o	o	o	o	o	o
6	o	x	o	x	o	o	o	o	o	o	o	o	o	o
7	x	x	o	x	o	o	o	o	o	o	o	o	o	o

```
> eacf(diff(log(oil.price)))
```

The results of the best subsets ARMA approach are displayed in Exhibit 6.31.

Exhibit 6.31 Best Subset ARMA Model for Difference of Log(Oil)

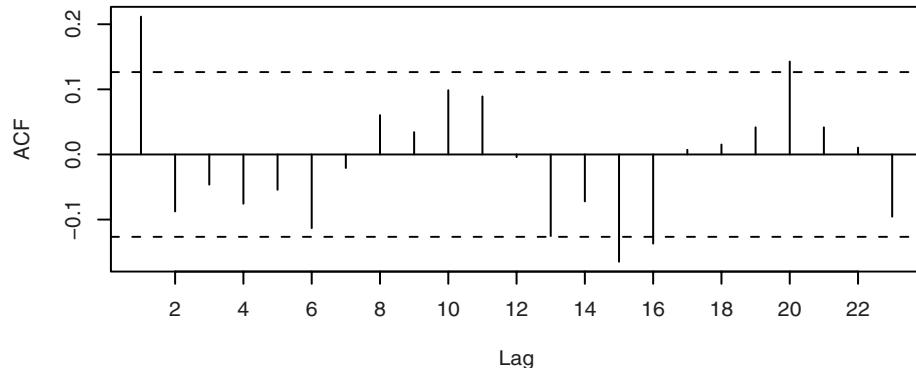


```
> res=armasubsets(y=diff(log(oil.price)),nar=7,nma=7,
+ y.name='test', ar.method='ols')
> plot(res)
```

Here the suggestion is that $Y_t = \nabla \log(\text{Oil}_t)$ should be modeled in terms of Y_{t-1} and Y_{t-4} and that no lags are needed in the error terms. The second best model omits the lag 4 term so that an ARIMA(1,1,0) model on the logarithms should also be investigated further.

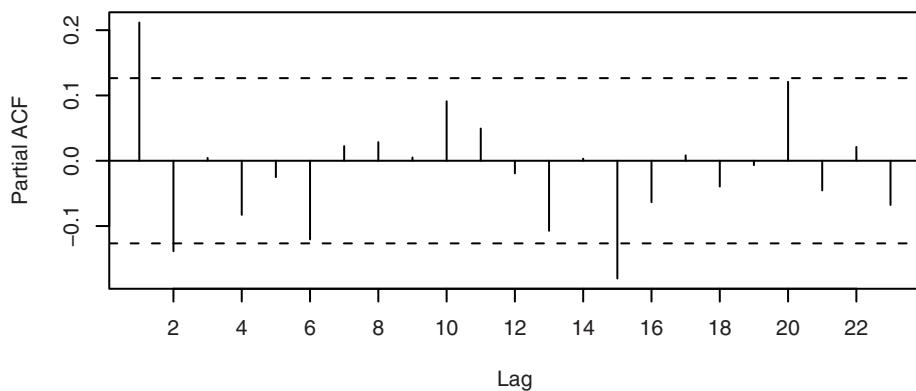
Exhibit 6.32 suggests that we specify an MA(1) model for the difference of the log oil prices, and Exhibit 6.33 says to consider an AR(2) model (ignoring some significant spikes at lags 15, 16, and 20). We will want to look at all of these models further when we estimate parameters and perform diagnostic tests in Chapters 7 and 8. (We will see later that to obtain a suitable model for the oil price series, the outliers in the series will need to be dealt with. (Can you spot the outliers in Exhibit 5.4 on page 91?)

Exhibit 6.32 Sample ACF of Difference of Logged Oil Prices



```
> acf(as.vector(diff(log(oil.price))), xaxp=c(0, 22, 11))
```

Exhibit 6.33 Sample PACF of Difference of Logged Oil Prices



```
> pacf(as.vector(diff(log(oil.price))), xaxp=c(0, 22, 11))
```

6.7 Summary

In this chapter, we considered the problem of specifying reasonable but simple models for observed times series. In particular, we investigated tools for choosing the orders (p , d , and q) for ARIMA(p, d, q) models. Three tools, the sample autocorrelation function, the sample partial autocorrelation function, and the sample extended autocorrelation function, were introduced and studied to help with this difficult task. The Dickey-Fuller unit-root test was also introduced to help distinguish between stationary and nonstationary series. These ideas were all illustrated with both simulated and actual time series.

EXERCISES

- 6.1** Verify Equation (6.1.3) on page 110 for the white noise process.
- 6.2** Verify Equation (6.1.4) on page 110 for the AR(1) process.
- 6.3** Verify the line in Exhibit 6.1 on page 111, for the values $\phi = \pm 0.9$.
- 6.4** Add new entries to Exhibit 6.1 on page 111, for the following values:
 - (a) $\phi = \pm 0.99$.
 - (b) $\phi = \pm 0.5$.
 - (c) $\phi = \pm 0.1$.
- 6.5** Verify Equation (6.1.9) on page 111 and Equation (6.1.10) for the MA(1) process.
- 6.6** Verify the line in Exhibit 6.2 on page 112, for the values $\theta = \pm 0.9$.
- 6.7** Add new entries to Exhibit 6.2 on page 112, for the following values:
 - (a) $\theta = \pm 0.99$.
 - (b) $\theta = \pm 0.8$.
 - (c) $\theta = \pm 0.2$.
- 6.8** Verify Equation (6.1.11) on page 112, for the general MA(q) process.
- 6.9** Use Equation (6.2.3) on page 113, to verify the value for the lag 2 partial autocorrelation function for the MA(1) process given in Equation (6.2.5) on page 114.
- 6.10** Show that the general expression for the partial autocorrelation function of an MA(1) process given in Equation (6.2.6) on page 114, satisfies the Yule-Walker recursion given in Equation (6.2.7).
- 6.11** Use Equation (6.2.8) on page 114, to find the (theoretical) partial autocorrelation function for an AR(2) model in terms of ϕ_1 and ϕ_2 and lag $k = 1, 2, 3, \dots$.
- 6.12** From a time series of 100 observations, we calculate $r_1 = -0.49$, $r_2 = 0.31$, $r_3 = -0.21$, $r_4 = 0.11$, and $|r_k| < 0.09$ for $k > 4$. On this basis alone, what ARIMA model would we tentatively specify for the series?
- 6.13** A stationary time series of length 121 produced sample partial autocorrelation of $\hat{\phi}_{11} = 0.8$, $\hat{\phi}_{22} = -0.6$, $\hat{\phi}_{33} = 0.08$, and $\hat{\phi}_{44} = 0.00$. Based on this information alone, what model would we tentatively specify for the series?
- 6.14** For a series of length 169, we find that $r_1 = 0.41$, $r_2 = 0.32$, $r_3 = 0.26$, $r_4 = 0.21$, and $r_5 = 0.16$. What ARIMA model fits this pattern of autocorrelations?

- 6.15** The sample ACF for a series and its first difference are given in the following table. Here $n = 100$.

lag	1	2	3	4	5	6
ACF for Y_t	0.97	0.97	0.93	0.85	0.80	0.71
ACF for ∇Y_t	-0.42	0.18	-0.02	0.07	-0.10	-0.09

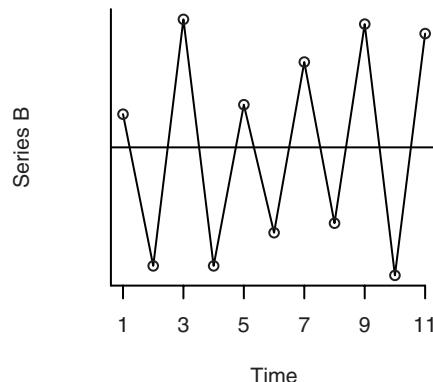
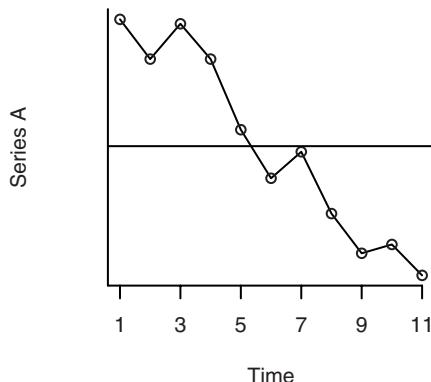
Based on this information alone, which ARIMA model(s) would we consider for the series?

- 6.16** For a series of length 64, the sample partial autocorrelations are given as:

Lag	1	2	3	4	5
PACF	0.47	-0.34	0.20	0.02	-0.06

Which models should we consider in this case?

- 6.17** Consider an AR(1) series of length 100 with $\phi = 0.7$.
- Would you be surprised if $r_1 = 0.6$?
 - Would $r_{10} = -0.15$ be unusual?
- 6.18** Suppose the $\{X_t\}$ is a stationary AR(1) process with parameter ϕ but that we can only observe $Y_t = X_t + N_t$ where $\{N_t\}$ is the white noise measurement error independent of $\{X_t\}$.
- Find the autocorrelation function for the observed process in terms of ϕ , σ_X^2 , and σ_N^2 .
 - Which ARIMA model might we specify for $\{Y_t\}$?
- 6.19** The time plots of two series are shown below.
- For each of the series, describe r_1 using the terms strongly positive, moderately positive, near zero, moderately negative, or strongly negative. Do you need to know the scale of measurement for the series to answer this?
 - Repeat part (a) for r_2 .



- 6.20** Simulate an AR(1) time series with $n = 48$ and with $\phi = 0.7$.
- (a) Calculate the theoretical autocorrelations at lag 1 and lag 5 for this model.
 - (b) Calculate the sample autocorrelations at lag 1 and lag 5 and compare the values with their theoretical values. Use Equations (6.1.5) and (6.1.6) page 111, to quantify the comparisons.
 - (c) Repeat part (b) with a new simulation. Describe how the precision of the estimate varies with different samples selected under identical conditions.
 - (d) If software permits, repeat the simulation of the series and calculation of r_1 and r_5 many times and form the sampling distributions of r_1 and r_5 . Describe how the precision of the estimate varies with different samples selected under identical conditions. How well does the large-sample variance given in Equation (6.1.5) on page 111, approximate the variance in your sampling distribution?
- 6.21** Simulate an MA(1) time series with $n = 60$ and with $\theta = 0.5$.
- (a) Calculate the theoretical autocorrelation at lag 1 for this model.
 - (b) Calculate the sample autocorrelation at lag 1, and compare the value with its theoretical value. Use Exhibit 6.2 on page 112, to quantify the comparisons.
 - (c) Repeat part (b) with a new simulation. Describe how the precision of the estimate varies with different samples selected under identical conditions.
 - (d) If software permits, repeat the simulation of the series and calculation of r_1 many times and form the sampling distribution of r_1 . Describe how the precision of the estimate varies with different samples selected under identical conditions. How well does the large-sample variance given in Exhibit 6.2 on page 112, approximate the variance in your sampling distribution?
- 6.22** Simulate an AR(1) time series with $n = 48$, with
- (a) $\phi = 0.9$, and calculate the theoretical autocorrelations at lag 1 and lag 5;
 - (b) $\phi = 0.6$, and calculate the theoretical autocorrelations at lag 1 and lag 5;
 - (c) $\phi = 0.3$, and calculate the theoretical autocorrelations at lag 1 and lag 5.
 - (d) For each of the series in parts (a), (b), and (c), calculate the sample autocorrelations at lag 1 and lag 5 and compare the values with their theoretical values. Use Equations (6.1.5) and 6.1.6, page 111, to quantify the comparisons. In general, describe how the precision of the estimate varies with the value of ϕ .
- 6.23** Simulate an AR(1) time series with $\phi = 0.6$, with
- (a) $n = 24$, and estimate $\rho_1 = \phi = 0.6$ with r_1 ;
 - (b) $n = 60$, and estimate $\rho_1 = \phi = 0.6$ with r_1 ;
 - (c) $n = 120$, and estimate $\rho_1 = \phi = 0.6$ with r_1 .
 - (d) For each of the series in parts (a), (b), and (c), compare the estimated values with the theoretical value. Use Equation (6.1.5) on page 111, to quantify the comparisons. In general, describe how the precision of the estimate varies with the sample size.

- 6.24** Simulate an MA(1) time series with $\theta = 0.7$, with
- (a) $n = 24$, and estimate ρ_1 with r_1 ;
 - (b) $n = 60$, and estimate ρ_1 with r_1 ;
 - (c) $n = 120$, and estimate ρ_1 with r_1 .
 - (d) For each of the series in parts (a), (b), and (c), compare the estimated values of ρ_1 with the theoretical value. Use Exhibit 6.2 on page 112, to quantify the comparisons. In general, describe how the precision of the estimate varies with the sample size.
- 6.25** Simulate an AR(1) time series of length $n = 36$ with $\phi = 0.7$.
- (a) Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible.
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) What are the theoretical partial autocorrelations for this model?
 - (d) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)? Use the large-sample standard errors reported in Exhibit 6.1 on page 111, to quantify your answer.
 - (e) Calculate and plot the sample PACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (c)? Use the large-sample standard errors reported on page 115 to quantify your answer.
- 6.26** Simulate an MA(1) time series of length $n = 48$ with $\theta = 0.5$.
- (a) What are the theoretical autocorrelations for this model?
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) Calculate and plot the theoretical partial autocorrelation function for this model. Plot sufficient lags until the correlations are negligible. (Hint: See Equation (6.2.6) on page 114.)
 - (d) Calculate and plot the sample PACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (c)?
- 6.27** Simulate an AR(2) time series of length $n = 72$ with $\phi_1 = 0.7$ and $\phi_2 = -0.4$.
- (a) Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible.
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) What are the theoretical partial autocorrelations for this model?
 - (d) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (e) Calculate and plot the sample PACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (c)?

- 6.28** Simulate an MA(2) time series of length $n = 36$ with $\theta_1 = 0.7$ and $\theta_2 = -0.4$.
- (a) What are the theoretical autocorrelations for this model?
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) Calculate and plot the theoretical partial autocorrelation function for this model. Plot sufficient lags until the correlations are negligible. (Hint: See Equation (6.2.6) on page 114.)
 - (d) Calculate and plot the sample PACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (c)?
- 6.29** Simulate a mixed ARMA(1,1) model of length $n = 60$ with $\phi = 0.4$ and $\theta = 0.6$.
- (a) Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible.
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) Calculate and interpret the sample EACF for this series. Does the EACF help you specify the correct orders for the model?
 - (d) Repeat parts (b) and (c) with a new simulation using the same parameter values and sample size.
 - (e) Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 36$.
 - (f) Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 120$.
- 6.30** Simulate a mixed ARMA(1,1) model of length $n = 100$ with $\phi = 0.8$ and $\theta = 0.4$.
- (a) Calculate and plot the theoretical autocorrelation function for this model. Plot sufficient lags until the correlations are negligible.
 - (b) Calculate and plot the sample ACF for your simulated series. How well do the values and patterns match the theoretical ACF from part (a)?
 - (c) Calculate and interpret the sample EACF for this series. Does the EACF help you specify the correct orders for the model?
 - (d) Repeat parts (b) and (c) with a new simulation using the same parameter values and sample size.
 - (e) Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 48$.
 - (f) Repeat parts (b) and (c) with a new simulation using the same parameter values but sample size $n = 200$.

- 6.31** Simulate a nonstationary time series with $n = 60$ according to the model ARIMA(0,1,1) with $\theta = 0.8$.
- (a) Perform the (augmented) Dickey-Fuller test on the series with $k = 0$ in Equation (6.4.1) on page 128. (With $k = 0$, this is the Dickey-Fuller test and is not augmented.) Comment on the results.
 - (b) Perform the augmented Dickey-Fuller test on the series with k chosen by the software—that is, the “best” value for k . Comment on the results.
 - (c) Repeat parts (a) and (b) but use the differences of the simulated series. Comment on the results. (Here, of course, you should reject the unit root hypothesis.)
- 6.32** Simulate a stationary time series of length $n = 36$ according to an AR(1) model with $\phi = 0.95$. This model is stationary, but just barely so. With such a series and a short history, it will be difficult if not impossible to distinguish between stationary and nonstationary with a unit root.
- (a) Plot the series and calculate the sample ACF and PACF and describe what you see.
 - (b) Perform the (augmented) Dickey-Fuller test on the series with $k = 0$ in Equation (6.4.1) on page 128. (With $k = 0$ this is the Dickey-Fuller test and is not augmented.) Comment on the results.
 - (c) Perform the augmented Dickey-Fuller test on the series with k chosen by the software—that is, the “best” value for k . Comment on the results.
 - (d) Repeat parts (a), (b), and (c) but with a new simulation with $n = 100$.
- 6.33** The data file named *deere1* contains 82 consecutive values for the amount of deviation (in 0.000025 inch units) from a specified target value that an industrial machining process at Deere & Co. produced under certain specified operating conditions.
- (a) Display the time series plot of this series and comment on any unusual points.
 - (b) Calculate the sample ACF for this series and comment on the results.
 - (c) Now replace the unusual value by a much more typical value and recalculate the sample ACF. Comment on the change from what you saw in part (b).
 - (d) Calculate the sample PACF based on the revised series that you used in part (c). What model would you specify for the revised series? (Later we will investigate other ways to handle outliers in time series modeling.)
- 6.34** The data file named *deere2* contains 102 consecutive values for the amount of deviation (in 0.0000025 inch units) from a specified target value that another industrial machining process produced at Deere & Co.
- (a) Display the time series plot of this series and comment on its appearance. Would a stationary model seem to be appropriate?
 - (b) Display the sample ACF and PACF for this series and select tentative orders for an ARMA model for the series.

- 6.35** The data file named `deere3` contains 57 consecutive measurements recorded from a complex machine tool at Deere & Co. The values given are deviations from a target value in units of ten millionths of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced.
- (a) Display the time series plot of this series and comment on its appearance. Would a stationary model be appropriate here?
- (b) Display the sample ACF and PACF for this series and select tentative orders for an ARMA model for the series.
- 6.36** The data file named `robot` contains a time series obtained from an industrial robot. The robot was put through a sequence of maneuvers, and the distance from a desired ending point was recorded in inches. This was repeated 324 times to form the time series.
- (a) Display the time series plot of the data. Based on this information, do these data appear to come from a stationary or nonstationary process?
- (b) Calculate and plot the sample ACF and PACF for these data. Based on this additional information, do these data appear to come from a stationary or nonstationary process?
- (c) Calculate and interpret the sample EACF.
- (d) Use the best subsets ARMA approach to specify a model for these data. Compare these results with what you discovered in parts (a), (b), and (c).
- 6.37** Calculate and interpret the sample EACF for the logarithms of the Los Angeles rainfall series. The data are in the file named `larain`. Do the results confirm that the logs are white noise?
- 6.38** Calculate and interpret the sample EACF for the color property time series. The data are in the `color` file. Does the sample EACF suggest the same model that was specified by looking at the sample PACF?
- 6.39** The data file named `days` contains accounting data from the Winegard Co. of Burlington, Iowa. The data are the number of days until Winegard receives payment for 130 consecutive orders from a particular distributor of Winegard products. (The name of the distributor must remain anonymous for confidentiality reasons.)
- (a) Plot the time series, and comment on the display. Are there any unusual values?
- (b) Calculate the sample ACF and PACF for this series.
- (c) Now replace each of the unusual values with a value of 35 days—much more typical values—and repeat the calculation of the sample ACF and PACF. What ARMA model would you specify for this series after removing the outliers? (Later we will investigate other ways to handle outliers in time series modeling.)

CHAPTER 7

PARAMETER ESTIMATION

This chapter deals with the problem of estimating the parameters of an ARIMA model based on the observed time series Y_1, Y_2, \dots, Y_n . We assume that a model has already been specified; that is, we have specified values for p , d , and q using the methods of Chapter 6. With regard to nonstationarity, since the d th difference of the observed series is assumed to be a stationary ARMA(p, q) process, we need only concern ourselves with the problem of estimating the parameters in such stationary models. In practice, then we treat the d th difference of the original time series as the time series from which we estimate the parameters of the complete model. For simplicity, we shall let Y_1, Y_2, \dots, Y_n denote our observed *stationary* process even though it may be an appropriate difference of the original series. We first discuss the method-of-moments estimators, then the least squares estimators, and finally full maximum likelihood estimators.

7.1 The Method of Moments

The method of moments is frequently one of the easiest, if not the most efficient, methods for obtaining parameter estimates. The method consists of equating sample moments to corresponding theoretical moments and solving the resulting equations to obtain estimates of any unknown parameters. The simplest example of the method is to estimate a stationary process mean by a sample mean. The properties of this estimator were studied extensively in Chapter 3.

Autoregressive Models

Consider first the AR(1) case. For this process, we have the simple relationship $\rho_1 = \phi$. In the method of moments, ρ_1 is equated to r_1 , the lag 1 sample autocorrelation. Thus we can estimate ϕ by

$$\hat{\phi} = r_1 \tag{7.1.1}$$

Now consider the AR(2) case. The relationships between the parameters ϕ_1 and ϕ_2 and various moments are given by the Yule-Walker equations (4.3.13) on page 72:

$$\rho_1 = \phi_1 + \phi_1\phi_2 \quad \text{and} \quad \rho_2 = \phi_1\phi_1 + \phi_2$$

The method of moments replaces ρ_1 by r_1 and ρ_2 by r_2 to obtain

$$r_1 = \phi_1 + r_1\phi_2 \quad \text{and} \quad r_2 = r_1\phi_1 + \phi_2$$

which are then solved to obtain

$$\hat{\phi}_1 = \frac{r_1(1 - r_2)}{1 - r_1^2} \text{ and } \hat{\phi}_2 = \frac{r_2 - r_1^2}{1 - r_1^2} \quad (7.1.2)$$

The general AR(p) case proceeds similarly. Replace ρ_k by r_k throughout the Yule-Walker equations on page 79 (or page 114) to obtain

$$\left. \begin{array}{l} \phi_1 + r_1\phi_2 + r_2\phi_3 + \cdots + r_{p-1}\phi_p = r_1 \\ r_1\phi_1 + \phi_2 + r_1\phi_3 + \cdots + r_{p-2}\phi_p = r_2 \\ \vdots \\ r_{p-1}\phi_1 + r_{p-2}\phi_2 + r_{p-3}\phi_3 + \cdots + \phi_p = r_p \end{array} \right\} \quad (7.1.3)$$

These linear equations are then solved for $\hat{\phi}_1, \hat{\phi}_2, \dots, \hat{\phi}_p$. The Durbin-Levinson recursion of Equation (6.2.9) on page 115 provides a convenient method of solution but is subject to substantial round-off errors if the solution is close to the boundary of the stationarity region. The estimates obtained in this way are also called **Yule-Walker estimates**.

Moving Average Models

Surprisingly, the method of moments is not nearly as convenient when applied to moving average models. Consider the simple MA(1) case. From Equations (4.2.2) on page 57, we know that

$$\rho_1 = -\frac{\theta}{1 + \theta^2}$$

Equating ρ_1 to r_1 , we are led to solve a quadratic equation in θ . If $|r_1| < 0.5$, then the two real roots are given by

$$-\frac{1}{2r_1} \pm \sqrt{\frac{1}{4r_1^2} - 1}$$

As can be easily checked, the product of the two solutions is always equal to 1; therefore, only one of the solutions satisfies the invertibility condition $|\theta| < 1$.

After further algebraic manipulation, we see that the invertible solution can be written as

$$\hat{\theta} = \frac{-1 + \sqrt{1 - 4r_1^2}}{2r_1} \quad (7.1.4)$$

If $r_1 = \pm 0.5$, unique, real solutions exist, namely ∓ 1 , but neither is invertible. If $|r_1| > 0.5$ (which is certainly possible even though $|\rho_1| < 0.5$), no real solutions exist, and so the method of moments fails to yield an estimator of θ . Of course, if $|r_1| > 0.5$, the specification of an MA(1) model would be in considerable doubt.

For higher-order MA models, the method of moments quickly gets complicated. We can use Equations (4.2.5) on page 65 and replace ρ_k by r_k for $k = 1, 2, \dots, q$, to obtain q equations in q unknowns $\theta_1, \theta_2, \dots, \theta_q$. The resulting equations are highly nonlinear in the θ 's, however, and their solution would of necessity be numerical. In addition, there will be multiple solutions, of which only one is invertible. We shall not pursue this further since we shall see in Section 7.4 that, for MA models, the method of moments generally produces poor estimates.

Mixed Models

We consider only the ARMA(1,1) case. Recall Equation (4.4.5) on page 78,

$$\rho_k = \frac{(1 - \theta\phi)(\phi - \theta)}{1 - 2\theta\phi + \theta^2} \phi^{k-1} \quad \text{for } k \geq 1$$

Noting that $\rho_2 / \rho_1 = \phi$, we can first estimate ϕ as

$$\hat{\phi} = \frac{r_2}{r_1} \quad (7.1.5)$$

Having done so, we can then use

$$r_1 = \frac{(1 - \theta\hat{\phi})(\hat{\phi} - \theta)}{1 - 2\theta\hat{\phi} + \theta^2} \quad (7.1.6)$$

to solve for $\hat{\theta}$. Note again that a quadratic equation must be solved and only the invertible solution, if any, retained.

Estimates of the Noise Variance

The final parameter to be estimated is the noise variance, σ_e^2 . In all cases, we can first estimate the process variance, $\gamma_0 = \text{Var}(Y_t)$, by the sample variance

$$s^2 = \frac{1}{n-1} \sum_{t=1}^n (Y_t - \bar{Y})^2 \quad (7.1.7)$$

and use known relationships from Chapter 4 among γ_0 , σ_e^2 , and the θ 's and ϕ 's to estimate σ_e^2 .

For the AR(p) models, Equation (4.3.31) on page 77 yields

$$\hat{\sigma}_e^2 = (1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2 - \dots - \hat{\phi}_p r_p) s^2 \quad (7.1.8)$$

In particular, for an AR(1) process,

$$\hat{\sigma}_e^2 = (1 - r_1^2) s^2$$

since $\hat{\phi} = r_1$.

For the MA(q) case, we have, using Equation (4.2.4) on page 65,

$$\hat{\sigma}_e^2 = \frac{s^2}{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \dots + \hat{\theta}_q^2} \quad (7.1.9)$$

For the ARMA(1,1) process, Equation (4.4.4) on page 78 yields

$$\hat{\sigma}_e^2 = \frac{1 - \hat{\phi}^2}{1 - 2\hat{\phi}\hat{\theta} + \hat{\theta}^2}s^2 \quad (7.1.10)$$

Numerical Examples

The table in Exhibit 7.1 displays method-of-moments estimates for the parameters from several simulated time series. Generally speaking, the estimates for all the autoregressive models are fairly good but the estimates for the moving average models are not acceptable. It can be shown that theory confirms this observation—method-of-moments estimators are very inefficient for models containing moving average terms.

Exhibit 7.1 Method-of-Moments Parameter Estimates for Simulated Series

Model	True Parameters			Method-of-Moments Estimates			n
	θ	ϕ_1	ϕ_2	θ	ϕ_1	ϕ_2	
MA(1)	-0.9			-0.554			120
MA(1)	0.9			0.719			120
MA(1)	-0.9			NA [†]			60
MA(1)	0.5			-0.314			60
AR(1)		0.9			0.831		60
AR(1)		0.4			0.470		60
AR(2)		1.5	-0.75		1.472	-0.767	120

[†] No method-of-moments estimate exists since $r_1 = 0.544$ for this simulation.

```
> data(ma1.2.s); data(ma1.1.s); data(ma1.3.s); data(ma1.4.s)
> estimate.ma1.mom(ma1.2.s); estimate.ma1.mom(ma1.1.s)
> estimate.ma1.mom(ma1.3.s); estimate.ma1.mom(ma1.4.s)
> arima(ma1.4.s,order=c(0,0,1),method='CSS',include.mean=F)
> data(ar1.s); data(ar1.2.s)
> ar(ar1.s,order.max=1,AIC=F,method='yw')
> ar(ar1.2.s,order.max=1,AIC=F,method='yw')
> data(ar2.s)
> ar(ar2.s,order.max=2,AIC=F,method='yw')
```

Consider now some actual time series. We start with the Canadian hare abundance series. Since we found in Exhibit 6.27 on page 136 that a square root transformation was appropriate here, we base all modeling on the square root of the original abundance numbers. We illustrate the estimation of an AR(2) model with the hare data, even

though we shall show later that an AR(3) model provides a better fit to the data. The first two sample autocorrelations displayed in Exhibit 6.28 on page 137 are $r_1 = 0.736$ and $r_2 = 0.304$. Using Equations (7.1.2), the method-of-moments estimates of ϕ_1 and ϕ_2 are

$$\hat{\phi}_1 = \frac{r_1(1 - r_2)}{1 - r_1^2} = \frac{0.736(1 - 0.304)}{1 - (0.736)^2} = 1.1178 \quad (7.1.11)$$

and

$$\hat{\phi}_2 = \frac{r_2 - r_1^2}{1 - r_1^2} = \frac{0.304 - (0.736)^2}{1 - (0.736)^2} = -0.519 \quad (7.1.12)$$

The sample mean and variance of this series (after taking the square root) are found to be 5.82 and 5.88, respectively. Then, using Equation (7.1.8), we estimate the noise variance as

$$\begin{aligned}\hat{\sigma}_e^2 &= (1 - \hat{\phi}_1 r_1 - \hat{\phi}_2 r_2)s^2 \\ &= [1 - (1.1178)(0.736) - (-0.519)(0.304)](5.88) \\ &= 1.97\end{aligned} \quad (7.1.13)$$

The estimated model (in original terms) is then

$$\sqrt{Y_t} - 5.82 = 1.1178(\sqrt{Y_{t-1}} - 5.82) - 0.519(\sqrt{Y_{t-2}} - 5.82) + e_t \quad (7.1.14)$$

or

$$\sqrt{Y_t} = 2.335 + 1.1178\sqrt{Y_{t-1}} - 0.519\sqrt{Y_{t-2}} + e_t \quad (7.1.15)$$

with estimated noise variance of 1.97.

Consider now the oil price series. Exhibit 6.32 on page 140 suggested that we specify an MA(1) model for the first differences of the logarithms of the series. The lag 1 sample autocorrelation in that exhibit is 0.212, so the method-of-moments estimate of θ is

$$\hat{\theta} = \frac{-1 + \sqrt{1 - 4(0.212)^2}}{2(0.212)} = -0.222 \quad (7.1.16)$$

The mean of the differences of the logs is 0.004 and the variance is 0.0072. The estimated model is

$$\nabla \log(Y_t) = 0.004 + e_t + 0.222e_{t-1} \quad (7.1.17)$$

or

$$\log(Y_t) = \log(Y_{t-1}) + 0.004 + e_t + 0.222e_{t-1} \quad (7.1.18)$$

with estimated noise variance of

$$\hat{\sigma}_e^2 = \frac{s^2}{1 + \hat{\theta}^2} = \frac{0.0072}{1 + (-0.222)^2} = 0.00686 \quad (7.1.19)$$

Using Equation (3.2.3) on page 28 with estimated parameters yields a standard error of the sample mean of 0.0060. Thus, the observed sample mean of 0.004 is not significantly different from zero and we would remove the constant term from the model, giving a final model of

$$\log(Y_t) = \log(Y_{t-1}) + e_t + 0.222e_{t-1} \quad (7.1.20)$$

7.2 Least Squares Estimation

Because the method of moments is unsatisfactory for many models, we must consider other methods of estimation. We begin with least squares. For autoregressive models, the ideas are quite straightforward. At this point, we introduce a possibly nonzero mean, μ , into our stationary models and treat it as another parameter to be estimated by least squares.

Autoregressive Models

Consider the first-order case where

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + e_t \quad (7.2.1)$$

We can view this as a regression model with predictor variable Y_{t-1} and response variable Y_t . Least squares estimation then proceeds by minimizing the sum of squares of the differences

$$(Y_t - \mu) - \phi(Y_{t-1} - \mu)$$

Since only Y_1, Y_2, \dots, Y_n are observed, we can only sum from $t = 2$ to $t = n$. Let

$$S_c(\phi, \mu) = \sum_{t=2}^n [(Y_t - \mu) - \phi(Y_{t-1} - \mu)]^2 \quad (7.2.2)$$

This is usually called the **conditional sum-of-squares function**. (The reason for the term *conditional* will become apparent later on.) According to the principle of least squares, we estimate ϕ and μ by the respective values that minimize $S_c(\phi, \mu)$ given the observed values of Y_1, Y_2, \dots, Y_n .

Consider the equation $\partial S_c / \partial \mu = 0$. We have

$$\frac{\partial S_c}{\partial \mu} = \sum_{t=2}^n 2[(Y_t - \mu) - \phi(Y_{t-1} - \mu)](-1 + \phi) = 0$$

or, simplifying and solving for μ ,

$$\mu = \frac{1}{(n-1)(1-\phi)} \left[\sum_{t=2}^n Y_t - \phi \sum_{t=2}^n Y_{t-1} \right] \quad (7.2.3)$$

Now, for large n ,

$$\frac{1}{n-1} \sum_{t=2}^n Y_t \approx \frac{1}{n-1} \sum_{t=2}^n Y_{t-1} \approx \bar{Y}$$

Thus, regardless of the value of ϕ , Equation (7.2.3) reduces to

$$\hat{\mu} \approx \frac{1}{1-\phi} (\bar{Y} - \phi \bar{Y}) = \bar{Y} \quad (7.2.4)$$

We sometimes say, except for end effects, $\hat{\mu} = \bar{Y}$.

Consider now the minimization of $S_c(\phi, \bar{Y})$ with respect to ϕ . We have

$$\frac{\partial S_c(\phi, \bar{Y})}{\partial \phi} = \sum_{t=2}^n 2[(Y_t - \bar{Y}) - \phi(Y_{t-1} - \bar{Y})](Y_{t-1} - \bar{Y})$$

Setting this equal to zero and solving for ϕ yields

$$\hat{\phi} = \frac{\sum_{t=2}^n (Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=2}^n (Y_{t-1} - \bar{Y})^2}$$

Except for one term missing in the denominator, namely $(Y_n - \bar{Y})^2$, this is the same as r_1 . The lone missing term is negligible for stationary processes, and thus the least squares and method-of-moments estimators are nearly identical, especially for large samples.

For the general AR(p) process, the methods used to obtain Equations (7.2.3) and (7.2.4) can easily be extended to yield the same result, namely

$$\hat{\mu} = \bar{Y} \quad (7.2.5)$$

To generalize the estimation of the ϕ 's, we consider the second-order model. In accordance with Equation (7.2.5), we replace μ by \bar{Y} in the conditional sum-of-squares function, so

$$S_c(\phi_1, \phi_2, \bar{Y}) = \sum_{t=3}^n [(Y_t - \bar{Y}) - \phi_1(Y_{t-1} - \bar{Y}) - \phi_2(Y_{t-2} - \bar{Y})]^2 \quad (7.2.6)$$

Setting $\partial S_c / \partial \phi_1 = 0$, we have

$$-2 \sum_{t=3}^n [(Y_t - \bar{Y}) - \phi_1(Y_{t-1} - \bar{Y}) - \phi_2(Y_{t-2} - \bar{Y})](Y_{t-1} - \bar{Y}) = 0 \quad (7.2.7)$$

which we can rewrite as

$$\begin{aligned} \sum_{t=3}^n (Y_t - \bar{Y})(Y_{t-1} - \bar{Y}) &= \left(\sum_{t=3}^n (Y_{t-1} - \bar{Y})^2 \right) \phi_1 \\ &\quad + \left(\sum_{t=3}^n (Y_{t-1} - \bar{Y})(Y_{t-2} - \bar{Y}) \right) \phi_2 \end{aligned} \tag{7.2.8}$$

The sum of the lagged products $\sum_{t=3}^n (Y_t - \bar{Y})(Y_{t-1} - \bar{Y})$ is very nearly the numerator of r_1 —we are missing one product, $(Y_2 - \bar{Y})(Y_1 - \bar{Y})$. A similar situation exists for $\sum_{t=3}^n (Y_{t-1} - \bar{Y})(Y_{t-2} - \bar{Y})$, but here we are missing $(Y_n - \bar{Y})(Y_{n-1} - \bar{Y})$. If we divide both sides of Equation (7.2.8) by $\sum_{t=3}^n (Y_t - \bar{Y})^2$, then, except for end effects, which are negligible under the stationarity assumption, we obtain

$$r_1 = \phi_1 + r_1 \phi_2 \tag{7.2.9}$$

Approximating in a similar way with the equation $\partial S_c / \partial \phi_2 = 0$ leads to

$$r_2 = r_1 \phi_1 + \phi_2 \tag{7.2.10}$$

But Equations (7.2.9) and (7.2.10) are just the sample Yule-Walker equations for an AR(2) model.

Entirely analogous results follow for the general stationary AR(p) case: To an excellent approximation, the conditional least squares estimates of the ϕ 's are obtained by solving the sample Yule-Walker equations (7.1.3).[†]

Moving Average Models

Consider now the least-squares estimation of θ in the MA(1) model:

$$Y_t = e_t - \theta e_{t-1} \tag{7.2.11}$$

At first glance, it is not apparent how a least squares or regression method can be applied to such models. However, recall from Equation (4.4.2) on page 77 that invertible MA(1) models can be expressed as

$$Y_t = -\theta Y_{t-1} - \theta^2 Y_{t-2} - \theta^3 Y_{t-3} - \dots + e_t$$

an autoregressive model but of infinite order. Thus least squares can be meaningfully carried out by choosing a value of θ that minimizes

[†] We note that Lai and Wei (1983) established that the conditional least squares estimators are consistent even for nonstationary autoregressive models where the Yule-Walker equations do not apply.

$$S_c(\theta) = \sum(e_t)^2 = \sum[Y_t + \theta Y_{t-1} + \theta^2 Y_{t-2} + \theta^3 Y_{t-3} + \dots]^2 \quad (7.2.12)$$

where, implicitly, $e_t = e_t(\theta)$ is a function of the observed series and the unknown parameter θ .

It is clear from Equation (7.2.12) that the least squares problem is *nonlinear* in the parameters. We will not be able to minimize $S_c(\theta)$ by taking a derivative with respect to θ , setting it to zero, and solving. Thus, even for the simple MA(1) model, we must resort to techniques of numerical optimization. Other problems exist in this case: We have not shown explicit limits on the summation in Equation (7.2.12) nor have we said how to deal with the infinite series under the summation sign.

To address these issues, consider evaluating $S_c(\theta)$ for a *single given value* of θ . The only Y 's we have available are our observed series, Y_1, Y_2, \dots, Y_n . Rewrite Equation (7.2.11) as

$$e_t = Y_t + \theta e_{t-1} \quad (7.2.13)$$

Using this equation, e_1, e_2, \dots, e_n can be calculated recursively if we have the initial value e_0 . A common approximation is to set $e_0 = 0$ —its expected value. Then, *conditional on $e_0 = 0$* , we can obtain

$$\left. \begin{array}{l} e_1 = Y_1 \\ e_2 = Y_2 + \theta e_1 \\ e_3 = Y_3 + \theta e_2 \\ \vdots \\ e_n = Y_n + \theta e_{n-1} \end{array} \right\} \quad (7.2.14)$$

and thus calculate $S_c(\theta) = \sum(e_t)^2$, conditional on $e_0 = 0$, for that single given value of θ .

For the simple case of one parameter, we could carry out a grid search over the invertible range $(-1, +1)$ for θ to find the minimum sum of squares. For more general MA(q) models, a numerical optimization algorithm, such as Gauss-Newton or Nelder-Mead, will be needed.

For higher-order moving average models, the ideas are analogous and no new difficulties arise. We compute $e_t = e_t(\theta_1, \theta_2, \dots, \theta_q)$ recursively from

$$e_t = Y_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \dots + \theta_q e_{t-q} \quad (7.2.15)$$

with $e_0 = e_{-1} = \dots = e_{-q} = 0$. The sum of squares is minimized jointly in $\theta_1, \theta_2, \dots, \theta_q$ using a multivariate numerical method.

Mixed Models

Consider the ARMA(1,1) case

$$Y_t = \phi Y_{t-1} + e_t - \theta e_{t-1} \quad (7.2.16)$$

As in the pure MA case, we consider $e_t = e_t(\phi, \theta)$ and wish to minimize $S_c(\phi, \theta) = \sum e_t^2$. We can rewrite Equation (7.2.16) as

$$e_t = Y_t - \phi Y_{t-1} + \theta e_{t-1} \quad (7.2.17)$$

To obtain e_1 , we now have an additional “startup” problem, namely Y_0 . One approach is to set $Y_0 = 0$ or to \bar{Y} if our model contains a nonzero mean. However, a better approach is to begin the recursion at $t = 2$, thus avoiding Y_0 altogether, and simply minimize

$$S_c(\phi, \theta) = \sum_{t=2}^n e_t^2$$

For the general ARMA(p, q) model, we compute

$$\begin{aligned} e_t &= Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \cdots - \phi_p Y_{t-p} \\ &\quad + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \end{aligned} \quad (7.2.18)$$

with $e_p = e_{p-1} = \cdots = e_{p+1-q} = 0$ and then minimize $S_c(\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q)$ numerically to obtain the conditional least squares estimates of all the parameters.

For parameter sets $\theta_1, \theta_2, \dots, \theta_q$ corresponding to invertible models, the start-up values $e_p, e_{p-1}, \dots, e_{p+1-q}$ will have very little influence on the final estimates of the parameters for large samples.

7.3 Maximum Likelihood and Unconditional Least Squares

For series of moderate length and also for stochastic seasonal models to be discussed in Chapter 10, the start-up values $e_p = e_{p-1} = \cdots = e_{p+1-q} = 0$ will have a more pronounced effect on the final estimates for the parameters. Thus we are led to consider the more difficult problem of maximum likelihood estimation.

The advantage of the method of maximum likelihood is that all of the information in the data is used rather than just the first and second moments, as is the case with least squares. Another advantage is that many large-sample results are known under very general conditions. One disadvantage is that we must for the first time work specifically with the joint probability density function of the process.

Maximum Likelihood Estimation

For any set of observations, Y_1, Y_2, \dots, Y_n , time series or not, the likelihood function L is defined to be the joint probability density of obtaining the data actually observed. However, it is considered as a function of the unknown parameters in the model with the observed data held fixed. For ARIMA models, L will be a function of the ϕ 's, θ 's, μ , and σ_e^2 given the observations Y_1, Y_2, \dots, Y_n . The maximum likelihood estimators are then defined as those values of the parameters for which the data actually observed are *most likely*, that is, the values that maximize the likelihood function.

We begin by looking in detail at the AR(1) model. The most common assumption is that the white noise terms are independent, normally distributed random variables with

zero means and common standard deviation σ_e . The probability density function (pdf) of each e_t is then

$$(2\pi\sigma_e^2)^{-1/2} \exp\left(-\frac{e_t^2}{2\sigma_e^2}\right) \text{ for } -\infty < e_t < \infty$$

and, by independence, the joint pdf for e_2, e_3, \dots, e_n is

$$(2\pi\sigma_e^2)^{-(n-1)/2} \exp\left(-\frac{1}{2\sigma_e^2} \sum_{t=2}^n e_t^2\right) \quad (7.3.1)$$

Now consider

$$\left. \begin{array}{l} Y_2 - \mu = \phi(Y_1 - \mu) + e_2 \\ Y_3 - \mu = \phi(Y_2 - \mu) + e_3 \\ \vdots \\ Y_n - \mu = \phi(Y_{n-1} - \mu) + e_n \end{array} \right\} \quad (7.3.2)$$

If we condition on $Y_1 = y_1$, Equation (7.3.2) defines a linear transformation between e_2, e_3, \dots, e_n and Y_2, Y_3, \dots, Y_n (with Jacobian equal to 1). Thus the joint pdf of Y_2, Y_3, \dots, Y_n given $Y_1 = y_1$ can be obtained by using Equation (7.3.2) to substitute for the e 's in terms of the Y 's in Equation (7.3.1). Thus we get

$$\begin{aligned} f(y_2, y_3, \dots, y_n | y_1) &= (2\pi\sigma_e^2)^{-(n-1)/2} \\ &\times \exp\left\{-\frac{1}{2\sigma_e^2} \sum_{t=2}^n [(y_t - \mu) - \phi(y_{t-1} - \mu)]^2\right\} \end{aligned} \quad (7.3.3)$$

Now consider the (marginal) distribution of Y_1 . It follows from the linear process representation of the AR(1) process (Equation (4.3.8) on page 70) that Y_1 will have a normal distribution with mean μ and variance $\sigma_e^2/(1-\phi^2)$. Multiplying the conditional pdf in Equation (7.3.3) by the marginal pdf of Y_1 gives us the joint pdf of Y_1, Y_2, \dots, Y_n that we require. Interpreted as a function of the parameters ϕ , μ , and σ_e^2 , the likelihood function for an AR(1) model is given by

$$L(\phi, \mu, \sigma_e^2) = (2\pi\sigma_e^2)^{-n/2} (1-\phi^2)^{1/2} \exp\left[-\frac{1}{2\sigma_e^2} S(\phi, \mu)\right] \quad (7.3.4)$$

where

$$S(\phi, \mu) = \sum_{t=2}^n [(Y_t - \mu) - \phi(Y_{t-1} - \mu)]^2 + (1-\phi^2)(Y_1 - \mu) \quad (7.3.5)$$

The function $S(\phi, \mu)$ is called the **unconditional sum-of-squares function**.

As a general rule, the logarithm of the likelihood function is more convenient to

work with than the likelihood itself. For the AR(1) case, the **log-likelihood function**, denoted $\ell(\phi, \mu, \sigma_e^2)$, is given by

$$\ell(\phi, \mu, \sigma_e^2) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma_e^2) + \frac{1}{2} \log(1 - \phi^2) - \frac{1}{2\sigma_e^2} S(\phi, \mu) \quad (7.3.6)$$

For given values of ϕ and μ , $\ell(\phi, \mu, \sigma_e^2)$ can be maximized analytically with respect to σ_e^2 in terms of the yet-to-be-determined estimators of ϕ and μ . We obtain

$$\hat{\sigma}_e^2 = \frac{S(\hat{\phi}, \hat{\mu})}{n} \quad (7.3.7)$$

As in many other similar contexts, we usually divide by $n - 2$ rather than n (since we are estimating *two* parameters, ϕ and μ) to obtain an estimator with less bias. For typical time series sample sizes, there will be very little difference.

Consider now the estimation of ϕ and μ . A comparison of the unconditional sum-of-squares function $S(\phi, \mu)$ with the earlier conditional sum-of-squares function $S_c(\phi, \mu)$ of Equation (7.2.2) on page 154, reveals one simple difference:

$$S(\phi, \mu) = S_c(\phi, \mu) + (1 - \phi^2)(Y_1 - \mu)^2 \quad (7.3.8)$$

Since $S_c(\phi, \mu)$ involves a sum of $n - 1$ components, whereas $(1 - \phi^2)(Y_1 - \mu)^2$ does not involve n , we shall have $S(\phi, \mu) \approx S_c(\phi, \mu)$. Thus the values of ϕ and μ that minimize $S(\phi, \mu)$ or $S_c(\phi, \mu)$ should be very similar, at least for larger sample sizes. The effect of the rightmost term in Equation (7.3.8) will be more substantial when the minimum for ϕ occurs near the stationarity boundary of ± 1 .

Unconditional Least Squares

As a compromise between conditional least squares estimates and full maximum likelihood estimates, we might consider obtaining unconditional least squares estimates; that is, estimates minimizing $S(\phi, \mu)$. Unfortunately, the term $(1 - \phi^2)(Y_1 - \mu)^2$ causes the equations $\partial S / \partial \phi = 0$ and $\partial S / \partial \mu = 0$ to be nonlinear in ϕ and μ , and reparameterization to a constant term $\theta_0 = \mu(1 - \phi)$ does not improve the situation substantially. Thus minimization must be carried out numerically. The resulting estimates are called **unconditional least squares estimates**.

The derivation of the likelihood function for more general ARMA models is considerably more involved. One derivation may be found in Appendix H: State Space Models on page 222. We refer the reader to Brockwell and Davis (1991) or Shumway and Stoffer (2006) for even more details.

7.4 Properties of the Estimates

The large-sample properties of the maximum likelihood and least squares (conditional or unconditional) estimators are identical and can be obtained by modifying standard maximum likelihood theory. Details can be found in Shumway and Stoffer (2006, pp. 125–129). We shall look at the results and their implications for simple ARMA models.

For large n , the estimators are approximately unbiased and normally distributed. The variances and correlations are as follows:

$$\text{AR}(1): \text{Var}(\hat{\phi}) \approx \frac{1 - \phi^2}{n} \quad (7.4.9)$$

$$\text{AR}(2): \begin{cases} \text{Var}(\hat{\phi}_1) \approx \text{Var}(\hat{\phi}_2) \approx \frac{1 - \phi_2^2}{n} \\ \text{Corr}(\hat{\phi}_1, \hat{\phi}_2) \approx -\frac{\phi_1}{1 - \phi_2} = -\rho_1 \end{cases} \quad (7.4.10)$$

$$\text{MA}(1): \text{Var}(\hat{\theta}) \approx \frac{1 - \theta^2}{n} \quad (7.4.11)$$

$$\text{MA}(2): \begin{cases} \text{Var}(\hat{\theta}_1) \approx \text{Var}(\hat{\theta}_2) \approx \frac{1 - \theta_2^2}{n} \\ \text{Corr}(\hat{\theta}_1, \hat{\theta}_2) \approx -\frac{\theta_1}{1 - \theta_2} \end{cases} \quad (7.4.12)$$

$$\text{ARMA}(1,1): \begin{cases} \text{Var}(\hat{\phi}) \approx \left[\frac{1 - \phi^2}{n} \right] \left[\frac{1 - \phi\theta}{\phi - \theta} \right]^2 \\ \text{Var}(\hat{\theta}) \approx \left[\frac{1 - \theta^2}{n} \right] \left[\frac{1 - \phi\theta}{\phi - \theta} \right]^2 \\ \text{Corr}(\hat{\phi}, \hat{\theta}) \approx \frac{\sqrt{(1 - \phi^2)(1 - \theta^2)}}{1 - \phi\theta} \end{cases} \quad (7.4.13)$$

Notice that, in the AR(1) case, the variance of the estimator of ϕ decreases as ϕ approaches ± 1 . Also notice that even though an AR(1) model is a special case of an AR(2) model, the variance of $\hat{\phi}_1$ shown in Equations (7.4.10) shows that our estimation of ϕ_1 will generally suffer if we erroneously fit an AR(2) model when, in fact, $\phi_2 = 0$. Similar comments could be made about fitting an MA(2) model when an MA(1) would suffice or fitting an ARMA(1,1) when an AR(1) or an MA(1) is adequate.

For the ARMA(1,1) case, note the denominator of $\phi - \theta$ in the variances in Equations (7.4.13). If ϕ and θ are nearly equal, the variability in the estimators of ϕ and θ can be extremely large.

Note that in all of the two-parameter models, the estimates can be highly correlated, even for very large sample sizes.

The table shown in Exhibit 7.2 gives numerical values for the large-sample approximate standard deviations of the estimates of ϕ in an AR(1) model for several values of ϕ and several sample sizes. Since the values in the table are equal to $\sqrt{(1 - \phi^2)/n}$, they apply equally well to standard deviations computed according to Equations (7.4.10),

(7.4.11), and (7.4.12).

Thus, in estimating an AR(1) model with, for example, $n = 100$ and $\phi = 0.7$, we can be about 95% confident that our estimate of ϕ is in error by no more than $\pm 2(0.07) = \pm 0.14$.

Exhibit 7.2 AR(1) Model Large-Sample Standard Deviations of $\hat{\phi}$

ϕ	<i>n</i>		
	50	100	200
0.4	0.13	0.09	0.06
0.7	0.10	0.07	0.05
0.9	0.06	0.04	0.03

For stationary autoregressive models, the method of moments yields estimators equivalent to least squares and maximum likelihood, at least for large samples. For models containing moving average terms, such is not the case. For an MA(1) model, it can be shown that the large-sample variance of the method-of-moments estimator of θ is equal to

$$Var(\hat{\theta}) \approx \frac{1 + \theta^2 + 4\theta^4 + \theta^6 + \theta^8}{n(1 - \theta^2)^2} \quad (7.4.14)$$

Comparing Equation (7.4.14) with that of Equation (7.4.11), we see that the variance for the method-of-moments estimator is always larger than the variance of the maximum likelihood estimator. The table in Exhibit 7.3 displays the ratio of the large-sample standard deviations for the two methods for several values of θ . For example, if θ is 0.5, the method-of-moments estimator has a large-sample standard deviation that is 42% larger than the standard deviation of the estimator obtained using maximum likelihood. It is clear from these ratios that the method-of-moments estimator should not be used for the MA(1) model. This same advice applies to all models that contain moving average terms.

Exhibit 7.3 Method of Moments (MM) vs. Maximum Likelihood (MLE) in MA(1) Models

θ	SD_{MM}/SD_{MLE}
0.25	1.07
0.50	1.42
0.75	2.66
0.90	5.33

7.5 Illustrations of Parameter Estimation

Consider the simulated MA(1) series with $\theta = -0.9$. The series was displayed in Exhibit 4.2 on page 59, and we found the method-of-moments estimate of θ to be a rather poor -0.554 ; see Exhibit 7.1 on page 152. In contrast, the maximum likelihood estimate is -0.915 , the unconditional sum-of-squares estimate is -0.923 , and the conditional least squares estimate is -0.879 . For this series, the maximum likelihood estimate of -0.915 is closest to the true value used in the simulation. Using Equation (7.4.11) on page 161 and replacing θ by its estimate, we have a standard error of about

$$\sqrt{Va\hat{r}(\hat{\theta})} \approx \sqrt{\frac{1 - \hat{\theta}^2}{n}} = \sqrt{\frac{1 - (0.91)^2}{120}} \approx 0.04$$

so none of the maximum likelihood, conditional sum-of-squares, or unconditional sum-of-squares estimates are significantly far from the true value of -0.9 .

The second MA(1) simulation with $\theta = 0.9$ produced the method-of-moments estimate of 0.719 shown in Exhibit 7.1. The conditional sum-of-squares estimate is 0.958 , the unconditional sum-of-squares estimate is 0.983 , and the maximum likelihood estimate is 1.000 . These all have a standard error of about 0.04 as above. Here the maximum likelihood estimate of $\hat{\theta} = 1$ is a little disconcerting since it corresponds to a noninvertible model.

The third MA(1) simulation with $\theta = -0.9$ produced a method-of-moments estimate of -0.719 (see Exhibit 7.1). The maximum likelihood estimate here is -0.894 with a standard error of about

$$\sqrt{Va\hat{r}(\hat{\theta})} \approx \sqrt{\frac{1 - (0.894)^2}{60}} \approx 0.06$$

For these data, the conditional sum-of-squares estimate is -0.979 and the unconditional sum-of-squares estimate is -0.961 . Of course, with a standard error of this magnitude, it is unwise to report digits in the estimates of θ beyond the tenths place.

For our simulated autoregressive models, the results are reported in Exhibits 7.4 and 7.5.

Exhibit 7.4 Parameter Estimation for Simulated AR(1) Models

Parameter ϕ	Method-of-Moments Estimate	Conditional SS Estimate	Unconditional SS Estimate	Maximum Likelihood Estimate	n
0.9	0.831	0.857	0.911	0.892	60
0.4	0.470	0.473	0.473	0.465	60

```
> data(ar1.s); data(ar1.2.s)
> ar(ar1.s,order.max=1,AIC=F,method='yw')
> ar(ar1.s,order.max=1,AIC=F,method='ols')
> ar(ar1.s,order.max=1,AIC=F,method='mle')
```

```
> ar(ar1.2.s,order.max=1,AIC=F,method='yw')
> ar(ar1.2.s,order.max=1,AIC=F,method='ols')
> ar(ar1.2.s,order.max=1,AIC=F,method='mle')
```

From Equation (7.4.9) on page 161, the standard errors for the estimates are

$$\sqrt{\text{Var}(\hat{\phi})} \approx \sqrt{\frac{1 - \hat{\phi}^2}{n}} = \sqrt{\frac{1 - (0.831)^2}{60}} \approx 0.07$$

and

$$\sqrt{\text{Var}(\hat{\phi})} = \sqrt{\frac{1 - (0.470)^2}{60}} \approx 0.11$$

respectively. Considering the magnitude of these standard errors, all four methods estimate reasonably well for AR(1) models.

Exhibit 7.5 Parameter Estimation for a Simulated AR(2) Model

Parameters	Method-of-Moments Estimates	Conditional SS Estimates	Unconditional SS Estimates	Maximum Likelihood Estimate	<i>n</i>
$\phi_1 = 1.5$	1.472	1.5137	1.5183	1.5061	120
$\phi_2 = -0.75$	-0.767	-0.8050	-0.8093	-0.7965	120

```
> data(ar2.s)
> ar(ar2.s,order.max=2,AIC=F,method='yw')
> ar(ar2.s,order.max=2,AIC=F,method='ols')
> ar(ar2.s,order.max=2,AIC=F,method='mle')
```

From Equation (7.4.10) on page 161, the standard errors for the estimates are

$$\sqrt{\text{Var}(\hat{\phi}_1)} \approx \sqrt{\text{Var}(\hat{\phi}_2)} \approx \sqrt{\frac{1 - \hat{\phi}_2^2}{n}} = \sqrt{\frac{1 - (0.75)^2}{120}} \approx 0.06$$

Again, considering the size of the standard errors, all four methods estimate reasonably well for AR(2) models.

As a final example using simulated data, consider the ARMA(1,1) shown in Exhibit 6.14 on page 123. Here $\phi = 0.6$, $\theta = -0.3$, and $n = 100$. Estimates using the various methods are shown in Exhibit 7.6.

Exhibit 7.6 Parameter Estimation for a Simulated ARMA(1,1) Model

Parameters	Method-of-Moments Estimates	Conditional SS Estimates	Unconditional SS Estimates	Maximum Likelihood Estimate	<i>n</i>
$\phi = 0.6$	0.637	0.5586	0.5691	0.5647	100
$\theta = -0.3$	-0.2066	-0.3669	-0.3618	-0.3557	100

```
> data(arma11.s)
> arima(arma11.s, order=c(1, 0, 1), method='CSS')
> arima(arma11.s, order=c(1, 0, 1), method='ML')
```

Now let's look at some real time series. The industrial chemical property time series was first shown in Exhibit 1.3 on page 3. The sample PACF displayed in Exhibit 6.26 on page 135, strongly suggested an AR(1) model for this series. Exhibit 7.7 shows the various estimates of the ϕ parameter using four different methods of estimation.

Exhibit 7.7 Parameter Estimation for the Color Property Series

Parameter	Method-of-Moments Estimate	Conditional SS Estimate	Unconditional SS Estimate	Maximum Likelihood Estimate	<i>n</i>
ϕ	0.5282	0.5549	0.5890	0.5703	35

```
> data(color)
> ar(color, order.max=1, AIC=F, method='yw')
> ar(color, order.max=1, AIC=F, method='ols')
> ar(color, order.max=1, AIC=F, method='mle')
```

Here the standard error of the estimates is about

$$\sqrt{\hat{Var}(\hat{\phi})} \approx \sqrt{\frac{1-(0.57)^2}{35}} \approx 0.14$$

so all of the estimates are comparable.

As a second example, consider again the Canadian hare abundance series. As before, we base all modeling on the square root of the original abundance numbers. Based on the partial autocorrelation function shown in Exhibit 6.29 on page 137, we will estimate an AR(3) model. For this illustration, we use maximum likelihood estimation and show the results obtained from the R software in Exhibit 7.8.

Exhibit 7.8 Maximum Likelihood Estimates from R Software: Hare Series

Coefficients:	ar1	ar2	ar3	Intercept [†]
	1.0519	-0.2292	-0.3931	5.6923
s.e.	0.1877	0.2942	0.1915	0.3371

sigma^2 estimated as 1.066: log-likelihood = -46.54, AIC = 101.08

[†] The intercept here is the estimate of the process mean μ —not of θ_0 .

```
> data(hare)
> arima(sqrt(hare), order=c(3, 0, 0))
```

Here we see that $\hat{\phi}_1 = 1.0519$, $\hat{\phi}_2 = -0.2292$, and $\hat{\phi}_3 = -0.3930$. We also see that the estimated noise variance is $\hat{\sigma}_e^2 = 1.066$. Noting the standard errors, the estimates of the lag 1 and lag 3 autoregressive coefficients are significantly different from zero, as is the intercept term, but the lag 2 autoregressive parameter estimate is not significant.

The estimated model would be written

$$\sqrt{Y_t} - 5.6923 = 1.0519(\sqrt{Y_{t-1}} - 5.6923) - 0.2292(\sqrt{Y_{t-2}} - 5.6923) \\ - 0.3930(\sqrt{Y_{t-3}} - 5.6923) + e_t$$

or

$$\sqrt{Y_t} = 3.25 + 1.0519\sqrt{Y_{t-1}} - 0.2292\sqrt{Y_{t-2}} - 0.3930\sqrt{Y_{t-3}} + e_t$$

where Y_t is the hare abundance in year t in original terms. Since the lag 2 autoregressive term is insignificant, we might drop that term (that is, set $\phi_2 = 0$) and obtain new estimates of ϕ_1 and ϕ_3 with this subset model.

As a last example, we return to the oil price series. The sample ACF shown in Exhibit 6.32 on page 140, suggested an MA(1) model on the differences of the logs of the prices. Exhibit 7.9 gives the estimates of θ by the various methods and, as we have seen earlier, the method-of-moments estimate differs quite a bit from the others. The others are nearly equal given their standard errors of about 0.07.

Exhibit 7.9 Estimation for the Difference of Logs of the Oil Price Series

Parameter	Method-of-Moments Estimate	Conditional SS Estimate	Unconditional SS Estimate	Maximum Likelihood Estimate	n
θ	-0.2225	-0.2731	-0.2954	-0.2956	241

```
> data(oil.price)
> arima(log(oil.price), order=c(0, 1, 1), method='CSS')
> arima(log(oil.price), order=c(0, 1, 1), method='ML')
```

7.6 Bootstrapping ARIMA Models

In Section 7.4, we summarized some approximate normal distribution results for the estimator $\hat{\gamma}$, where γ is the vector consisting of all the ARMA parameters. These normal approximations are accurate for large samples, and statistical software generally uses those results in calculating and reporting standard errors. The standard error of some complex function of the model parameters, for example the quasi-period of the model, if it exists, is then usually obtained by the delta method. However, the general theory provides no practical guidance on how large the sample size should be for the normal approximation to be reliable. Bootstrap methods (Efron and Tibshirani, 1993; Davison and Hinkley, 2003) provide an alternative approach to assessing the uncertainty of an estimator and may be more accurate for small samples. There are several variants of the bootstrap method for dependent data—see Politis (2003). We shall confine our discussion to the parametric bootstrap that generates the bootstrap time series $Y_1^*, Y_2^*, \dots, Y_n^*$ by simulation from the fitted ARIMA(p, d, q) model. (The bootstrap may be done by fixing the first $p + d$ initial values of Y^* to those of the observed data. For stationary models, an alternative procedure is to simulate stationary realizations from the fitted model, which can be done approximately by simulating a long time series from the fitted model and then deleting the transient initial segment of the simulated data—the so-called burn-in.) If the errors are assumed to be normally distributed, the errors may be drawn randomly and with replacement from $N(0, \hat{\sigma}_e^2)$. For the case of an unknown error distribution, the errors can be drawn randomly and with replacement from the residuals of the fitted model. For each bootstrap series, let $\hat{\gamma}^*$ be the estimator computed based on the bootstrap time series data using the method of full maximum likelihood estimation assuming stationarity. (Other estimation methods may be used.) The bootstrap is replicated, say, B times. (For example, $B = 1000$.) From the B bootstrap parameter estimates, we can form an empirical distribution and use it to calibrate the uncertainty in $\hat{\gamma}$. Suppose we are interested in estimating some function of γ , say $h(\gamma)$ —for example, the AR(1) coefficient. Using the percentile method, a 95% bootstrap confidence interval for $h(\gamma)$ can be obtained as the interval from the 2.5 percentile to the 97.5 percentile of the bootstrap distribution of $h(\hat{\gamma}^*)$.

We illustrate the bootstrap method with the hare data. The bootstrap 95% confidence intervals reported in the first row of the table in Exhibit 7.10 are based on the bootstrap obtained by conditioning on the initial three observations and assuming normal errors. Those in the second row are obtained using the same method except that the errors are drawn from the residuals. The third and fourth rows report the confidence intervals based on the stationary bootstrap with a normal error distribution for the third row and the empirical residual distribution for the fourth row. The fifth row in the table shows the theoretical 95% confidence intervals based on the large-sample distribution results for the estimators. In particular, the bootstrap time series for the first bootstrap method is generated recursively using the equation

$$Y_t^* - \hat{\phi}_1 Y_{t-1}^* - \hat{\phi}_2 Y_{t-2}^* - \hat{\phi}_3 Y_{t-3}^* = \hat{\theta}_0 + e_t^* \quad (7.6.1)$$

for $t = 4, 5, \dots, 31$, where the e_t^* are chosen independently from $N(0, \hat{\sigma}_e^2)$, $Y_1^* = Y_1$, $Y_2^* = Y_2$, $Y_3^* = Y_3$; and the parameters are set to be the estimates from the AR(3) model fitted to the (square root transformed) hare data with $\hat{\theta}_0 = \hat{\mu}(1 - \hat{\phi}_1 - \hat{\phi}_2 - \hat{\phi}_3)$. All results are based on about 1000 bootstrap replications, but full maximum likelihood estimation fails for 6.3%, 6.3%, 3.8%, and 4.8% of 1000 cases for the four bootstrap methods I, II, III, and IV, respectively.

Exhibit 7.10 Bootstrap and Theoretical Confidence Intervals for the AR(3) Model Fitted to the Hare Data

Method	ar1	ar2	ar3	intercept	noise var.
I	(0.593, 1.269)	(-0.655, 0.237)	(-0.666, -0.018)	(5.115, 6.394)	(0.551, 1.546)
II	(0.612, 1.296)	(-0.702, 0.243)	(-0.669, -0.026)	(5.004, 6.324)	(0.510, 1.510)
III	(0.699, 1.369)	(-0.746, 0.195)	(-0.666, -0.021)	(5.056, 6.379)	(0.499, 1.515)
IV	(0.674, 1.389)	(-0.769, 0.194)	(-0.665, -0.002)	(4.995, 6.312)	(0.477, 1.530)
Theoretical	(0.684, 1.42)	(-0.8058, 0.3474)	(-0.7684, -0.01776)	(5.032, 6.353)	(0.536, 1.597)

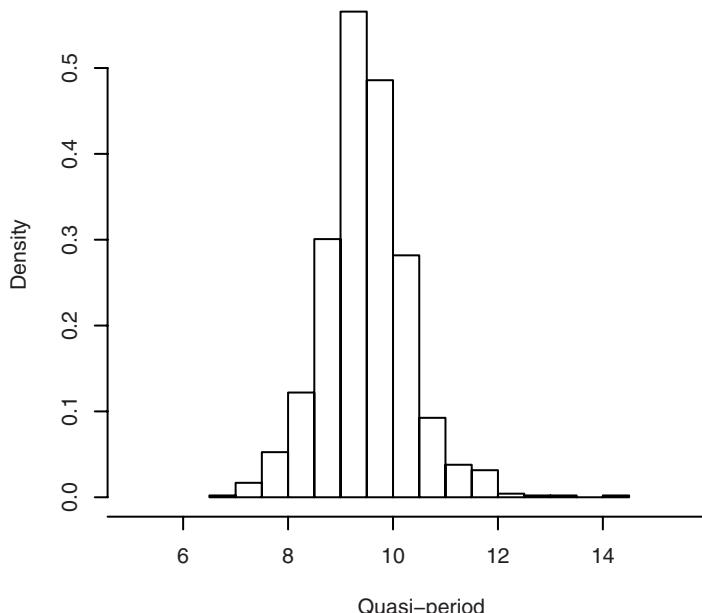
> See the Chapter 7 R scripts file for the extensive code required to generate these results.

All four methods yield similar bootstrap confidence intervals, although the conditional bootstrap approach generally yields slightly narrower confidence intervals. This is expected, as the conditional bootstrap time series bear more resemblance to each other because all are subject to identical initial conditions. The bootstrap confidence intervals are generally slightly wider than their theoretical counterparts that are derived from the large-sample results. Overall, we can draw the inference that the ϕ_2 coefficient estimate is insignificant, whereas both the ϕ_1 and ϕ_3 coefficient estimates are significant at the 5% significance level.

The bootstrap method has the advantage of allowing easy construction of confidence intervals for a model characteristic that is a nonlinear function of the model parameters. For example, the characteristic AR polynomial of the fitted AR(3) model for the hare data admits a pair of complex roots. Indeed, the roots are $0.84 \pm 0.647i$ and -2.26 , where $i = \sqrt{-1}$. The two complex roots can be written in polar form: $1.06\exp(\pm 0.657i)$. As in the discussion of the quasi-period for the AR(2) model on page 74, the quasi-period of the fitted AR(3) model can be defined as $2\pi/0.657 = 9.57$. Thus, the fitted model suggests that the hare abundance underwent cyclical fluctuation with a period of about 9.57 years. The interesting question of constructing a 95% confidence interval for the quasi-period could be studied using the delta method. However, this will be quite complex, as the quasi-period is a complicated function of the parameters. But the bootstrap provides a simple solution: For each set of bootstrap parameter estimates, we can compute the quasi-period and hence obtain the bootstrap distribution of the quasi-period. Confidence intervals for the quasi-period can then be constructed using the percentile method, and the shape of the distribution can be explored via the histogram of the bootstrap quasi-period estimates. (Note that the quasi-period will be unde-

fined whenever the roots of the AR characteristic equation are all real numbers.) Among the 1000 stationary bootstrap time series obtained by simulating from the fitted model with the errors drawn randomly from the residuals with replacement, 952 series lead to successful full maximum likelihood estimation. All but one of the 952 series have well-defined quasi-periods, and the histogram of these is shown in Exhibit 7.11. The histogram shows that the sampling distribution of the quasi-period estimate is slightly skewed to the right.[†] The Q-Q normal plot (Exhibit 7.12) suggests that the quasi-period estimator has, furthermore, a thick-tailed distribution. Thus, the delta method and the corresponding normal distribution approximation may be inappropriate for approximating the sampling distribution of the quasi-period estimator. Finally, using the percentile method, a 95% confidence interval of the quasi-period is found to be (7.84,11.34).

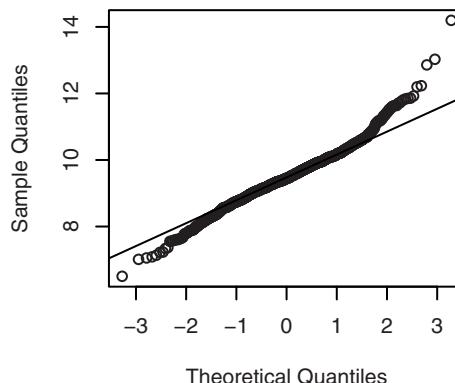
Exhibit 7.11 Histogram of Bootstrap Quasi-period Estimates



```
> win.graph(width=3.9,height=3.8,pointsize=8)
> hist(period.replace,prob=T,xlab='Quasi-period',axes=F,
      xlim=c(5,16))
> axis(2); axis(1,c(4,6,8,10,12,14,16),c(4,6,8,10,12,14,NA))
```

[†] However, see the discussion below Equation (13.5.9) on page 338 where it is argued that, from the perspective of frequency domain, there is a small parametric region corresponding to complex roots and yet the associated quasi-period may not be physically meaningful. This illustrates the subtlety of the concept of quasi-period.

Exhibit 7.12 Q-Q Normal Plot of Bootstrap Quasi-period Estimates



```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(period.replace); qqline(period.replace)
```

7.7 Summary

This chapter delved into the estimation of the parameters of ARIMA models. We considered estimation criteria based on the method of moments, various types of least squares, and maximizing the likelihood function. The properties of the various estimators were given, and the estimators were illustrated both with simulated and actual time series data. Bootstrapping with ARIMA models was also discussed and illustrated.

EXERCISES

- 7.1 From a series of length 100, we have computed $r_1 = 0.8$, $r_2 = 0.5$, $r_3 = 0.4$, $\bar{Y} = 2$, and a sample variance of 5. If we assume that an AR(2) model with a constant term is appropriate, how can we get (simple) estimates of ϕ_1 , ϕ_2 , θ_0 , and σ_ε^2 ?
- 7.2 Assuming that the following data arise from a stationary process, calculate method-of-moments estimates of μ , γ_0 , and ρ_1 : 6, 5, 4, 6, 4.
- 7.3 If $\{Y_t\}$ satisfies an AR(1) model with ϕ of about 0.7, how long of a series do we need to estimate $\phi = \rho_1$ with 95% confidence that our estimation error is no more than ± 0.1 ?
- 7.4 Consider an MA(1) process for which it is *known* that the process mean is zero. Based on a series of length $n = 3$, we observe $Y_1 = 0$, $Y_2 = -1$, and $Y_3 = \frac{1}{2}$.
 - (a) Show that the conditional least-squares estimate of θ is $\frac{1}{2}$.
 - (b) Find an estimate of the noise variance. (Hint: Iterative methods are not needed in this simple case.)

- 7.5** Given the data $Y_1 = 10$, $Y_2 = 9$, and $Y_3 = 9.5$, we wish to fit an IMA(1,1) model without a constant term.

- (a) Find the conditional least squares estimate of θ . (Hint: Do Exercise 7.4 first.)
(b) Estimate σ_e^2 .

- 7.6** Consider two different parameterizations of the AR(1) process with nonzero mean:

Model I. $Y_t - \mu = \phi(Y_{t-1} - \mu) + e_t$.

Model II. $Y_t = \phi Y_{t-1} + \theta_0 + e_t$.

We want to estimate ϕ and μ or ϕ and θ_0 using conditional least squares conditional on Y_1 . Show that with Model I we are led to solve nonlinear equations to obtain the estimates, while with Model II we need only solve linear equations.

- 7.7** Verify Equation (7.1.4) on page 150.

- 7.8** Consider an ARMA(1,1) model with $\phi = 0.5$ and $\theta = 0.45$.

- (a) For $n = 48$, evaluate the variances and correlation of the maximum likelihood estimators of ϕ and θ using Equations (7.4.13) on page 161. Comment on the results.

- (b) Repeat part (a) but now with $n = 120$. Comment on the new results.

- 7.9** Simulate an MA(1) series with $\theta = 0.8$ and $n = 48$.

- (a) Find the method-of-moments estimate of θ .

- (b) Find the conditional least squares estimate of θ and compare it with part (a).

- (c) Find the maximum likelihood estimate of θ and compare it with parts (a) and (b).

- (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare your results with your results from the first simulation.

- 7.10** Simulate an MA(1) series with $\theta = -0.6$ and $n = 36$.

- (a) Find the method-of-moments estimate of θ .

- (b) Find the conditional least squares estimate of θ and compare it with part (a).

- (c) Find the maximum likelihood estimate of θ and compare it with parts (a) and (b).

- (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare your results with your results from the first simulation.

- 7.11** Simulate an MA(1) series with $\theta = -0.6$ and $n = 48$.

- (a) Find the maximum likelihood estimate of θ .

- (b) If your software permits, repeat part (a) many times with a new simulated series using the same parameters and same sample size.

- (c) Form the sampling distribution of the maximum likelihood estimates of θ .

- (d) Are the estimates (approximately) unbiased?

- (e) Calculate the variance of your sampling distribution and compare it with the large-sample result in Equation (7.4.11) on page 161.

- 7.12** Repeat Exercise 7.11 using a sample size of $n = 120$.

- 7.13** Simulate an AR(1) series with $\phi = 0.8$ and $n = 48$.
- (a) Find the method-of-moments estimate of ϕ .
 - (b) Find the conditional least squares estimate of ϕ and compare it with part (a).
 - (c) Find the maximum likelihood estimate of ϕ and compare it with parts (a) and (b).
 - (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare your results with your results from the first simulation.
- 7.14** Simulate an AR(1) series with $\phi = -0.5$ and $n = 60$.
- (a) Find the method-of-moments estimate of ϕ .
 - (b) Find the conditional least squares estimate of ϕ and compare it with part (a).
 - (c) Find the maximum likelihood estimate of ϕ and compare it with parts (a) and (b).
 - (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare your results with your results from the first simulation.
- 7.15** Simulate an AR(1) series with $\phi = 0.7$ and $n = 100$.
- (a) Find the maximum likelihood estimate of ϕ .
 - (b) If your software permits, repeat part (a) many times with a new simulated series using the same parameters and same sample size.
 - (c) Form the sampling distribution of the maximum likelihood estimates of ϕ .
 - (d) Are the estimates (approximately) unbiased?
 - (e) Calculate the variance of your sampling distribution and compare it with the large-sample result in Equation (7.4.9) on page 161.
- 7.16** Simulate an AR(2) series with $\phi_1 = 0.6$, $\phi_2 = 0.3$, and $n = 60$.
- (a) Find the method-of-moments estimates of ϕ_1 and ϕ_2 .
 - (b) Find the conditional least squares estimates of ϕ_1 and ϕ_2 and compare them with part (a).
 - (c) Find the maximum likelihood estimates of ϕ_1 and ϕ_2 and compare them with parts (a) and (b).
 - (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare these results to your results from the first simulation.
- 7.17** Simulate an ARMA(1,1) series with $\phi = 0.7$, $\theta = 0.4$, and $n = 72$.
- (a) Find the method-of-moments estimates of ϕ and θ .
 - (b) Find the conditional least squares estimates of ϕ and θ and compare them with part (a).
 - (c) Find the maximum likelihood estimates of ϕ and θ and compare them with parts (a) and (b).
 - (d) Repeat parts (a), (b), and (c) with a new simulated series using the same parameters and same sample size. Compare your new results with your results from the first simulation.
- 7.18** Simulate an AR(1) series with $\phi = 0.6$, $n = 36$ but with error terms from a t -distribution with 3 degrees of freedom.

- (a) Display the sample PACF of the series. Is an AR(1) model suggested?
(b) Estimate ϕ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.19 Simulate an MA(1) series with $\theta = -0.8$, $n = 60$ but with error terms from a t -distribution with 4 degrees of freedom.
(a) Display the sample ACF of the series. Is an MA(1) model suggested?
(b) Estimate θ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.20 Simulate an AR(2) series with $\phi_1 = 1.0$, $\phi_2 = -0.6$, $n = 48$ but with error terms from a t -distribution with 5 degrees of freedom.
(a) Display the sample PACF of the series. Is an AR(2) model suggested?
(b) Estimate ϕ_1 and ϕ_2 from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.21 Simulate an ARMA(1,1) series with $\phi = 0.7$, $\theta = -0.6$, $n = 48$ but with error terms from a t -distribution with 6 degrees of freedom.
(a) Display the sample EACF of the series. Is an ARMA(1,1) model suggested?
(b) Estimate ϕ and θ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.22 Simulate an AR(1) series with $\phi = 0.6$, $n = 36$ but with error terms from a chi-square distribution with 6 degrees of freedom.
(a) Display the sample PACF of the series. Is an AR(1) model suggested?
(b) Estimate ϕ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.23 Simulate an MA(1) series with $\theta = -0.8$, $n = 60$ but with error terms from a chi-square distribution with 7 degrees of freedom.
(a) Display the sample ACF of the series. Is an MA(1) model suggested?
(b) Estimate θ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.24 Simulate an AR(2) series with $\phi_1 = 1.0$, $\phi_2 = -0.6$, $n = 48$ but with error terms from a chi-square distribution with 8 degrees of freedom.
(a) Display the sample PACF of the series. Is an AR(2) model suggested?
(b) Estimate ϕ_1 and ϕ_2 from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new simulated series under the same conditions.
- 7.25 Simulate an ARMA(1,1) series with $\phi = 0.7$, $\theta = -0.6$, $n = 48$ but with error terms from a chi-square distribution with 9 degrees of freedom.
(a) Display the sample EACF of the series. Is an ARMA(1,1) model suggested?
(b) Estimate ϕ and θ from the series and comment on the results.
(c) Repeat parts (a) and (b) with a new series under the same conditions.

- 7.26** Consider the AR(1) model specified for the color property time series displayed in Exhibit 1.3 on page 3. The data are in the file named *color*.
- (a) Find the method-of-moments estimate of ϕ .
 - (b) Find the maximum likelihood estimate of ϕ and compare it with part (a).
- 7.27** Exhibit 6.31 on page 139 suggested specifying either an AR(1) or possibly an AR(4) model for the difference of the logarithms of the oil price series. The data are in the file named *oil.price*.
- (a) Estimate both of these models using maximum likelihood and compare it with the results using the AIC criteria.
 - (b) Exhibit 6.32 on page 140 suggested specifying an MA(1) model for the difference of the logs. Estimate this model by maximum likelihood and compare to your results in part (a).
- 7.28** The data file named *deere3* contains 57 consecutive values from a complex machine tool at Deere & Co. The values given are deviations from a target value in units of ten millionths of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced.
- (a) Estimate the parameters of an AR(1) model for this series.
 - (b) Estimate the parameters of an AR(2) model for this series and compare the results with those in part (a).
- 7.29** The data file named *robot* contains a time series obtained from an industrial robot. The robot was put through a sequence of maneuvers, and the distance from a desired ending point was recorded in inches. This was repeated 324 times to form the time series.
- (a) Estimate the parameters of an AR(1) model for these data.
 - (b) Estimate the parameters of an IMA(1,1) model for these data.
 - (c) Compare the results from parts (a) and (b) in terms of AIC.
- 7.30** The data file named *days* contains accounting data from the Winegard Co. of Burlington, Iowa. The data are the number of days until Winegard receives payment for 130 consecutive orders from a particular distributor of Winegard products. (The name of the distributor must remain anonymous for confidentiality reasons.) The time series contains outliers that are quite obvious in the time series plot.
- (a) Replace each of the unusual values with a value of 35 days, a much more typical value, and then estimate the parameters of an MA(2) model.
 - (b) Now assume an MA(5) model and estimate the parameters. Compare these results with those obtained in part (a).
- 7.31** Simulate a time series of length $n = 48$ from an AR(1) model with $\phi = 0.7$. Use that series as if it were real data. Now compare the theoretical asymptotic distribution of the estimator of ϕ with the distribution of the bootstrap estimator of ϕ .
- 7.32** The industrial color property time series was fitted quite well by an AR(1) model. However, the series is rather short, with $n = 35$. Compare the theoretical asymptotic distribution of the estimator of ϕ with the distribution of the bootstrap estimator of ϕ . The data are in the file named *color*.

CHAPTER 8

MODEL DIAGNOSTICS

We have now discussed methods for specifying models and for efficiently estimating the parameters in those models. Model diagnostics, or model criticism, is concerned with testing the goodness of fit of a model and, if the fit is poor, suggesting appropriate modifications. We shall present two complementary approaches: analysis of residuals from the fitted model and analysis of overparameterized models; that is, models that are more general than the proposed model but that contain the proposed model as a special case.

8.1 Residual Analysis

We already used the basic ideas of residual analysis in Section 3.6 on page 42 when we checked the adequacy of fitted deterministic trend models. With autoregressive models, residuals are defined in direct analogy to that earlier work. Consider in particular an AR(2) model with a constant term:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \theta_0 + e_t \quad (8.1.1)$$

Having estimated ϕ_1 , ϕ_2 , and θ_0 , the residuals are defined as

$$\hat{e}_t = Y_t - \hat{\phi}_1 Y_{t-1} - \hat{\phi}_2 Y_{t-2} - \hat{\theta}_0 \quad (8.1.2)$$

For general ARMA models containing moving average terms, we use the inverted, infinite autoregressive form of the model to define residuals. For simplicity, we assume that θ_0 is zero. From the inverted form of the model, Equation (4.5.5) on page 80, we have

$$Y_t = \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \pi_3 Y_{t-3} + \cdots + e_t$$

so that the residuals are defined as

$$\hat{e}_t = Y_t - \hat{\pi}_1 Y_{t-1} - \hat{\pi}_2 Y_{t-2} - \hat{\pi}_3 Y_{t-3} - \cdots \quad (8.1.3)$$

Here the π 's are not estimated directly but rather implicitly as functions of the ϕ 's and θ 's. In fact, the residuals are not calculated using this equation but as a by-product of the estimation of the ϕ 's and θ 's. In Chapter 9, we shall argue, that

$$\hat{Y}_t = \hat{\pi}_1 Y_{t-1} + \hat{\pi}_2 Y_{t-2} + \hat{\pi}_3 Y_{t-3} + \cdots$$

is the best forecast of Y_t based on $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots$. Thus Equation (8.1.3) can be rewritten as

$$\text{residual} = \text{actual} - \text{predicted}$$

in direct analogy with regression models. Compare this with Section 3.6 on page 42.

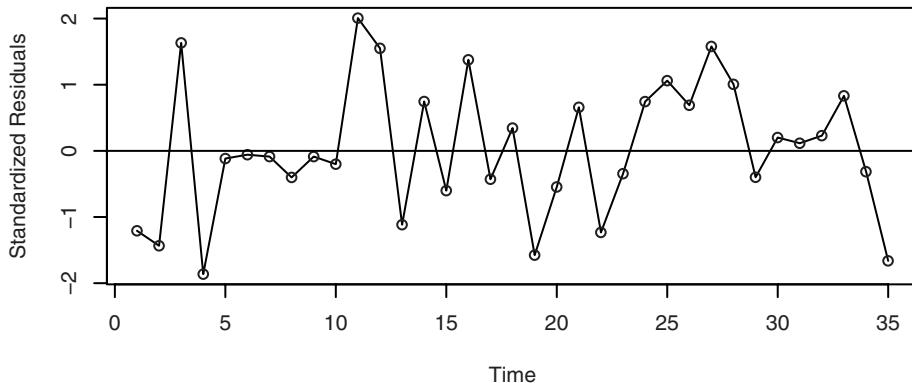
If the model is correctly specified and the parameter estimates are reasonably close to the true values, then the residuals should have nearly the properties of white noise. They should behave roughly like independent, identically distributed normal variables with zero means and common standard deviations. Deviations from these properties can help us discover a more appropriate model.

Plots of the Residuals

Our first diagnostic check is to inspect a plot of the residuals over time. If the model is adequate, we expect the plot to suggest a rectangular scatter around a zero horizontal level with no trends whatsoever.

Exhibit 8.1 shows such a plot for the standardized residuals from the AR(1) model fitted to the industrial color property series. Standardization allows us to see residuals of unusual size much more easily. The parameters were estimated using maximum likelihood. This plot supports the model, as no trends are present.

Exhibit 8.1 Standardized Residuals from AR(1) Model of Color



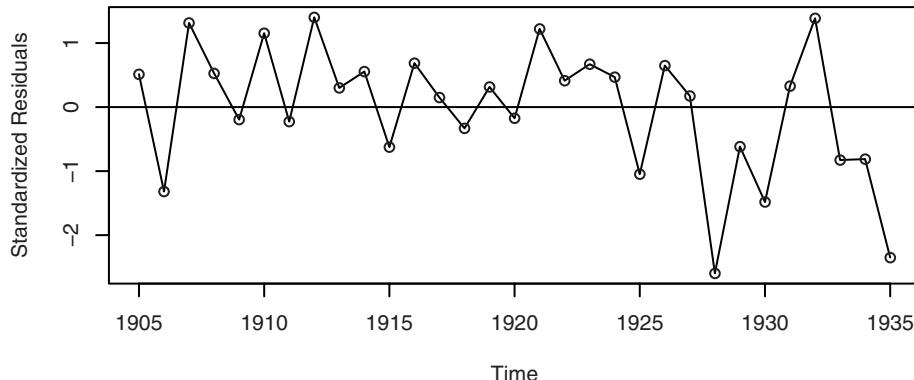
```
> win.graph(width=4.875, height=3, pointsize=8)
> data(color)
> m1.color=arima(color, order=c(1, 0, 0)); m1.color
> plot(rstandard(m1.color), ylab = 'Standardized Residuals',
  type='o'); abline(h=0)
```

As a second example, we consider the Canadian hare abundance series. We estimate a subset AR(3) model with ϕ_2 set to zero, as suggested by the discussion following Exhibit 7.8 on page 166. The estimated model is

$$\sqrt{Y_t} = 3.483 + 0.919\sqrt{Y_{t-1}} - 0.5313\sqrt{Y_{t-3}} + e_t \quad (8.1.4)$$

and the time series plot of the standardized residuals from this model is shown in Exhibit 8.2. Here we see possible reduced variation in the middle of the series and increased variation near the end of the series—not exactly an ideal plot of residuals.[†]

Exhibit 8.2 Standardized Residuals from AR(3) Model for Sqrt(Hare)



```

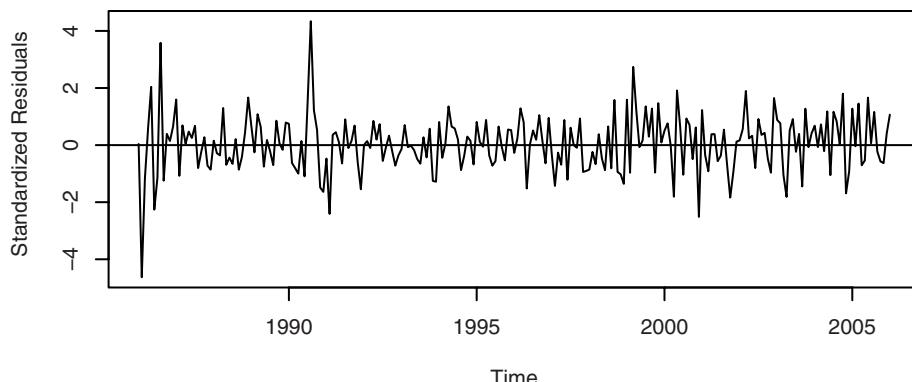
> data(hare)
> m1.hare=arima(sqrt(hare),order=c(3,0,0)); m1.hare
> m2.hare=arima(sqrt(hare),order=c(3,0,0),fixed=c(NA,0,NA,NA))
> m2.hare
> # Note that the intercept term given in R is actually the mean
   in the centered form of the ARMA model; that is, if
   y(t)=sqrt(hare)-intercept, then the model is
   y(t)=0.919*y(t-1)-0.5313*y(t-3)+e(t)
> # So the 'true' intercept equals 5.6889*(1-0.919+0.5313)=3.483
> plot(rstandard(m2.hare),ylab='Standardized Residuals',type='o')
> abline(h=0)

```

Exhibit 8.3 displays the time series plot of the standardized residuals from the IMA(1,1) model estimated for the logarithms of the oil price time series. The model was fitted using maximum likelihood estimation. There are at least two or three residuals early in the series with magnitudes larger than 3—very unusual in a standard normal distribution.[‡] Ideally, we should go back to those months and try to learn what outside factors may have influenced unusually large drops or unusually large increases in the price of oil.

[†] The seemingly large negative standardized residuals are not outliers according to the Bonferroni outlier criterion with critical values ± 3.15 .

[‡] The Bonferroni critical values with $n = 241$ and $\alpha = 0.05$ are ± 3.71 , so the outliers do appear to be real. We will model them in Chapter 11.

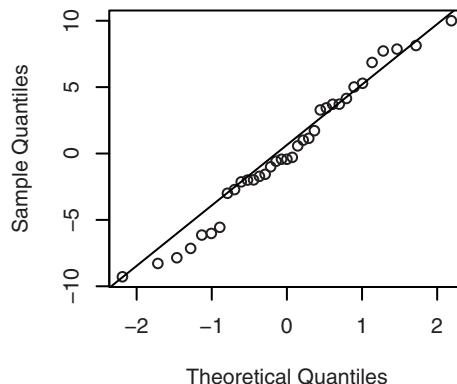
Exhibit 8.3 Standardized Residuals from Log Oil Price IMA(1,1) Model

```
> data(oil.price)
> m1.oil=arima(log(oil.price),order=c(0,1,1))
> plot(rstandard(m1.oil),ylab='Standardized residuals',type='l')
> abline(h=0)
```

Normality of the Residuals

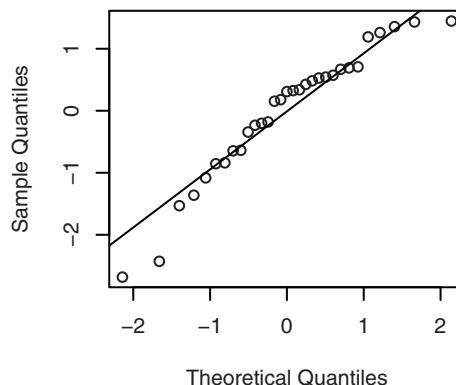
As we saw in Chapter 3, quantile-quantile plots are an effective tool for assessing normality. Here we apply them to residuals.

A quantile-quantile plot of the residuals from the AR(1) model estimated for the industrial color property series is shown in Exhibit 8.4. The points seem to follow the straight line fairly closely—especially the extreme values. This graph would not lead us to reject normality of the error terms in this model. In addition, the Shapiro-Wilk normality test applied to the residuals produces a test statistic of $W = 0.9754$, which corresponds to a p -value of 0.6057, and we would not reject normality based on this test.

Exhibit 8.4 Quantile-Quantile Plot: Residuals from AR(1) Color Model

```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(residuals(m1.color)); qqline(residuals(m1.color))
```

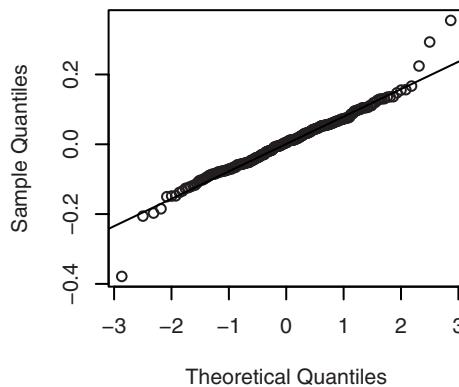
The quantile-quantile plot for the residuals from the AR(3) model for the square root of the hare abundance time series is displayed in Exhibit 8.5. Here the extreme values look suspect. However, the sample is small ($n = 31$) and, as stated earlier, the Bonferroni criteria for outliers do not indicate cause for alarm.

Exhibit 8.5 Quantile-Quantile Plot: Residuals from AR(3) for Hare

```
> qqnorm(residuals(m1.hare)); qqline(residuals(m1.hare))
```

Exhibit 8.6 gives the quantile-quantile plot for the residuals from the IMA(1,1) model that was used to model the logarithms of the oil price series. Here the outliers are quite prominent, and we will deal with them in Chapter 11.

Exhibit 8.6 Quantile-Quantile Plot: Residuals from IMA(1,1) Model for Oil



```
> qqnorm(residuals(m1.oil)); qqline(residuals(m1.oil))
```

Autocorrelation of the Residuals

To check on the independence of the noise terms in the model, we consider the sample autocorrelation function of the residuals, denoted \hat{r}_k . From Equation (6.1.3) on page 110, we know that for true white noise and large n , the sample autocorrelations are approximately uncorrelated and normally distributed with zero means and variance $1/n$. Unfortunately, even residuals from a correctly specified model with efficiently estimated parameters have somewhat different properties. This was first explored for multiple-regression models in a series of papers by Durbin and Watson (1950, 1951, 1971) and for autoregressive models in Durbin (1970). The key reference on the distribution of residual autocorrelations in ARIMA models is Box and Pierce (1970), the results of which were generalized in McLeod (1978).

Generally speaking, the residuals are approximately normally distributed with zero means; however, for small lags k and j , the variance of \hat{r}_k can be substantially less than $1/n$ and the estimates \hat{r}_k and \hat{r}_j can be highly correlated. For larger lags, the approximate variance $1/n$ does apply, and further \hat{r}_k and \hat{r}_j are approximately uncorrelated.

As an example of these results, consider a correctly specified and efficiently estimated AR(1) model. It can be shown that, for large n ,

$$\text{Var}(\hat{r}_1) \approx \frac{\phi^2}{n} \quad (8.1.5)$$

$$\text{Var}(\hat{r}_k) \approx \frac{1 - (1 - \phi^2)\phi^{2k-2}}{n} \quad \text{for } k > 1 \quad (8.1.6)$$

$$\text{Corr}(\hat{r}_1, \hat{r}_k) \approx -\text{sign}(\phi) \frac{(1 - \phi^2)\phi^{k-2}}{1 - (1 - \phi^2)\phi^{2k-2}} \quad \text{for } k > 1 \quad (8.1.7)$$

where

$$\text{sign}(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{if } \phi = 0 \\ -1 & \text{if } \phi < 0 \end{cases}$$

The table in Exhibit 8.7 illustrates these formulas for a variety of values of ϕ and k . Notice that $\text{Var}(\hat{r}_1) \approx 1/n$ is a reasonable approximation for $k \geq 2$ over a wide range of ϕ -values.

Exhibit 8.7 Approximations for Residual Autocorrelations in AR(1) Models

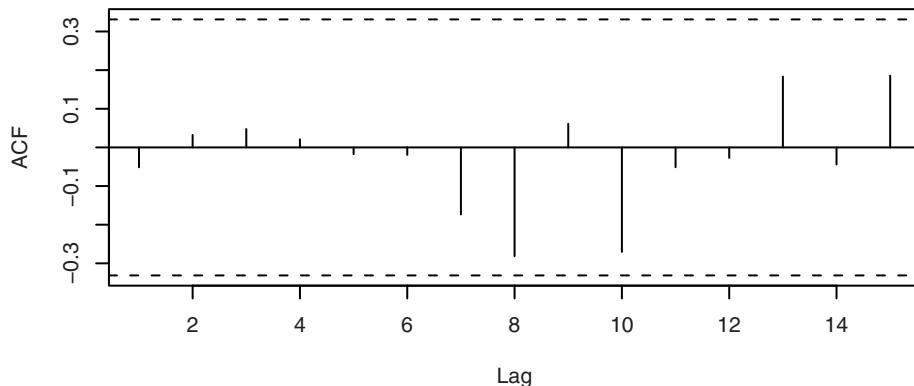
ϕ	0.3	0.5	0.7	0.9	ϕ	0.3	0.5	0.7	0.9
k	Standard deviation of \hat{r}_k times \sqrt{n}				Correlation \hat{r}_1 with \hat{r}_k				
1	0.30	0.50	0.70	0.90		1.00	1.00	1.00	1.00
2	0.96	0.90	0.87	0.92		-0.95	-0.83	-0.59	-0.21
3	1.00	0.98	0.94	0.94		-0.27	-0.38	-0.38	-0.18
4	1.00	0.99	0.97	0.95		-0.08	-0.19	-0.26	-0.16
5	1.00	1.00	0.99	0.96		-0.02	-0.09	-0.18	-0.14
6	1.00	1.00	0.99	0.97		-0.01	-0.05	-0.12	-0.13
7	1.00	1.00	1.00	0.97		-0.00	-0.02	-0.09	-0.12
8	1.00	1.00	1.00	0.98		-0.00	-0.01	-0.06	-0.10
9	1.00	1.00	1.00	0.99		-0.00	-0.00	-0.03	-0.08

If we apply these results to the AR(1) model that was estimated for the industrial color property time series with $\hat{\phi} = 0.57$ and $n = 35$, we obtain the results shown in Exhibit 8.8.

Exhibit 8.8 Approximate Standard Deviations of Residual ACF values

Lag k	1	2	3	4	5	> 5
$\sqrt{\text{Var}(\hat{r}_k)}$	0.096	0.149	0.163	0.167	0.168	0.169

A graph of the sample ACF of these residuals is shown in Exhibit 8.9. The dashed horizontal lines plotted are based on the large lag standard error of $\pm 2/\sqrt{n}$. There is no evidence of autocorrelation in the residuals of this model.

Exhibit 8.9 Sample ACF of Residuals from AR(1) Model for Color

```
> win.graph(width=4.875, height=3, pointsize=8)
> acf(residuals(m1.color))
```

For an AR(2) model, it can be shown that

$$\text{Var}(\hat{r}_1) \approx \frac{\phi_2^2}{n} \quad (8.1.8)$$

and

$$\text{Var}(\hat{r}_2) \approx \frac{\phi_2^2 + \phi_1^2(1 + \phi_2)^2}{n} \quad (8.1.9)$$

If the AR(2) parameters are not too close to the stationarity boundary shown in Exhibit 4.17 on page 72, then

$$\text{Var}(\hat{r}_k) \approx \frac{1}{n} \quad \text{for } k \geq 3 \quad (8.1.10)$$

If we fit an AR(2) model[†] by maximum likelihood to the square root of the hare abundance series, we find that $\hat{\phi}_1 = 1.351$ and $\hat{\phi}_2 = -0.776$. Thus we have

$$\sqrt{\text{Var}(\hat{r}_1)} \approx \frac{|-0.776|}{\sqrt{35}} = 0.131$$

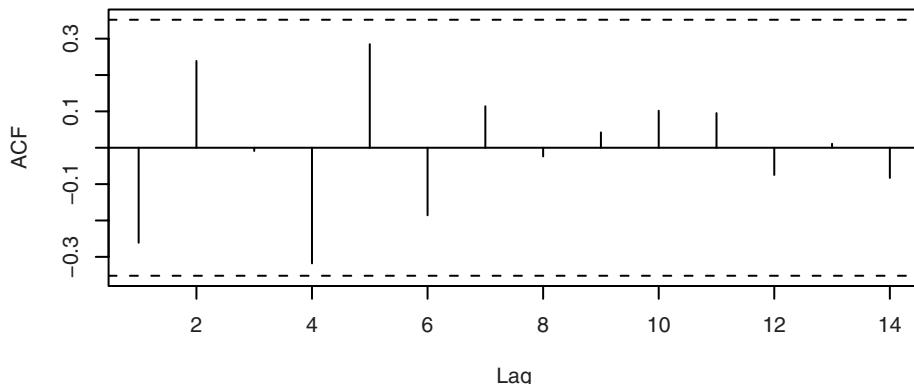
$$\sqrt{\text{Var}(\hat{r}_2)} \approx \sqrt{\frac{(-0.776)^2 + (1.351)^2(1 + (-0.776))^2}{35}} = 0.141$$

$$\sqrt{\text{Var}(\hat{r}_k)} \approx 1/\sqrt{35} = 0.169 \quad \text{for } k \geq 3$$

[†] The AR(2) model is not quite as good as the AR(3) model that we estimated earlier, but it still fits quite well and serves as a reasonable example here.

Exhibit 8.10 displays the sample ACF of the residuals from the AR(2) model of the square root of the hare abundance. The lag 1 autocorrelation here equals -0.261 , which is close to 2 standard errors below zero but not quite. The lag 4 autocorrelation equals -0.318 , but its standard error is 0.169 . We conclude that the graph does not show statistically significant evidence of nonzero autocorrelation in the residuals.[†]

Exhibit 8.10 Sample ACF of Residuals from AR(2) Model for Hare



```
> acf(residuals(arima(sqrt(hare), order=c(2, 0, 0))))
```

With monthly data, we would pay special attention to possible excessive autocorrelation in the residuals at lags 12, 24, and so forth. With quarterly series, lags 4, 8, and so forth would merit special attention. Chapter 10 contains examples of these ideas.

It can be shown that results analogous to those for AR models hold for MA models. In particular, replacing ϕ by θ in Equations (8.1.5), (8.1.6), and (8.1.7) gives the results for the MA(1) case. Similarly, results for the MA(2) case can be stated by replacing ϕ_1 and ϕ_2 by θ_1 and θ_2 , respectively, in Equations (8.1.8), (8.1.9), and (8.1.10). Results for general ARMA models may be found in Box and Pierce (1970) and McLeod (1978).

The Ljung-Box Test

In addition to looking at residual correlations at individual lags, it is useful to have a test that takes into account their magnitudes as a group. For example, it may be that most of the residual autocorrelations are moderate, some even close to their critical values, but, taken together, they seem excessive. Box and Pierce (1970) proposed the statistic

$$Q = n(\hat{r}_1^2 + \hat{r}_2^2 + \dots + \hat{r}_K^2) \quad (8.1.11)$$

to address this possibility. They showed that if the correct ARMA(p, q) model is estimated, then, for large n , Q has an approximate chi-square distribution with $K - p - q$

[†] Recall that an AR(3) model fits these data even better and has even less autocorrelation in its residuals, see Exercise 8.7.

degrees of freedom. Fitting an erroneous model would tend to inflate Q . Thus, a general “portmanteau” test would reject the ARMA(p, q) model if the observed value of Q exceeded an appropriate critical value in a chi-square distribution with $K - p - q$ degrees of freedom. (Here the maximum lag K is selected somewhat arbitrarily but large enough that the ψ -weights are negligible for $j > K$.)

The chi-square distribution for Q is based on a limit theorem as $n \rightarrow \infty$, but Ljung and Box (1978) subsequently discovered that even for $n = 100$, the approximation is not satisfactory. By modifying the Q statistic slightly, they defined a test statistic whose null distribution is much closer to chi-square for typical sample sizes. The modified Box-Pierce, or **Ljung-Box**, statistic is given by

$$Q_* = n(n+2) \left(\frac{\hat{r}_1^2}{n-1} + \frac{\hat{r}_2^2}{n-2} + \dots + \frac{\hat{r}_K^2}{n-K} \right) \quad (8.1.12)$$

Notice that since $(n+2)/(n-k) > 1$ for every $k \geq 1$, we have $Q_* > Q$, which partly explains why the original statistic Q tended to overlook inadequate models. More details on the exact distributions of Q_* and Q for finite samples can be found in Ljung and Box (1978), see also Davies, Triggs, and Newbold (1977).

Exhibit 8.11 lists the first six autocorrelations of the residuals from the AR(1) fitted model for the color property series. Here $n = 35$.

Exhibit 8.11 Residual Autocorrelation Values from AR(1) Model for Color

Lag k	1	2	3	4	5	6
Residual ACF	-0.051	0.032	0.047	0.021	-0.017	-0.019

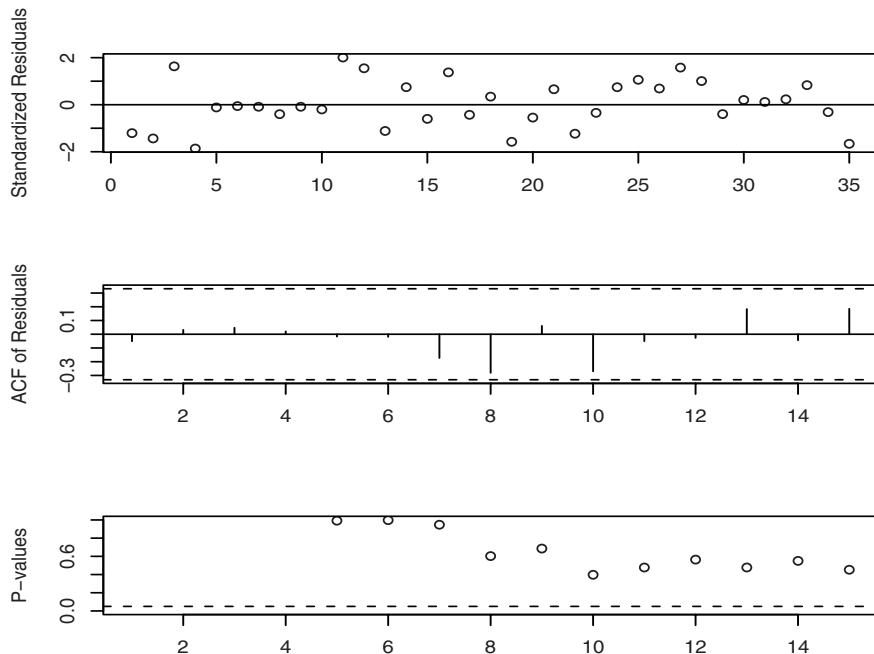
```
> acf(residuals(m1.color), plot=F)$acf
> signif(acf(residuals(m1.color), plot=F)$acf[1:6], 2)
> # display the first 6 acf values to 2 significant digits
```

The Ljung-Box test statistic with $K = 6$ is equal to

$$\begin{aligned} Q_* &= 35(35+2) \left(\frac{(-0.051)^2}{35-1} + \frac{(0.032)^2}{35-2} + \frac{(0.047)^2}{35-3} \right. \\ &\quad \left. + \frac{(0.021)^2}{35-4} + \frac{(-0.017)^2}{35-5} + \frac{(-0.019)^2}{35-6} \right) \approx 0.28 \end{aligned}$$

This is referred to a chi-square distribution with $6 - 1 = 5$ degrees of freedom. This leads to a p -value of 0.998, so we have no evidence to reject the null hypothesis that the error terms are uncorrelated.

Exhibit 8.12 shows three of our diagnostic tools in one display—a sequence plot of the standardized residuals, the sample ACF of the residuals, and p -values for the Ljung-Box test statistic for a whole range of values of K from 5 to 15. The horizontal dashed line at 5% helps judge the size of the p -values. In this instance, everything looks very good. The estimated AR(1) model seems to be capturing the dependence structure of the color property time series quite well.

Exhibit 8.12 Diagnostic Display for the AR(1) Model of Color Property

```
> win.graph(width=4.875, height=4.5)
> tsdiag(m1.color, gof=15, omit.initial=F)
```

As in Chapter 3, the runs test may also be used to assess dependence in error terms via the residuals. Applying the test to the residuals from the AR(3) model for the Canadian hare abundance series, we obtain expected runs of 16.09677 versus observed runs of 18. The corresponding p -value is 0.602, so we do not have statistically significant evidence against independence of the error terms in this model.

8.2 Overfitting and Parameter Redundancy

Our second basic diagnostic tool is that of **overfitting**. After specifying and fitting what we believe to be an adequate model, we fit a slightly more general model; that is, a model “close by” that contains the original model as a special case. For example, if an AR(2) model seems appropriate, we might overfit with an AR(3) model. The original AR(2) model would be confirmed if:

1. the estimate of the additional parameter, ϕ_3 , is not significantly different from zero, and
2. the estimates for the parameters in common, ϕ_1 and ϕ_2 , do not change significantly from their original estimates.

As an example, we have specified, fitted, and examined the residuals of an AR(1) model for the industrial color property time series. Exhibit 8.13 displays the output from the R software from fitting the AR(1) model, and Exhibit 8.14 shows the results from fitting an AR(2) model to the same series. First note that, in Exhibit 8.14, the estimate of ϕ_2 is not statistically different from zero. This fact supports the choice of the AR(1) model. Secondly, we note that the two estimates of ϕ_1 are quite close—especially when we take into account the magnitude of their standard errors. Finally, note that while the AR(2) model has a slightly larger log-likelihood value, the AR(1) fit has a smaller AIC value. The penalty for fitting the more complex AR(2) model is sufficient to choose the simpler AR(1) model.

Exhibit 8.13 AR(1) Model Results for the Color Property Series

Coefficients: [†]	ar1	Intercept [‡]
	0.5705	74.3293
s.e.	0.1435	1.9151

sigma² estimated as 24.83: log-likelihood = -106.07, AIC = 216.15

[†] m1.color # R code to obtain table

[‡] Recall that the intercept here is the estimate of the process mean μ —not θ_0 .

Exhibit 8.14 AR(2) Model Results for the Color Property Series

Coefficients:	ar1	ar2	Intercept
	0.5173	0.1005	74.1551
s.e.	0.1717	0.1815	2.1463

sigma² estimated as 24.6: log-likelihood = -105.92, AIC = 217.84

> arima(color, order=c(2, 0, 0))

A different overfit for this series would be to try an ARMA(1,1) model. Exhibit 8.15 displays the results of this fit. Notice that the standard errors of the estimated coefficients for this fit are rather larger than what we see in Exhibits 8.13 and 8.14. Regardless, the estimate of ϕ_1 from this fit is not significantly different from the estimate in Exhibit 8.13. Furthermore, as before, the estimate of the new parameter, θ , is not significantly different from zero. This adds further support to the AR(1) model.

Exhibit 8.15 Overfit of an ARMA(1,1) Model for the Color Series

Coefficients:	ar1	ma1	Intercept
	0.6721	-0.1467	74.1730
s.e.	0.2147	0.2742	2.1357
sigma^2 estimated as 24.63: log-likelihood = -105.94, AIC = 219.88			
> arima(color, order=c(1, 0, 1))			

As we have noted, any ARMA(p, q) model can be considered as a special case of a more general ARMA model with the additional parameters equal to zero. However, when generalizing ARMA models, we must be aware of the problem of **parameter redundancy or lack of identifiability**.

To make these points clear, consider an ARMA(1,2) model:

$$Y_t = \phi Y_{t-1} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} \quad (8.2.1)$$

Now replace t by $t-1$ to obtain

$$Y_{t-1} = \phi Y_{t-2} + e_{t-1} - \theta_1 e_{t-2} - \theta_2 e_{t-3} \quad (8.2.2)$$

If we multiply both sides of Equation (8.2.2) by *any constant c* and then subtract it from Equation (8.2.1), we obtain (after rearranging)

$$Y_t - (\phi + c)Y_{t-1} + \phi c Y_{t-2} = e_t - (\theta_1 + c)e_{t-1} - (\theta_2 - \theta_1 c)e_{t-2} + c\theta_2 e_{t-3}$$

This apparently defines an ARMA(2,3) process. But notice that we have the factorizations

$$1 - (\phi + c)x + \phi cx^2 = (1 - \phi x)(1 - cx)$$

and

$$1 - (\theta_1 + c)x - (\theta_2 - c\theta_1)x^2 + c\theta_2 x^3 = (1 - \theta_1 x - \theta_2 x^2)(1 - cx)$$

Thus the AR and MA characteristic polynomials in the ARMA(2,3) process have a common factor of $(1 - cx)$. Even though Y_t does satisfy the ARMA(2,3) model, clearly the parameters in that model are not unique—the constant c is completely arbitrary. We say that we have **parameter redundancy** in the ARMA(2,3) model.[†]

The implications for fitting and overfitting models are as follows:

1. Specify the original model carefully. If a simple model seems at all promising, check it out before trying a more complicated model.
2. When overfitting, do not increase the orders of both the AR and MA parts of the model simultaneously.

[†]In backshift notation, if $\phi(B)Y_t = \theta(B)e_t$ is a correct model, then so is $(1 - cB)\phi(B)Y_t = (1 - cB)\theta(B)e_t$ for *any constant c*. To have unique parameterization in an ARMA model, we must cancel any common factors in the AR and MA characteristic polynomials.

3. Extend the model in directions suggested by the analysis of the residuals. For example, if after fitting an MA(1) model, substantial correlation remains at lag 2 in the residuals, try an MA(2), not an ARMA(1,1).

As an example, consider the color property series once more. We have seen that an AR(1) model fits quite well. Suppose we try an ARMA(2,1) model. The results of this fit are shown in Exhibit 8.16. Notice that even though the estimate of σ_e^2 and the log-likelihood and AIC values are not too far from their best values, the estimates of ϕ_1 , ϕ_2 , and θ are way off, and none would be considered different from zero statistically.

Exhibit 8.16 Overfitted ARMA(2,1) Model for the Color Property Series

Coefficients:	ar1	ar2	ma1	Intercept
	0.2189	0.2735	0.3036	74.1653
s.e.	2.0056	1.1376	2.0650	2.1121

sigma² estimated as 24.58: log-likelihood = -105.91, AIC = 219.82

```
> arima(color, order=c(2, 0, 1))
```

8.3 Summary

The ideas of residual analysis begun in Chapter 3 were considerably expanded in this chapter. We looked at various plots of the residuals, checking the error terms for constant variance, normality, and independence. The properties of the sample autocorrelation of the residuals play a significant role in these diagnostics. The Ljung-Box statistic portmanteau test was discussed as a summary of the autocorrelation in the residuals. Lastly, the ideas of overfitting and parameter redundancy were presented.

EXERCISES

- 8.1** For an AR(1) model with $\phi \approx 0.5$ and $n = 100$, the lag 1 sample autocorrelation of the residuals is 0.5. Should we consider this unusual? Why or why not?
- 8.2** Repeat Exercise 8.1 for an MA(1) model with $\theta \approx 0.5$ and $n = 100$.
- 8.3** Based on a series of length $n = 200$, we fit an AR(2) model and obtain residual autocorrelations of $\hat{r}_1 = 0.13$, $\hat{r}_2 = 0.13$, and $\hat{r}_3 = 0.12$. If $\hat{\phi}_1 = 1.1$ and $\hat{\phi}_2 = -0.8$, do these residual autocorrelations support the AR(2) specification? Individually? Jointly?

- 8.4** Simulate an AR(1) model with $n = 30$ and $\phi = 0.5$.
- (a) Fit the correctly specified AR(1) model and look at a time series plot of the residuals. Does the plot support the AR(1) specification?
 - (b) Display a normal quantile-quantile plot of the standardized residuals. Does the plot support the AR(1) specification?
 - (c) Display the sample ACF of the residuals. Does the plot support the AR(1) specification?
 - (d) Calculate the Ljung-Box statistic summing to $K = 8$. Does this statistic support the AR(1) specification?
- 8.5** Simulate an MA(1) model with $n = 36$ and $\theta = -0.5$.
- (a) Fit the correctly specified MA(1) model and look at a time series plot of the residuals. Does the plot support the MA(1) specification?
 - (b) Display a normal quantile-quantile plot of the standardized residuals. Does the plot support the MA(1) specification?
 - (c) Display the sample ACF of the residuals. Does the plot support the MA(1) specification?
 - (d) Calculate the Ljung-Box statistic summing to $K = 6$. Does this statistic support the MA(1) specification?
- 8.6** Simulate an AR(2) model with $n = 48$, $\phi_1 = 1.5$, and $\phi_2 = -0.75$.
- (a) Fit the correctly specified AR(2) model and look at a time series plot of the residuals. Does the plot support the AR(2) specification?
 - (b) Display a normal quantile-quantile plot of the standardized residuals. Does the plot support the AR(2) specification?
 - (c) Display the sample ACF of the residuals. Does the plot support the AR(2) specification?
 - (d) Calculate the Ljung-Box statistic summing to $K = 12$. Does this statistic support the AR(2) specification?
- 8.7** Fit an AR(3) model by maximum likelihood to the square root of the hare abundance series (filename `hare`).
- (a) Plot the sample ACF of the residuals. Comment on the size of the correlations.
 - (b) Calculate the Ljung-Box statistic summing to $K = 9$. Does this statistic support the AR(3) specification?
 - (c) Perform a runs test on the residuals and comment on the results.
 - (d) Display the quantile-quantile normal plot of the residuals. Comment on the plot.
 - (e) Perform the Shapiro-Wilk test of normality on the residuals.
- 8.8** Consider the oil filter sales data shown in Exhibit 1.8 on page 7. The data are in the file named `oilfilters`.
- (a) Fit an AR(1) model to this series. Is the estimate of the ϕ parameter significantly different from zero statistically?
 - (b) Display the sample ACF of the residuals from the AR(1) fitted model. Comment on the display.

- 8.9** The data file named `robot` contains a time series obtained from an industrial robot. The robot was put through a sequence of maneuvers, and the distance from a desired ending point was recorded in inches. This was repeated 324 times to form the time series. Compare the fits of an AR(1) model and an IMA(1,1) model for these data in terms of the diagnostic tests discussed in this chapter.
- 8.10** The data file named `deere3` contains 57 consecutive values from a complex machine tool at Deere & Co. The values given are deviations from a target value in units of ten millionths of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced. Diagnose the fit of an AR(1) model for these data in terms of the tests discussed in this chapter.
- 8.11** Exhibit 6.31 on page 139, suggested specifying either an AR(1) or possibly an AR(4) model for the difference of the logarithms of the oil price series. (The file-name is `oil.price`).
- (a) Estimate both of these models using maximum likelihood and compare the results using the diagnostic tests considered in this chapter.
- (b) Exhibit 6.32 on page 140, suggested specifying an MA(1) model for the difference of the logs. Estimate this model by maximum likelihood and perform the diagnostic tests considered in this chapter.
- (c) Which of the three models AR(1), AR(4), or MA(1) would you prefer given the results of parts (a) and (b)?

CHAPTER 9

FORECASTING

One of the primary objectives of building a model for a time series is to be able to forecast the values for that series at future times. Of equal importance is the assessment of the precision of those forecasts. In this chapter, we shall consider the calculation of forecasts and their properties for both deterministic trend models and ARIMA models. Forecasts for models that combine deterministic trends with ARIMA stochastic components are considered also.

For the most part, we shall assume that the model is known *exactly*, including specific values for all the parameters. Although this is never true in practice, the use of estimated parameters for large sample sizes does not seriously affect the results.

9.1 Minimum Mean Square Error Forecasting

Based on the available history of the series up to time t , namely $Y_1, Y_2, \dots, Y_{t-1}, Y_t$, we would like to forecast the value of $Y_{t+\ell}$ that will occur ℓ time units into the future. We call time t the **forecast origin** and ℓ the **lead time** for the forecast, and denote the forecast itself as $\hat{Y}_t(\ell)$.

As shown in Appendix F, the minimum mean square error forecast is given by

$$\hat{Y}_t(\ell) = E(Y_{t+\ell} | Y_1, Y_2, \dots, Y_t) \quad (9.1.1)$$

(Appendices E and F on page 218 review the properties of conditional expectation and minimum mean square error prediction.)

The computation and properties of this conditional expectation as related to forecasting will be our concern for the remainder of this chapter.

9.2 Deterministic Trends

Consider once more the deterministic trend model of Chapter 3,

$$Y_t = \mu_t + X_t \quad (9.2.1)$$

where the stochastic component, X_t , has a mean of zero. For this section, we shall assume that $\{X_t\}$ is in fact white noise with variance γ_0 . For the model in Equation (9.2.1), we have

$$\begin{aligned}
\hat{Y}_t(\ell) &= E(\mu_{t+\ell} + X_{t+\ell} | Y_1, Y_2, \dots, Y_t) \\
&= E(\mu_{t+\ell} | Y_1, Y_2, \dots, Y_t) + E(X_{t+\ell} | Y_1, Y_2, \dots, Y_t) \\
&= \mu_{t+\ell} + E(X_{t+\ell})
\end{aligned}$$

or

$$\hat{Y}_t(\ell) = \mu_{t+\ell} \quad (9.2.2)$$

since for $\ell \geq 1$, $X_{t+\ell}$ is independent of $Y_1, Y_2, \dots, Y_{t-1}, Y_t$ and has expected value zero. Thus, in this simple case, forecasting amounts to extrapolating the deterministic time trend into the future.

For the linear trend case, $\mu_t = \beta_0 + \beta_1 t$, the forecast is

$$\hat{Y}_t(\ell) = \beta_0 + \beta_1(t + \ell) \quad (9.2.3)$$

As we emphasized in Chapter 3, this model assumes that the *same* linear time trend persists into the future, and the forecast reflects that assumption. Note that it is the lack of statistical dependence between $Y_{t+\ell}$ and $Y_1, Y_2, \dots, Y_{t-1}, Y_t$ that prevents us from improving on $\mu_{t+\ell}$ as a forecast.

For seasonal models where, say, $\mu_t = \mu_{t+12}$, our forecast is $\hat{Y}_t(\ell) = \mu_{t+12+\ell} = \hat{Y}_{t+\ell}(12)$. Thus the forecast will also be periodic, as desired.

The **forecast error**, $e_t(\ell)$, is given by

$$\begin{aligned}
e_t(\ell) &= Y_{t+\ell} - \hat{Y}_t(\ell) \\
&= \mu_{t+\ell} + X_{t+\ell} - \mu_{t+\ell} \\
&= X_{t+\ell}
\end{aligned}$$

so that

$$E(e_t(\ell)) = E(X_{t+\ell}) = 0$$

That is, the forecasts are **unbiased**. Also

$$Var(e_t(\ell)) = Var(X_{t+\ell}) = \gamma_0 \quad (9.2.4)$$

is the **forecast error variance** for all lead times ℓ .

The cosine trend model for the average monthly temperature series was estimated in Chapter 3 on page 35 as

$$\hat{\mu}_t = 46.2660 + (-26.7079)\cos(2\pi t) + (-2.1697)\sin(2\pi t)$$

Here time is measured in years with a starting value of January 1964, frequency $f = 1$ per year, and the final observed value is for December 1975. To forecast the June 1976 temperature value, we use $t = 1976.41667$ as the time value[†] and obtain

[†] June is the fifth month of the year, and $5/12 \approx 0.416666666\dots$.

$$\begin{aligned}\hat{\mu}_t &= 46.2660 + (-26.7079)\cos(2\pi(1976.41667)) + (-2.1697)\sin(2\pi(1976.41667)) \\ &= 68.3^{\circ}\text{F}\end{aligned}$$

Forecasts for other months are obtained similarly.

9.3 ARIMA Forecasting

For ARIMA models, the forecasts can be expressed in several different ways. Each expression contributes to our understanding of the overall forecasting procedure with respect to computing, updating, assessing precision, or long-term forecasting behavior.

AR(1)

We shall first illustrate many of the ideas with the simple AR(1) process with a nonzero mean that satisfies

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + e_t \quad (9.3.1)$$

Consider the problem of forecasting one time unit into the future. Replacing t by $t+1$ in Equation (9.3.1), we have

$$Y_{t+1} - \mu = \phi(Y_t - \mu) + e_{t+1} \quad (9.3.2)$$

Given $Y_1, Y_2, \dots, Y_{t-1}, Y_t$, we take the conditional expectations of both sides of Equation (9.3.2) and obtain

$$\hat{Y}_t(1) - \mu = \phi[E(Y_t | Y_1, Y_2, \dots, Y_t) - \mu] + E(e_{t+1} | Y_1, Y_2, \dots, Y_t) \quad (9.3.3)$$

Now, from the properties of conditional expectation, we have

$$E(Y_t | Y_1, Y_2, \dots, Y_t) = Y_t \quad (9.3.4)$$

Also, since e_{t+1} is independent of $Y_1, Y_2, \dots, Y_{t-1}, Y_t$, we obtain

$$E(e_{t+1} | Y_1, Y_2, \dots, Y_t) = E(e_{t+1}) = 0 \quad (9.3.5)$$

Thus, Equation (9.3.3) can be written as

$$\hat{Y}_t(1) = \mu + \phi(Y_t - \mu) \quad (9.3.6)$$

In words, a proportion ϕ of the current deviation from the process mean is added to the process mean to forecast the next process value.

Now consider a general lead time ℓ . Replacing t by $t+\ell$ in Equation (9.3.1) and taking the conditional expectations of both sides produces

$$\hat{Y}_t(\ell) = \mu + \phi[\hat{Y}_t(\ell-1) - \mu] \quad \text{for } \ell \geq 1 \quad (9.3.7)$$

since $E(Y_{t+\ell-1} | Y_1, Y_2, \dots, Y_t) = \hat{Y}_t(\ell-1)$ and, for $\ell \geq 1$, $e_{t+\ell}$ is independent of $Y_1, Y_2, \dots, Y_{t-1}, Y_t$.

Equation (9.3.7), which is recursive in the lead time ℓ , shows how the forecast for any lead time ℓ can be built up from the forecasts for shorter lead times by starting with the initial forecast $\hat{Y}_t(1)$ computed using Equation (9.3.6). The forecast $\hat{Y}_t(2)$ is then obtained from $\hat{Y}_t(2) = \mu + \phi[\hat{Y}_t(1) - \mu]$, then $\hat{Y}_t(3)$ from $\hat{Y}_t(2)$, and so on until the desired $\hat{Y}_t(\ell)$ is found. Equation (9.3.7) and its generalizations for other ARIMA models are most convenient for actually computing the forecasts. Equation (9.3.7) is sometimes called the **difference equation form** of the forecasts.

However, Equation (9.3.7) can also be solved to yield an explicit expression for the forecasts in terms of the observed history of the series. Iterating backward on ℓ in Equation (9.3.7), we have

$$\begin{aligned}\hat{Y}_t(\ell) &= \phi[\hat{Y}_t(\ell-1) - \mu] + \mu \\ &= \phi\{\phi[\hat{Y}_t(\ell-2) - \mu]\} + \mu \\ &\vdots \\ &= \phi^{\ell-1}[\hat{Y}_t(1) - \mu] + \mu\end{aligned}$$

or

$$\hat{Y}_t(\ell) = \mu + \phi^\ell(Y_t - \mu) \quad (9.3.8)$$

The current deviation from the mean is discounted by a factor ϕ^ℓ , whose magnitude decreases with increasing lead time. The discounted deviation is then added to the process mean to produce the lead ℓ forecast.

As a numerical example, consider the AR(1) model that we have fitted to the industrial color property time series. The maximum likelihood estimation results were partially shown in Exhibit 7.7 on page 165, but more complete results are shown in Exhibit 9.1.

Exhibit 9.1 Maximum Likelihood Estimation of an AR(1) Model for Color

Coefficients:	ar1	intercept [†]
	0.5705	74.3293
s.e.	0.1435	1.9151

sigma^2 estimated as 24.8: log-likelihood = -106.07, AIC = 216.15

[†]Remember that the intercept here is the estimate of the process mean μ —not θ_0 .

```
> data(color)
> m1.color=arima(color,order=c(1,0,0))
> m1.color
```

For illustration purposes, we assume that the estimates $\phi = 0.5705$ and $\mu = 74.3293$ are true values. The final forecasts may then be rounded.

The last observed value of the color property is 67, so we would forecast one time period ahead as[†]

$$\begin{aligned}\hat{Y}_t(1) &= 74.3293 + (0.5705)(67 - 74.3293) \\ &= 74.3293 - 4.181366 \\ &= 70.14793\end{aligned}$$

For lead time 2, we have from Equation (9.3.7)

$$\begin{aligned}\hat{Y}_t(2) &= 74.3293 + 0.5705(70.14793 - 74.3293) \\ &= 74.3293 - 2.385472 \\ &= 71.94383\end{aligned}$$

Alternatively, we can use Equation (9.3.8):

$$\begin{aligned}\hat{Y}_t(2) &= 74.3293 + (0.5705)^2(67 - 74.3293) \\ &= 71.92823\end{aligned}$$

At lead 5, we have

$$\begin{aligned}\hat{Y}_t(5) &= 74.3293 + (0.5705)^5(67 - 74.3293) \\ &= 73.88636\end{aligned}$$

and by lead 10 the forecast is

$$\hat{Y}_t(10) = 74.30253$$

which is very nearly μ ($= 74.3293$). In reporting these forecasts we would probably round to the nearest tenth.

In general, since $|\phi| < 1$, we have simply

$$\hat{Y}_t(\ell) \approx \mu \text{ for large } \ell \quad (9.3.9)$$

Later we shall see that Equation (9.3.9) holds for *all stationary ARMA models*.

Consider now the **one-step-ahead forecast error**, $e_t(1)$. From Equations (9.3.2) and (9.3.6), we have

$$\begin{aligned}e_t(1) &= Y_{t+1} - \hat{Y}_t(1) \\ &= [\phi(Y_t - \mu) + \mu + e_{t+1}] - [\phi(Y_t - \mu) + \mu]\end{aligned}$$

or

$$e_t(1) = e_{t+1} \quad (9.3.10)$$

[†] As round off error will accumulate, you should use many decimal places when performing recursive calculations.

The white noise process $\{e_t\}$ can now be reinterpreted as a sequence of one-step-ahead forecast errors. We shall see that Equation (9.3.10) persists for completely general ARIMA models. Note also that Equation (9.3.10) implies that the forecast error $e_t(1)$ is independent of the history of the process $Y_1, Y_2, \dots, Y_{t-1}, Y_t$ up to time t . If this were not so, the dependence could be exploited to improve our forecast.

Equation (9.3.10) also implies that our one-step-ahead forecast error variance is given by

$$\text{Var}(e_t(1)) = \sigma_e^2 \quad (9.3.11)$$

To investigate the properties of the forecast errors for longer leads, it is convenient to express the AR(1) model in general linear process, or MA(∞), form. From Equation (4.3.8) on page 70, we recall that

$$Y_t = e_t + \phi e_{t-1} + \phi^2 e_{t-2} + \phi^3 e_{t-3} + \dots \quad (9.3.12)$$

Then Equations (9.3.8) and (9.3.12) together yield

$$\begin{aligned} e_t(\ell) &= Y_{t+\ell} - \mu - \phi^\ell (Y_t - \mu) \\ &= e_{t+\ell} + \phi e_{t+\ell-1} + \dots + \phi^{\ell-1} e_{t+1} + \phi^\ell e_t \\ &\quad + \dots - \phi^\ell (e_t + \phi e_{t-1} + \dots) \end{aligned}$$

so that

$$e_t(\ell) = e_{t+\ell} + \phi e_{t+\ell-1} + \dots + \phi^{\ell-1} e_{t+1} \quad (9.3.13)$$

which can also be written as

$$e_t(\ell) = e_{t+\ell} + \psi_1 e_{t+\ell-1} + \psi_2 e_{t+\ell-2} + \dots + \psi_{\ell-1} e_{t+1} \quad (9.3.14)$$

Equation (9.3.14) will be shown to hold for all ARIMA models (see Equation (9.3.43) on page 202).

Note that $E(e_t(\ell)) = 0$; thus the forecasts are **unbiased**. Furthermore, from Equation (9.3.14), we have

$$\text{Var}(e_t(\ell)) = \sigma_e^2 (1 + \psi_1^2 + \psi_2^2 + \dots + \psi_{\ell-1}^2) \quad (9.3.15)$$

We see that the forecast error variance increases as the lead ℓ increases. Contrast this with the result given in Equation (9.2.4) on page 192, for deterministic trend models.

In particular, for the AR(1) case,

$$\text{Var}(e_t(\ell)) = \sigma_e^2 \left[\frac{1 - \phi^{2\ell}}{1 - \phi^2} \right] \quad (9.3.16)$$

which we obtain by summing a finite geometric series.

For long lead times, we have

$$Var(e_t(\ell)) \approx \frac{\sigma_e^2}{1 - \phi^2} \text{ for large } \ell \quad (9.3.17)$$

or, by Equation (4.3.3), page 66,

$$Var(e_t(\ell)) \approx Var(Y_t) = \gamma_0 \text{ for large } \ell \quad (9.3.18)$$

Equation (9.3.18) will be shown to be valid for *all stationary ARMA processes* (see Equation (9.3.39) on page 201).

MA(1)

To illustrate how to solve the problems that arise in forecasting moving average or mixed models, consider the MA(1) case with nonzero mean:

$$Y_t = \mu + e_t - \theta e_{t-1}$$

Again replacing t by $t+1$ and taking conditional expectations of both sides, we have

$$\hat{Y}_t(1) = \mu - \theta E(e_t | Y_1, Y_2, \dots, Y_t) \quad (9.3.19)$$

However, for an invertible model, Equation (4.5.2) on page 80 shows that e_t is a function of Y_1, Y_2, \dots, Y_t and so

$$E(e_t | Y_1, Y_2, \dots, Y_t) = e_t \quad (9.3.20)$$

In fact, an approximation is involved in this equation since we are conditioning only on Y_1, Y_2, \dots, Y_t and not on the infinite history of the process. However, if, as in practice, t is large and the model is invertible, the error in the approximation will be very small. If the model is not invertible—for example, if we have overdifferenced the data—then Equation (9.3.20) is not even approximately valid; see Harvey (1981c, p.161).

Using Equations (9.3.19) and (9.3.20), we have the one-step-ahead forecast for an invertible MA(1) expressed as

$$\hat{Y}_t(1) = \mu - \theta e_t \quad (9.3.21)$$

The computation of e_t will be a by-product of estimating the parameters in the model.

Notice once more that the one-step-ahead forecast error is

$$\begin{aligned} e_t(1) &= Y_{t+1} - \hat{Y}_t(1) \\ &= (\mu + e_{t+1} - \theta e_t) - (\mu - \theta e_t) \\ &= e_{t+1} \end{aligned}$$

as in Equation (9.3.10), and thus Equation (9.3.11) also obtains.

For longer lead times, we have

$$\hat{Y}_t(\ell) = \mu + E(e_{t+\ell} | Y_1, Y_2, \dots, Y_t) - \theta E(e_{t+\ell-1} | Y_1, Y_2, \dots, Y_t)$$

But, for $\ell > 1$, both $e_{t+\ell}$ and $e_{t+\ell-1}$ are independent of Y_1, Y_2, \dots, Y_t . Consequently, these conditional expected values are the unconditional expected values, namely zero, and we have

$$\hat{Y}_t(\ell) = \mu \text{ for } \ell > 1 \quad (9.3.22)$$

Notice here that Equation (9.3.9) on page 195 holds exactly for the MA(1) case when $\ell > 1$. Since for this model we trivially have $\psi_1 = -\theta$ and $\psi_j = 0$ for $j > 1$, Equations (9.3.14) and (9.3.15) also hold.

The Random Walk with Drift

To illustrate forecasting with nonstationary ARIMA series, consider the random walk with drift defined by

$$Y_t = Y_{t-1} + \theta_0 + e_t \quad (9.3.23)$$

Here

$$\hat{Y}_t(1) = E(Y_t | Y_1, Y_2, \dots, Y_t) + \theta_0 + E(e_{t+1} | Y_1, Y_2, \dots, Y_t)$$

so that

$$\hat{Y}_t(1) = Y_t + \theta_0 \quad (9.3.24)$$

Similarly, the difference equation form for the lead ℓ forecast is

$$\hat{Y}_t(\ell) = \hat{Y}_t(\ell-1) + \theta_0 \text{ for } \ell \geq 1 \quad (9.3.25)$$

and iterating backward on ℓ yields the explicit expression

$$\hat{Y}_t(\ell) = Y_t + \theta_0 \ell \text{ for } \ell \geq 1 \quad (9.3.26)$$

In contrast to Equation (9.3.9) on page 195, if $\theta_0 \neq 0$, the forecast does not converge for long leads but rather follows a straight line with slope θ_0 for all ℓ .

Note that the presence or absence of the constant term θ_0 significantly alters the nature of the forecast. For this reason, constant terms should not be included in nonstationary ARIMA models unless the evidence is clear that the mean of the differenced series is significantly different from zero. Equation (3.2.3) on page 28 for the variance of the sample mean will help assess this significance.

However, as we have seen in the AR(1) and MA(1) cases, the one-step-ahead forecast error is

$$e_t(1) = Y_{t+1} - \hat{Y}_t(1) = e_{t+1}$$

Also

$$\begin{aligned}
e_t(\ell) &= Y_{t+\ell} - \hat{Y}_t(\ell) \\
&= (Y_t + \ell\theta_0 + e_{t+1} + \dots + e_{t+\ell}) - (Y_t + \ell\theta_0) \\
&= e_{t+1} + e_{t+2} + \dots + e_{t+\ell}
\end{aligned}$$

which agrees with Equation (9.3.14) on page 196 since in this model $\psi_j = 1$ for all j . (See Equation (5.2.6) on page 93 with $\theta = 0$.)

So, as in Equation (9.3.15), we have

$$Var(e_t(\ell)) = \sigma_e^2 \sum_{j=0}^{\ell-1} \psi_j^2 = \ell \sigma_e^2 \quad (9.3.27)$$

In contrast to the stationary case, here $Var(e_t(\ell))$ grows without limit as the forecast lead time ℓ increases. We shall see that this property is characteristic of the forecast error variance for all *nonstationary* ARIMA processes.

ARMA(p,q)

For the general stationary ARMA(p,q) model, the difference equation form for computing forecasts is given by

$$\begin{aligned}
\hat{Y}_t(\ell) &= \phi_1 \hat{Y}_t(\ell-1) + \phi_2 \hat{Y}_t(\ell-2) + \dots + \phi_p \hat{Y}_t(\ell-p) + \theta_0 \\
&\quad - \theta_1 E(e_{t+\ell-1}|Y_1, Y_2, \dots, Y_t) - \theta_2 E(e_{t+\ell-2}|Y_1, Y_2, \dots, Y_t) \quad (9.3.28) \\
&\quad \dots - \theta_q E(e_{t+\ell-q}|Y_1, Y_2, \dots, Y_t)
\end{aligned}$$

where

$$E(e_{t+j}|Y_1, Y_2, \dots, Y_t) = \begin{cases} 0 & \text{for } j > 0 \\ e_{t+j} & \text{for } j \leq 0 \end{cases} \quad (9.3.29)$$

We note that $\hat{Y}_t(j)$ is a true forecast for $j > 0$, but for $j \leq 0$, $\hat{Y}_t(j) = Y_{t+j}$. As in Equation (9.3.20) on page 197, Equation (9.3.29) involves some minor approximation. For an invertible model, Equation (4.5.5) on page 80 shows that, using the π -weights, e_t can be expressed as a linear combination of the infinite sequence $Y_t, Y_{t-1}, Y_{t-2}, \dots$. However, the π -weights die out exponentially fast, and the approximation assumes that π_j is negligible for $j > t - q$.

As an example, consider an ARMA(1,1) model. We have

$$\hat{Y}_t(1) = \phi Y_t + \theta_0 - \theta e_t \quad (9.3.30)$$

with

$$\hat{Y}_t(2) = \phi \hat{Y}_t(1) + \theta_0$$

and, more generally,

$$\hat{Y}_t(\ell) = \phi \hat{Y}_t(\ell-1) + \theta_0 \text{ for } \ell \geq 2 \quad (9.3.31)$$

using Equation (9.3.30) to get the recursion started.

Equations (9.3.30) and (9.3.31) can be rewritten in terms of the process mean and then solved by iteration to get the alternative explicit expression

$$\hat{Y}_t(\ell) = \mu + \phi^\ell (Y_t - \mu) - \phi^{\ell-1} e_t \text{ for } \ell \geq 1 \quad (9.3.32)$$

As Equations (9.3.28) and (9.3.29) indicate, the noise terms $e_{t-(q-1)}, \dots, e_{t-1}, e_t$ appear directly in the computation of the forecasts for leads $\ell = 1, 2, \dots, q$. However, for $\ell > q$, the autoregressive portion of the difference equation takes over, and we have

$$\hat{Y}_t(\ell) = \phi_1 \hat{Y}_t(\ell-1) + \phi_2 \hat{Y}_t(\ell-2) + \dots + \phi_p \hat{Y}_t(\ell-p) + \theta_0 \text{ for } \ell > q \quad (9.3.33)$$

Thus the general nature of the forecast for long lead times will be determined by the autoregressive parameters $\phi_1, \phi_2, \dots, \phi_p$ (and the constant term, θ_0 , which is related to the mean of the process).

Recalling from Equation (5.3.17) on page 97 that $\theta_0 = \mu(1 - \phi_1 - \phi_2 - \dots - \phi_p)$, we can rewrite Equation (9.3.33) in terms of deviations from μ as

$$\begin{aligned} \hat{Y}_t(\ell) - \mu &= \phi_1 [\hat{Y}_t(\ell-1) - \mu] + \phi_2 [\hat{Y}_t(\ell-2) - \mu] + \dots \\ &\quad + \phi_p [\hat{Y}_t(\ell-p) - \mu] \text{ for } \ell > q \end{aligned} \quad (9.3.34)$$

As a function of lead time ℓ , $\hat{Y}_t(\ell) - \mu$ follows the same Yule-Walker recursion as the autocorrelation function ρ_k of the process (see Equation (4.4.8), page 79). Thus, as in Section 4.3 on page 66 and Section 4.4 on page 77, the roots of the characteristic equation will determine the general behavior of $\hat{Y}_t(\ell) - \mu$ for large lead times. In particular, $\hat{Y}_t(\ell) - \mu$ can be expressed as a linear combination of exponentially decaying terms in ℓ (corresponding to the real roots) and damped sine wave terms (corresponding to the pairs of complex roots).

Thus, for any stationary ARMA model, $\hat{Y}_t(\ell) - \mu$ decays to zero as ℓ increases, and the long-term forecast is simply the process mean μ as given in Equation (9.3.9) on page 195. This agrees with common sense since for stationary ARMA models the dependence dies out as the time span between observations increases, and this dependence is the only reason we can improve on the “naive” forecast of using μ alone.

To argue the validity of Equation (9.3.15) for $e_t(\ell)$ in the present generality, we need to consider a new representation for ARIMA processes. Appendix G shows that any ARIMA model can be written in **truncated linear process** form as

$$Y_{t+\ell} = C_t(\ell) + I_t(\ell) \text{ for } \ell > 1 \quad (9.3.35)$$

where, for our present purposes, we need only know that $C_t(\ell)$ is a certain function of Y_t, Y_{t-1}, \dots and

$$I_t(\ell) = e_{t+\ell} + \psi_1 e_{t+\ell-1} + \psi_2 e_{t+\ell-2} + \dots + \psi_{\ell-1} e_{t+1} \text{ for } \ell \geq 1 \quad (9.3.36)$$

Furthermore, for invertible models with t reasonably large, $C_t(\ell)$ is a certain function of the finite history Y_t, Y_{t-1}, \dots, Y_1 . Thus we have

$$\begin{aligned}\hat{Y}_t(\ell) &= E(C_t(\ell)|Y_1, Y_2, \dots, Y_t) + E(I_t(\ell)|Y_1, Y_2, \dots, Y_t) \\ &= C_t(\ell)\end{aligned}$$

Finally,

$$\begin{aligned}e_t(\ell) &= Y_{t+\ell} - \hat{Y}_t(\ell) \\ &= [C_t(\ell) + I_t(\ell)] - C_t(\ell) \\ &= I_t(\ell) \\ &= e_{t+\ell} + \psi_1 e_{t+\ell-1} + \psi_2 e_{t+\ell-2} + \cdots + \psi_{\ell-1} e_{t+1}\end{aligned}$$

Thus, for a general invertible ARIMA process,

$$E[e_t(\ell)] = 0 \text{ for } \ell \geq 1 \quad (9.3.37)$$

and

$$Var(e_t(\ell)) = \sigma_e^2 \sum_{j=0}^{\ell-1} \psi_j^2 \text{ for } \ell \geq 1 \quad (9.3.38)$$

From Equations (4.1.4) and (9.3.38), we see that for long lead times in stationary ARMA models, we have

$$Var(e_t(\ell)) \approx \sigma_e^2 \sum_{j=0}^{\infty} \psi_j^2$$

or

$$Var(e_t(\ell)) \approx \gamma_0 \text{ for large } \ell \quad (9.3.39)$$

Nonstationary Models

As the random walk shows, forecasting for nonstationary ARIMA models is quite similar to forecasting for stationary ARMA models, but there are some striking differences. Recall from Equation (5.2.2) on page 92 that an ARIMA($p, 1, q$) model can be written as a nonstationary ARMA($p+1, q$) model. We shall write this as

$$\begin{aligned}Y_t &= \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \varphi_3 Y_{t-3} + \cdots + \varphi_p Y_{t-p} + \varphi_{p+1} Y_{t-p-1} \\ &\quad + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q}\end{aligned} \quad (9.3.40)$$

where the script coefficients φ are directly related to the block ϕ coefficients. In particular,

$$\left. \begin{array}{l} \varphi_1 = 1 + \phi_1, \varphi_j = \phi_j - \phi_{j-1} \text{ for } j = 1, 2, \dots, p \\ \text{and} \\ \varphi_{p+1} = -\phi_p \end{array} \right\} \quad (9.3.41)$$

For a general order of differencing d , we would have $p+d$ of the φ coefficients.

From this representation, we can immediately extend Equations (9.3.28), (9.3.29), and (9.3.30) on page 199 to cover the nonstationary cases by replacing p by $p+d$ and ϕ_j by φ_j .

As an example of the necessary calculations, consider the ARIMA(1,1,1) case. Here

$$Y_t - Y_{t-1} = \phi(Y_{t-1} - Y_{t-2}) + \theta_0 + e_t - \theta e_{t-1}$$

so that

$$Y_t = (1 + \phi)Y_{t-1} - \phi Y_{t-2} + \theta_0 + e_t - \theta e_{t-1}$$

Thus

$$\left. \begin{array}{l} \hat{Y}_t(1) = (1 + \phi)Y_t - \phi Y_{t-1} + \theta_0 - \theta e_t \\ \hat{Y}_t(2) = (1 + \phi)\hat{Y}_t(1) - \phi Y_t + \theta_0 \\ \text{and} \\ \hat{Y}_t(\ell) = (1 + \phi)\hat{Y}_t(\ell-1) - \phi \hat{Y}_t(\ell-2) + \theta_0 \end{array} \right\} \quad (9.3.42)$$

For the general invertible ARIMA model, the truncated linear process representation given in Equations (9.3.35) and (9.3.36) and the calculations following these equations show that we can write

$$e_t(\ell) = e_{t+\ell} + \psi_1 e_{t+\ell-1} + \psi_2 e_{t+\ell-2} + \cdots + \psi_{\ell-1} e_{t+1} \quad \text{for } \ell \geq 1 \quad (9.3.43)$$

and so

$$E(e_t(\ell)) = 0 \quad \text{for } \ell \geq 1 \quad (9.3.44)$$

and

$$Var(e_t(\ell)) = \sigma_e^2 \sum_{j=0}^{\ell-1} \psi_j^2 \quad \text{for } \ell \geq 1 \quad (9.3.45)$$

However, for nonstationary series, the ψ_j -weights do not decay to zero as j increases. For example, for the random walk model, $\psi_j = 1$ for all j ; for the IMA(1,1) model, $\psi_j = 1 - \theta$ for $j \geq 1$; for the IMA(2,2) case, $\psi_j = 1 + \theta_2 + (1 - \theta_1 - \theta_2)j$ for $j \geq 1$; and for the ARI(1,1) model, $\psi_j = (1 - \phi^{j+1})/(1 - \phi)$ for $j \geq 1$ (see Chapter 5).

Thus, for any nonstationary model, Equation (9.3.45) shows that the forecast error variance will grow without bound as the lead time ℓ increases. This fact should not be too surprising since with nonstationary series the distant future is quite uncertain.

9.4 Prediction Limits

As in all statistical endeavors, in addition to forecasting or predicting the unknown $Y_{t+\ell}$, we would like to assess the precision of our predictions.

Deterministic Trends

For the deterministic trend model with a white noise stochastic component $\{X_t\}$, we recall that

$$\hat{Y}_t(\ell) = \mu_{t+\ell}$$

and

$$\text{Var}(e_t(\ell)) = \text{Var}(X_{t+\ell}) = \gamma_0$$

If the stochastic component is normally distributed, then the forecast error

$$e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = X_{t+\ell} \quad (9.4.1)$$

is also normally distributed. Thus, for a given confidence level $1 - \alpha$, we could use a standard normal percentile, $z_{1-\alpha/2}$, to claim that

$$P\left[-z_{1-\alpha/2} < \frac{Y_{t+\ell} - \hat{Y}_t(\ell)}{\sqrt{\text{Var}(e_t(\ell))}} < z_{1-\alpha/2}\right] = 1 - \alpha$$

or, equivalently,

$$P[\hat{Y}_t(\ell) - z_{1-\alpha/2}\sqrt{\text{Var}(e_t(\ell))} < Y_{t+\ell} < \hat{Y}_t(\ell) + z_{1-\alpha/2}\sqrt{\text{Var}(e_t(\ell))}] = 1 - \alpha$$

Thus we may be $(1 - \alpha)100\%$ confident that the future observation $Y_{t+\ell}$ will be contained within the prediction limits

$$\hat{Y}_t(\ell) \pm z_{1-\alpha/2}\sqrt{\text{Var}(e_t(\ell))} \quad (9.4.2)$$

As a numerical example, consider the monthly average temperature series once more. On page 192, we used the cosine model to predict the June 1976 average temperature as 68.3°F. The estimate of $\sqrt{\text{Var}(e_t(\ell))} = \sqrt{\gamma_0}$ for this model is 3.7°F. Thus 95% prediction limits for the average June 1976 temperature are

$$68.3 \pm 1.96(3.7) = 68.3 \pm 7.252 \text{ or } 61.05^\circ\text{F to } 75.55^\circ\text{F}$$

Readers who are familiar with standard regression analysis will recall that since the forecast involves *estimated* regression parameters, the correct forecast error variance is given by $\gamma_0[1 + (1/n) + c_{n,\ell}]$, where $c_{n,\ell}$ is a certain function of the sample size n and the lead time ℓ . However, it may be shown that for the types of trends that we are considering (namely, cosines and polynomials in time) and for large sample sizes n , the $1/n$ and $c_{n,\ell}$ are both negligible relative to 1. For example, with a cosine trend of period 12 over $N = n/12$ years, we have that $c_{n,\ell} = 2/n$; thus the correct forecast error variance is

$\gamma_0[1 + (3/n)]$ rather than our approximate γ_0 . For the linear time trend model, it can be shown that $c_{n,\ell} = 3(n + 2\ell - 1)^2/[n(n^2 - 1)] \approx 3/n$ for moderate lead ℓ and large n . Thus, again our approximation seems justified.

ARIMA Models

If the white noise terms $\{e_t\}$ in a general ARIMA series each arise independently from a normal distribution, then from Equation (9.3.43) on page 202, the forecast error $e_t(\ell)$ will also have a normal distribution, and the steps leading to Equation (9.4.2) remain valid. However, in contrast to the deterministic trend model, recall that in the present case

$$\text{Var}(e_t(\ell)) = \sigma_e^2 \sum_{j=0}^{\ell-1} \psi_j^2$$

In practice, σ_e^2 will be unknown and must be estimated from the observed time series. The necessary ψ -weights are, of course, also unknown since they are certain functions of the unknown ϕ 's and θ 's. For large sample sizes, these estimations will have little effect on the actual prediction limits given above.

As a numerical example, consider the AR(1) model that we estimated for the industrial color property series. From Exhibit 9.1 on page 194, we use $\phi = 0.5705$, $\mu = 74.3293$, and $\sigma_e^2 = 24.8$. For an AR(1) model, we recall Equation (9.3.16) on page 196

$$\text{Var}(e_t(\ell)) = \sigma_e^2 \left[\frac{1 - \phi^{2\ell}}{1 - \phi^2} \right]$$

For a one-step-ahead prediction, we have

$$70.14793 \pm 1.96\sqrt{24.8} = 70.14793 \pm 9.760721 \text{ or } 60.39 \text{ to } 79.91$$

Two steps ahead, we obtain

$$71.86072 \pm 11.88343 \text{ or } 60.71 \text{ to } 83.18$$

Notice that this prediction interval is wider than the previous interval. Forecasting ten steps ahead leads to

$$74.173934 \pm 11.88451 \text{ or } 62.42 \text{ to } 86.19$$

By lead 10, both the forecast and the forecast limits have settled down to their long-lead values.

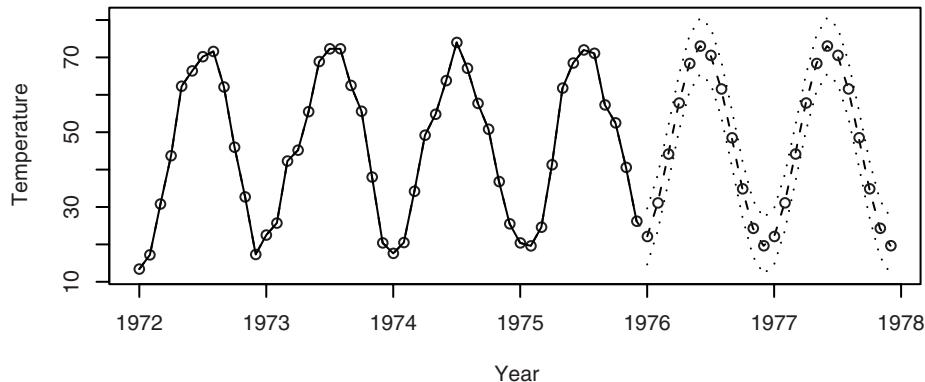
9.5 Forecasting Illustrations

Rather than showing forecast and forecast limit calculations, it is often more instructive to display appropriate plots of the forecasts and their limits.

Deterministic Trends

Exhibit 9.2 displays the last four years of the average monthly temperature time series together with forecasts and 95% forecast limits for two additional years. Since the model fits quite well with a relatively small error variance, the forecast limits are quite close to the fitted trend forecast.

Exhibit 9.2 Forecasts and Limits for the Temperature Cosine Trend

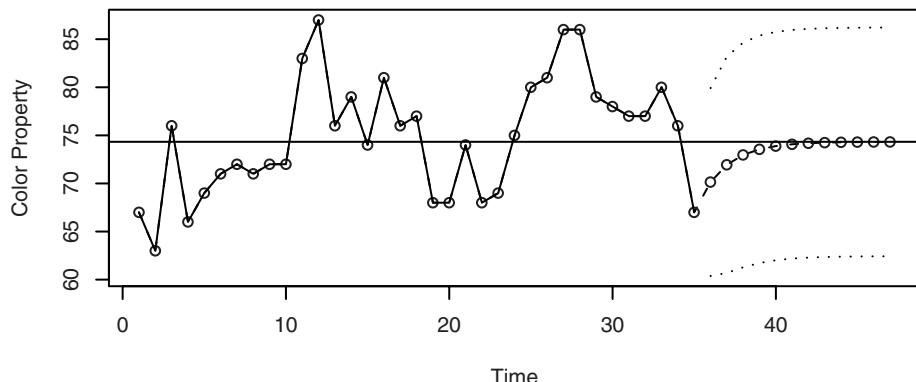


```
> data(tempdub)
> tempdub1=ts(c(tempdub, rep(NA, 24)), start=start(tempdub),
+   freq=frequency(tempdub))
> har.=harmonic(tempdub, 1)
> m5.tempdub=arima(tempdub, order=c(0, 0, 0), xreg=har.)
> newhar.=harmonic(ts(rep(1, 24), start=c(1976, 1), freq=12), 1)
> win.graph(width=4.875, height=2.5, pointsize=8)
> plot(m5.tempdub, n.ahead=24, n1=c(1972, 1), newxreg=newhar.,
+   type='b', ylab='Temperature', xlab='Year'))
```

ARIMA Models

We use the industrial color property series as our first illustration of ARIMA forecasting. Exhibit 9.3 displays this series together with forecasts out to lead time 12 with the upper and lower 95% prediction limits for those forecasts. In addition, a horizontal line at the estimate for the process mean is shown. Notice how the forecasts approach the mean exponentially as the lead time increases. Also note how the prediction limits increase in width.

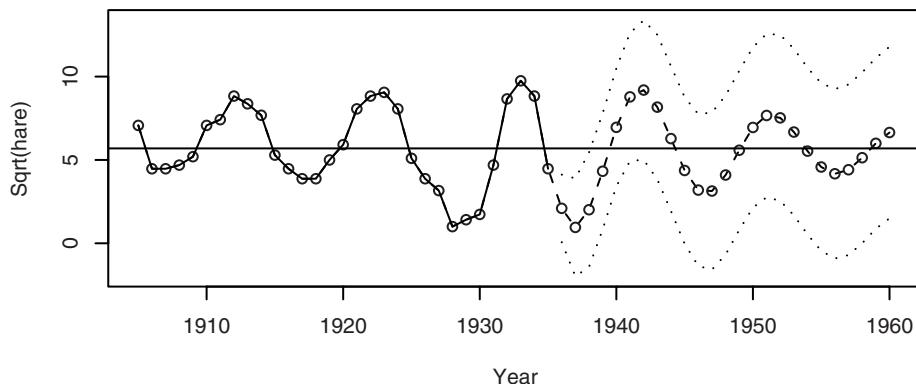
Exhibit 9.3 Forecasts and Forecast Limits for the AR(1) Model for Color



```
> data(color)
> m1.color=arima(color,order=c(1,0,0))
> plot(m1.color,n.ahead=12,type='b',xlab='Time',
      ylab='Color Property')
> abline(h=coef(m1.color)[names(coef(m1.color))=='intercept'])
```

The Canadian hare abundance series was fitted by working with the square root of the abundance numbers and then fitting an AR(3) model. Notice how the forecasts mimic the approximate cycle in the actual series even when we forecast with a lead time out to 25 years in Exhibit 9.4.

Exhibit 9.4 Forecasts from an AR(3) Model for Sqrt(Hare)



```
> data(hare)
> m1.hare=arima(sqrt(hare),order=c(3,0,0))
> plot(m1.hare, n.ahead=25,type='b',
       xlab='Year',ylab='Sqrt(hare)')
> abline(h=coef(m1.hare)[names(coef(m1.hare))=='intercept'])
```

9.6 Updating ARIMA Forecasts

Suppose we are forecasting a monthly time series. Our last observation is, say, for February, and we forecast for March, April, and May. As time goes by, the actual value for March becomes available. With this new value in hand, we would like to update or revise (and, one hopes, improve) our forecasts for April and May. Of course, we could compute new forecasts from scratch. However, there is a simpler way.

For a general forecast origin t and lead time $\ell + 1$, our original forecast is denoted $\hat{Y}_t(\ell + 1)$. Once the observation at time $t + 1$ becomes available, we would like to update our forecast as $\hat{Y}_{t+1}(\ell)$. Equations (9.3.35) and (9.3.36) on page 200 yield

$$Y_{t+\ell+1} = C_t(\ell+1) + e_{t+\ell+1} + \psi_1 e_{t+\ell} + \psi_2 e_{t+\ell-1} + \cdots + \psi_\ell e_{t+1}$$

Since $C_t(\ell+1)$ and e_{t+1} are functions of Y_{t+1}, Y_t, \dots , whereas $e_{t+\ell+1}, e_{t+\ell}, \dots, e_{t+2}$ are independent of Y_{t+1}, Y_t, \dots , we quickly obtain the expression

$$\hat{Y}_{t+1}(\ell) = C_t(\ell+1) + \psi_\ell e_{t+1}$$

However, $\hat{Y}_t(\ell+1) = C_t(\ell+1)$, and, of course, $e_{t+1} = Y_{t+1} - \hat{Y}_t(1)$. Thus we have the general **updating equation**

$$\hat{Y}_{t+1}(\ell) = \hat{Y}_t(\ell+1) + \psi_\ell [Y_{t+1} - \hat{Y}_t(1)] \quad (9.6.1)$$

Notice that $[Y_{t+1} - \hat{Y}_t(1)]$ is the actual forecast error at time $t + 1$ once Y_{t+1} has been observed.

As a numerical example, consider the industrial color property time series. Following Exhibit 9.1 on page 194, we fit an AR(1) model to forecast one step ahead as $\hat{Y}_{35}(1) = 70.096$ and two steps ahead as $\hat{Y}_{35}(2) = 71.86072$. If now the next color value becomes available as $Y_{t+1} = Y_{36} = 65$, then we update the forecast for time $t = 37$ as

$$\hat{Y}_{t+1}(1) = \hat{Y}_{36}(1) = 71.86072 + 0.5705(65 - 70.096) = 68.953452$$

9.7 Forecast Weights and Exponentially Weighted Moving Averages

For ARIMA models without moving average terms, it is clear how the forecasts are explicitly determined from the observed series Y_t, Y_{t-1}, \dots, Y_1 . However, for any model with $q > 0$, the noise terms appear in the forecasts, and the nature of the forecasts explicitly in terms of Y_t, Y_{t-1}, \dots, Y_1 is hidden. To bring out this aspect of the forecasts, we return to the inverted form of any invertible ARIMA process, namely

$$Y_t = \pi_1 Y_{t-1} + \pi_2 Y_{t-2} + \pi_3 Y_{t-3} + \cdots + e_t$$

(See Equation (4.5.5) on page 80.) Thus we can also write

$$Y_{t+1} = \pi_1 Y_t + \pi_2 Y_{t-1} + \pi_3 Y_{t-2} + \cdots + e_{t+1}$$

Taking conditional expectations of both sides, given Y_t, Y_{t-1}, \dots, Y_1 , we obtain

$$\hat{Y}_t(1) = \pi_1 Y_t + \pi_2 Y_{t-1} + \pi_3 Y_{t-2} + \cdots \quad (9.7.1)$$

(We are assuming the t is sufficiently large and/or that the π -weights die out sufficiently quickly so that π_t, π_{t+1}, \dots are all negligible.)

For any invertible ARIMA model, the π -weights can be calculated recursively from the expressions

$$\pi_j = \begin{cases} \sum_{i=1}^{\min(j, q)} \theta_i \pi_{j-i} + \varphi_j & \text{for } 1 \leq j \leq p+d \\ \sum_{i=1}^{\min(j, q)} \theta_i \pi_{j-i} & \text{for } j > p+d \end{cases} \quad (9.7.2)$$

with initial value $\pi_0 = -1$. (Compare this with Equations (4.4.7) on page 79 for the ψ -weights.)

Consider in particular the nonstationary IMA(1,1) model

$$Y_t = Y_{t-1} + e_t - \theta e_{t-1}$$

Here $p = 0, d = 1, q = 1$, with $\varphi_1 = 1$; thus

$$\begin{aligned} \pi_1 &= \theta \pi_0 + 1 = 1 - \theta \\ \pi_2 &= \theta \pi_1 = \theta(1 - \theta) \end{aligned}$$

and, generally,

$$\pi_j = \theta \pi_{j-1} \text{ for } j > 1$$

Thus we have explicitly

$$\pi_j = (1 - \theta) \theta^{j-1} \text{ for } j \geq 1 \quad (9.7.3)$$

so that, from Equation (9.7.1), we can write

$$\hat{Y}_t(1) = (1 - \theta) Y_t + (1 - \theta) \theta Y_{t-1} + (1 - \theta) \theta^2 Y_{t-2} + \cdots \quad (9.7.4)$$

In this case, the π -weights *decrease exponentially*, and furthermore,

$$\sum_{j=1}^{\infty} \pi_j = (1 - \theta) \sum_{j=1}^{\infty} \theta^{j-1} = \frac{1 - \theta}{1 - \theta} = 1$$

Thus $\hat{Y}_t(1)$ is called an **exponentially weighted moving average (EWMA)**.

Simple algebra shows that we can also write

$$\hat{Y}_t(1) = (1 - \theta) Y_t + \theta \hat{Y}_{t-1}(1) \quad (9.7.5)$$

and

$$\hat{Y}_t(1) = \hat{Y}_{t-1}(1) + (1-\theta)[Y_t - \hat{Y}_{t-1}(1)] \quad (9.7.6)$$

Equations (9.7.5) and (9.7.6) show how to update forecasts from origin $t-1$ to origin t , and they express the result as a linear combination of the new observation and the old forecast or in terms of the old forecast and the last observed forecast error.

Using EWMA to forecast time series has been advocated, mostly on an ad hoc basis, for a number of years; see Brown (1962) and Montgomery and Johnson (1976).

The parameter $1 - \theta$ is called the **smoothing constant** in EWMA literature, and its selection (estimation) is often quite arbitrary. From the ARIMA model-building approach, we let the data indicate whether an IMA(1,1) model is appropriate for the series under consideration. If so, we then estimate θ in an efficient manner and compute an EWMA forecast that we are confident is the minimum mean square error forecast. A comprehensive treatment of exponential smoothing methods and their relationships with ARIMA models is given in Abraham and Ledolter (1983).

9.8 Forecasting Transformed Series

Differencing

Suppose we are interested in forecasting a series whose model involves a first difference to achieve stationarity. Two methods of forecasting can be considered:

1. forecasting the original nonstationary series, for example by using the difference equation form of Equation (9.3.28) on page 199, with ϕ 's replaced by φ 's throughout, or
2. forecasting the stationary differenced series $W_t = Y_t - Y_{t-1}$ and then “undoing” the difference by summing to obtain the forecast in original terms.

We shall show that both methods lead to the same forecasts. This follows essentially because differencing is a *linear* operation and because conditional expectation of a linear combination is the same linear combination of the conditional expectations.

Consider in particular the IMA(1,1) model. Basing our work on the original nonstationary series, we forecast as

$$\hat{Y}_t(1) = Y_t - \theta e_t \quad (9.8.1)$$

and

$$\hat{Y}_t(\ell) = \hat{Y}_t(\ell-1) \text{ for } \ell > 1 \quad (9.8.2)$$

Consider now the differenced stationary MA(1) series $W_t = Y_t - Y_{t-1}$. We would forecast $W_{t+\ell}$ as

$$\hat{W}_t(1) = -\theta e_t \quad (9.8.3)$$

and

$$\hat{W}_t(\ell) = 0 \text{ for } \ell > 1 \quad (9.8.4)$$

However, $\hat{W}_t(1) = \hat{Y}_t(1) - Y_t$; thus $\hat{W}_t(1) = -\theta e_t$ is equivalent to $\hat{Y}_t(1) = Y_t - \theta e_t$ as before. Similarly, $\hat{W}_t(\ell) = \hat{Y}_t(\ell) - \hat{Y}_t(\ell-1)$, and Equation (9.8.4) becomes Equation (9.8.2), as we have claimed.

The same result would apply to any model involving differences of any order and indeed to any type of *linear* transformation with constant coefficients. (Certain linear transformations other than differencing may be applicable to seasonal time series. See Chapter 10.)

Log Transformations

As we saw earlier, it is frequently appropriate to model the logarithms of the original series—a nonlinear transformation. Let Y_t denote the original series value and let $Z_t = \log(Y_t)$. It can be shown that we always have

$$E(Y_{t+\ell}|Y_r, Y_{t-1}, \dots, Y_1) \geq \exp[E(Z_{t+\ell}|Z_r, Z_{t-1}, \dots, Z_1)] \quad (9.8.5)$$

with equality holding only in trivial cases. Thus, the naive forecast $\exp[\hat{Z}_t(\ell)]$ is *not* the minimum mean square error forecast of $Y_{t+\ell}$. To evaluate the minimum mean square error forecast in original terms, we shall find the following fact useful: If X has a normal distribution with mean μ and variance σ^2 , then

$$E[\exp(X)] = \exp\left[\mu + \frac{\sigma^2}{2}\right]$$

(This follows, for example, from the moment-generating function for X .) In our application

$$\mu = E(Z_{t+\ell}|Z_r, Z_{t-1}, \dots, Z_1)$$

and

$$\begin{aligned} \sigma^2 &= \text{Var}(Z_{t+\ell}|Z_r, Z_{t-1}, \dots, Z_1) \\ &= \text{Var}[e_t(\ell) + C_t(\ell)|Z_r, Z_{t-1}, \dots, Z_1] \\ &= \text{Var}[e_t(\ell)|Z_r, Z_{t-1}, \dots, Z_1] + \text{Var}[C_t(\ell)|Z_r, Z_{t-1}, \dots, Z_1] \\ &= \text{Var}[e_t(\ell)|Z_r, Z_{t-1}, \dots, Z_1] \\ &= \text{Var}[e_t(\ell)] \end{aligned}$$

These follow from Equations (9.3.35) and (9.3.36) (applied to Z_t) and the fact that $C_t(\ell)$ is a function of Z_t, Z_{t-1}, \dots , whereas $e_t(\ell)$ is independent of Z_t, Z_{t-1}, \dots . Thus the minimum mean square error forecast in the original series is given by

$$\exp\left\{\hat{Z}_t(\ell) + \frac{1}{2}\text{Var}[e_t(\ell)]\right\} \quad (9.8.6)$$

Throughout our discussion of forecasting, we have assumed that minimum mean square forecast error is the criterion of choice. For normally distributed variables, this is an

excellent criterion. However, if Z_t has a normal distribution, then $Y_t = \exp(Z_t)$ has a log-normal distribution, for which a different criterion may be desirable. In particular, since the log-normal distribution is asymmetric and has a long right tail, a criterion based on the mean absolute error may be more appropriate. For this criterion, the optimal forecast is the **median** of the distribution of $Z_{t+\ell}$ conditional on Z_t, Z_{t-1}, \dots, Z_1 . Since the log transformation preserves medians and since, for a normal distribution, the mean and median are identical, the naive forecast $\hat{Y}_t(\ell)$ is the optimal forecast for $Y_{t+\ell}$ in the sense that it minimizes the mean absolute forecast error.

9.9 Summary of Forecasting with Certain ARIMA Models

Here we bring together various forecasting results for special ARIMA models.

$$\textbf{AR(1): } Y_t = \mu + \phi(Y_{t-1} - \mu) + e_t$$

$$\hat{Y}_t(\ell) = \mu + \phi[\hat{Y}_t(\ell-1) - \mu] \quad \text{for } \ell \geq 1$$

$$= \mu + \phi^\ell(Y_t - \mu) \quad \text{for } \ell \geq 1$$

$$\hat{Y}_t(\ell) \approx \mu \quad \text{for large } \ell$$

$$e_t(\ell) = e_{t+\ell} + \phi e_{t+\ell-1} + \cdots + \phi^{\ell-1} e_{t+1}$$

$$Var(e_t(\ell)) = \sigma_e^2 \left[\frac{1 - \phi^{2\ell}}{1 - \phi^2} \right]$$

$$Var(e_t(\ell)) \approx \frac{\sigma_e^2}{1 - \phi^2} = \gamma_0 \quad \text{for large } \ell$$

$$\psi_j = \phi^j \quad \text{for } j > 0$$

$$\textbf{MA(1): } Y_t = \mu + e_t - \theta e_{t-1}$$

$$\hat{Y}_t(1) = \mu - \theta e_t$$

$$\hat{Y}_t(\ell) = \mu \quad \text{for } \ell > 1$$

$$e_t(1) = e_{t+1}$$

$$e_t(\ell) = e_{t+\ell} - \theta e_{t+\ell-1} \quad \text{for } \ell > 1$$

$$Var(e_t(\ell)) = \begin{cases} \sigma_e^2 & \text{for } \ell = 1 \\ \sigma_e^2(1 + \theta^2) & \text{for } \ell > 1 \end{cases}$$

$$\psi_j = \begin{cases} -\theta & \text{for } j = 1 \\ 0 & \text{for } j > 1 \end{cases}$$

IMA (1,1) with Constant Term: $Y_t = Y_{t-1} + \theta_0 + e_t - \theta e_{t-1}$

$$\begin{aligned}\hat{Y}_t(\ell) &= \hat{Y}_t(\ell-1) + \theta_0 - \theta e_t \\ &= Y_t + \ell\theta_0 - \theta e_t\end{aligned}$$

$$\hat{Y}_t(1) = (1-\theta)Y_t + (1-\theta)\theta Y_{t-1} + (1-\theta)\theta^2 Y_{t-2} + \dots \quad (\text{the EWMA for } \theta_0 = 0)$$

$$e_t(\ell) = e_{t+\ell} + (1-\theta)e_{t+\ell-1} + (1-\theta)e_{t+\ell-2} + \dots + (1-\theta)e_{t+1} \quad \text{for } \ell \geq 1$$

$$Var(e_t(\ell)) = \sigma_e^2 [1 + (\ell-1)(1-\theta)^2]$$

$$\psi_j = 1 - \theta \quad \text{for } j > 0$$

Note that if $\theta_0 \neq 0$, the forecasts follow a straight line with slope θ_0 , but if $\theta_0 = 0$, which is the usual case, then the forecast is the same for all lead times, namely

$$\hat{Y}_t(\ell) = Y_t - \theta e_t$$

IMA(2,2): $Y_t = 2Y_{t-1} - Y_{t-2} + \theta_0 + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}$

$$\left. \begin{aligned}\hat{Y}_t(1) &= 2Y_t - Y_{t-1} + \theta_0 - \theta_1 e_t - \theta_2 e_{t-1} \\ \hat{Y}_t(2) &= 2\hat{Y}_t(1) - Y_t + \theta_0 - \theta_2 e_t \\ \hat{Y}_t(\ell) &= 2\hat{Y}_t(\ell-1) - \hat{Y}_t(\ell-2) + \theta_0\end{aligned} \right\} \quad (9.9.1)$$

$$\hat{Y}_t(\ell) = A + B\ell + \frac{\theta_0}{2}\ell^2 \quad (9.9.2)$$

where

$$A = 2\hat{Y}_t(1) - \hat{Y}_t(2) + \theta_0 \quad (9.9.3)$$

and

$$B = \hat{Y}_t(2) - \hat{Y}_t(1) - \frac{3}{2}\theta_0 \quad (9.9.4)$$

If $\theta_0 \neq 0$, the forecasts follow a quadratic curve in ℓ , but if $\theta_0 = 0$, the forecasts form a straight line with slope $\hat{Y}_t(2) - \hat{Y}_t(1)$ and will pass through the two initial forecasts $\hat{Y}_t(1)$ and $\hat{Y}_t(2)$. It can be shown that $Var(e_t(\ell))$ is a certain cubic function of ℓ ; see Box, Jenkins, and Reinsel (1994, p. 156). We also have

$$\psi_j = 1 + \theta_2 + (1 - \theta_1 - \theta_2)j \quad \text{for } j > 0 \quad (9.9.5)$$

It can also be shown that forecasting the special case with $\theta_1 = 2\omega$ and $\theta_2 = -\omega^2$ is equivalent to so-called **double exponential smoothing** with smoothing constant $1 - \omega$; see Abraham and Ledolter (1983).

9.10 Summary

Forecasting or predicting future as yet unobserved values is one of the main reasons for developing time series models. Methods discussed in this chapter are all based on minimizing the mean square forecasting error. When the model is simply deterministic trend plus zero mean white noise error, forecasting amounts to extrapolating the trend. However, if the model contains autocorrelation, the forecasts exploit the correlation to produce better forecasts than would otherwise be obtained. We showed how to do this with ARIMA models and investigated the computation and properties of the forecasts. In special cases, the computation and properties of the forecasts are especially interesting and we presented them separately. Prediction limits are especially important to assess the potential accuracy (or otherwise) of the forecasts. Finally, we addressed the problem of forecasting time series for which the models involve transformation of the original series.

EXERCISES

- 9.1 For an AR(1) model with $Y_t = 12.2$, $\phi = -0.5$, and $\mu = 10.8$,
 - (a) Find $\hat{Y}_t(1)$.
 - (b) Calculate $\hat{Y}_t(2)$ in two different ways.
 - (c) Calculate $\hat{Y}_t(10)$.
- 9.2 Suppose that annual sales (in millions of dollars) of the Acme Corporation follow the AR(2) model $Y_t = 5 + 1.1Y_{t-1} - 0.5Y_{t-2} + e_t$ with $\sigma_e^2 = 2$.
 - (a) If sales for 2005, 2006, and 2007 were \$9 million, \$11 million, and \$10 million, respectively, forecast sales for 2008 and 2009.
 - (b) Show that $\psi_1 = 1.1$ for this model.
 - (c) Calculate 95% prediction limits for your forecast in part (a) for 2006.
 - (d) If sales in 2006 turn out to be \$12 million, update your forecast for 2007.
- 9.3 Using the estimated cosine trend on page 192:
 - (a) Forecast the average monthly temperature in Dubuque, Iowa, for April 1976.
 - (b) Find a 95% prediction interval for that April forecast. (The estimate of $\sqrt{\gamma_0}$ for this model is 3.719°F .)
 - (c) What is the forecast for April, 1977? For April 2009?
- 9.4 Using the estimated cosine trend on page 192:
 - (a) Forecast the average monthly temperature in Dubuque, Iowa, for May 1976.
 - (b) Find a 95% prediction interval for that May 1976 forecast. (The estimate of $\sqrt{\gamma_0}$ for this model is 3.719°F .)

- 9.5** Using the seasonal means model *without* an intercept shown in Exhibit 3.3 on page 32:
- Forecast the average monthly temperature in Dubuque, Iowa, for April, 1976.
 - Find a 95% prediction interval for that April forecast. (The estimate of $\sqrt{\gamma_0}$ for this model is 3.419°F.)
 - Compare your forecast with the one obtained in Exercise 9.3.
 - What is the forecast for April 1977? April 2009?
- 9.6** Using the seasonal means model *with* an intercept shown in Exhibit 3.4 on page 33:
- Forecast the average monthly temperature in Dubuque, Iowa, for April 1976.
 - Find a 95% prediction interval for that April forecast. (The estimate of $\sqrt{\gamma_0}$ for this model is 3.419°F.)
 - Compare your forecast with the one obtained in Exercise 9.5.
- 9.7** Using the seasonal means model *with* an intercept shown in Exhibit 3.4 on page 33
- Forecast the average monthly temperature in Dubuque, Iowa, for January 1976.
 - Find a 95% prediction interval for that January forecast. (The estimate of $\sqrt{\gamma_0}$ for this model is 3.419°F.)
- 9.8** Consider the monthly electricity generation time series shown in Exhibit 5.8 on page 99. The data are in the file named *electricity*.
- Fit a deterministic trend model containing seasonal means together with a linear time trend to the logarithms of the electricity values.
 - Plot the last five years of the series together with two years of forecasts and the 95% forecast limits. Interpret the plot.
- 9.9** Simulate an AR(1) process with $\phi = 0.8$ and $\mu = 100$. Simulate 48 values but set aside the last 8 values to compare forecasts to actual values.
- Using the first 40 values of the series, find the values for the maximum likelihood estimates of ϕ and μ .
 - Using the estimated model, forecast the next eight values of the series. Plot the series together with the eight forecasts. Place a horizontal line at the estimate of the process mean.
 - Compare the eight forecasts with the actual values that you set aside.
 - Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and the same sample size.
- 9.10** Simulate an AR(2) process with $\phi_1 = 1.5$, $\phi_2 = -0.75$, and $\mu = 100$. Simulate 52 values but set aside the last 12 values to compare forecasts to actual values.
- Using the first 40 values of the series, find the values for the maximum likelihood estimates of the ϕ 's and μ .
 - Using the estimated model, forecast the next 12 values of the series. Plot the series together with the 12 forecasts. Place a horizontal line at the estimate of

- the process mean.
- (c) Compare the 12 forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.
- 9.11** Simulate an MA(1) process with $\theta = 0.6$ and $\mu = 100$. Simulate 36 values but set aside the last 4 values to compare forecasts to actual values.
- (a) Using the first 32 values of the series, find the values for the maximum likelihood estimates of the θ and μ .
 - (b) Using the estimated model, forecast the next four values of the series. Plot the series together with the four forecasts. Place a horizontal line at the estimate of the process mean.
 - (c) Compare the four forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.
- 9.12** Simulate an MA(2) process with $\theta_1 = 1$, $\theta_2 = -0.6$, and $\mu = 100$. Simulate 36 values but set aside the last 4 values with compare forecasts to actual values.
- (a) Using the first 32 values of the series, find the values for the maximum likelihood estimates of the θ 's and μ .
 - (b) Using the estimated model, forecast the next four values of the series. Plot the series together with the four forecasts. Place a horizontal line at the estimate of the process mean.
 - (c) What is special about the forecasts at lead times 3 and 4?
 - (d) Compare the four forecasts with the actual values that you set aside.
 - (e) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (f) Repeat parts (a) through (e) with a new simulated series using the same values of the parameters and same sample size.
- 9.13** Simulate an ARMA(1,1) process with $\phi = 0.7$, $\theta = -0.5$, and $\mu = 100$. Simulate 50 values but set aside the last 10 values to compare forecasts with actual values.
- (a) Using the first 40 values of the series, find the values for the maximum likelihood estimates of ϕ , θ , and μ .
 - (b) Using the estimated model, forecast the next ten values of the series. Plot the series together with the ten forecasts. Place a horizontal line at the estimate of the process mean.
 - (c) Compare the ten forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.

- 9.14** Simulate an IMA(1,1) process with $\theta = 0.8$ and $\theta_0 = 0$. Simulate 35 values, but set aside the last five values to compare forecasts with actual values.
- (a) Using the first 30 values of the series, find the value for the maximum likelihood estimate of θ .
 - (b) Using the estimated model, forecast the next five values of the series. Plot the series together with the five forecasts. What is special about the forecasts?
 - (c) Compare the five forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.
- 9.15** Simulate an IMA(1,1) process with $\theta = 0.8$ and $\theta_0 = 10$. Simulate 35 values, but set aside the last five values to compare forecasts to actual values.
- (a) Using the first 30 values of the series, find the values for the maximum likelihood estimates of θ and θ_0 .
 - (b) Using the estimated model, forecast the next five values of the series. Plot the series together with the five forecasts. What is special about these forecasts?
 - (c) Compare the five forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.
- 9.16** Simulate an IMA(2,2) process with $\theta_1 = 1$, $\theta_2 = -0.75$, and $\theta_0 = 0$. Simulate 45 values, but set aside the last five values to compare forecasts with actual values.
- (a) Using the first 40 values of the series, find the value for the maximum likelihood estimate of θ_1 and θ_2 .
 - (b) Using the estimated model, forecast the next five values of the series. Plot the series together with the five forecasts. What is special about the forecasts?
 - (c) Compare the five forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.
- 9.17** Simulate an IMA(2,2) process with $\theta_1 = 1$, $\theta_2 = -0.75$, and $\theta_0 = 10$. Simulate 45 values, but set aside the last five values to compare forecasts with actual values.
- (a) Using the first 40 values of the series, find the values for the maximum likelihood estimates of θ_1 , θ_2 , and θ_0 .
 - (b) Using the estimated model, forecast the next five values of the series. Plot the series together with the five forecasts. What is special about these forecasts?
 - (c) Compare the five forecasts with the actual values that you set aside.
 - (d) Plot the forecasts together with 95% forecast limits. Do the actual values fall within the forecast limits?
 - (e) Repeat parts (a) through (d) with a new simulated series using the same values of the parameters and same sample size.

- 9.18** Consider the model $Y_t = \beta_0 + \beta_1 t + X_t$, where $X_t = \phi X_{t-1} + e_t$. We assume that β_0 , β_1 , and ϕ are known. Show that the minimum mean square error forecast ℓ steps ahead can be written as $\hat{Y}_t(\ell) = \beta_0 + \beta_1(t + \ell) + \phi^\ell(Y_t - \beta_0 - \beta_1 t)$.
- 9.19** Verify Equation (9.3.16) on page 196.
- 9.20** Verify Equation (9.3.32) on page 200.
- 9.21** The data file named *deere3* contains 57 consecutive values from a complex machine tool process at Deere & Co. The values given are deviations from a target value in units of ten millionths of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced.
- (a) Using an AR(1) model for this series, forecast the next ten values.
- (b) Plot the series, the forecasts, and 95% forecast limits, and interpret the results.
- 9.22** The data file named *days* contains accounting data from the Winegard Co. of Burlington, Iowa. The data are the number of days until Winegard receives payment for 130 consecutive orders from a particular distributor of Winegard products. (The name of the distributor must remain anonymous for confidentiality reasons.) The time series contains outliers that are quite obvious in the time series plot. Replace each of the unusual values at “times” 63, 106, and 129 with the much more typical value of 35 days.
- (a) Use an MA(2) model to forecast the next ten values of this modified series.
- (b) Plot the series, the forecasts, and 95% forecast limits, and interpret the results.
- 9.23** The time series in the data file *robot* gives the final position in the “*x*-direction” after an industrial robot has finished a planned set of exercises. The measurements are expressed as deviations from a target position. The robot is put through this planned set of exercises in the hope that its behavior is repeatable and thus predictable.
- (a) Use an IMA(1,1) model to forecast five values ahead. Obtain 95% forecast limits also.
- (b) Display the forecasts, forecast limits, and actual values in a graph and interpret the results.
- (c) Now use an ARMA(1,1) model to forecast five values ahead and obtain 95% forecast limits. Compare these results with those obtained in part (a).
- 9.24** Exhibit 9.4 on page 206 displayed the forecasts and 95% forecast limits for the square root of the Canadian hare abundance. The data are in the file named *hare*. Produce a similar plot in original terms. That is, plot the original abundance values together with the squares of the forecasts and squares of the forecast limits.
- 9.25** Consider the seasonal means plus linear time trend model for the logarithms of the monthly electricity generation time series in Exercise 9.8. (The data are in the file named *electricity*.)
- (a) Find the two-year forecasts and forecast limits in original terms. That is, exponentiate (antilog) the results obtained in Exercise 9.8.
- (b) Plot the last five years of the original time series together with two years of forecasts and the 95% forecast limits, all in original terms. Interpret the plot.

Appendix E: Conditional Expectation

If X and Y have joint pdf $f(x,y)$ and we denote the marginal pdf of X by $f(x)$, then the **conditional pdf** of Y given $X = x$ is given by

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

For a given value of x , the conditional pdf has all of the usual properties of a pdf. In particular, the **conditional expectation** of Y given $X = x$ is defined as

$$E(Y|X=x) = \int_{-\infty}^{\infty} yf(y|x)dy$$

As an expected value or mean, the **conditional expectation** of Y given $X = x$ has all of the usual properties. For example,

$$E(aY + bZ + c|X=x) = aE(Y|X=x) + bE(Z|X=x) + c \quad (9.E.1)$$

and

$$E[h(Y)|X=x] = \int_{-\infty}^{\infty} yf(y|x)dx \quad (9.E.2)$$

In addition, several new properties arise:

$$E[h(X)|X=x] = h(x) \quad (9.E.3)$$

That is, given $X = x$, the random variable $h(X)$ can be treated like a constant $h(x)$. More generally,

$$E[h(X, Y)|X=x] = E(h(x, Y)|X=x) \quad (9.E.4)$$

If we set $E(Y|X=x) = g(x)$, then $g(X)$ is a random variable and we can consider $E[g(X)]$. It can be shown that

$$E[g(X)] = E(Y)$$

which is often written as

$$E[E(Y|X)] = E(Y) \quad (9.E.5)$$

If Y and X are independent, then

$$E(Y|X) = E(Y) \quad (9.E.6)$$

Appendix F: Minimum Mean Square Error Prediction

Suppose Y is a random variable with mean μ_Y and variance σ_Y^2 . If our object is to predict Y using only a constant c , what is the *best* choice for c ? Clearly, we must first define *best*. A common (and convenient) criterion is to choose c to minimize the **mean square error of prediction**, that is, to minimize

$$g(c) = E[(Y - c)^2]$$

If we expand $g(c)$, we have

$$g(c) = E(Y^2) - 2cE(Y) + c^2$$

Since $g(c)$ is quadratic in c and opens upward, solving $g'(c) = 0$ will produce the required minimum. We have

$$g'(c) = -2E(Y) + 2c$$

so that the optimal c is

$$c = E(Y) = \mu \quad (9.F.1)$$

Note also that

$$\min_{-\infty < c < \infty} g(c) = E(Y - \mu)^2 = \sigma_Y^2 \quad (9.F.2)$$

Now consider the situation where a second random variable X is available and we wish to use the observed value of X to help predict Y . Let $\rho = \text{Corr}(X, Y)$. We first suppose, for simplicity, that only *linear* functions $a + bX$ can be used for the prediction. The mean square error is then given by

$$g(a, b) = E(Y - a - bX)^2$$

and expanding we gave

$$g(a, b) = E(Y^2) + a^2 + b^2E(X^2) - 2aE(Y) + 2abE(X) - 2bE(XY)$$

This is also quadratic in a and b and opens upward. Thus we can find the point of minimum by solving simultaneous linear equations $\partial g(a, b)/\partial a = 0$ and $\partial g(a, b)/\partial b = 0$. We have

$$\partial g(a, b)/\partial a = 2a - 2E(Y) + 2bE(X) = 0$$

$$\partial g(a, b)/\partial b = 2bE(X^2) + 2aE(X) - 2E(XY) = 0$$

which we rewrite as

$$a + E(X)b = E(Y)$$

$$E(X)a + E(X^2)b = EXY$$

Multiplying the first equation by $E(X)$ and subtracting yields

$$b = \frac{E(XY) - E(X)E(Y)}{E(X^2) - [E(X)]^2} = \frac{\text{Cov}(X, Y)}{\text{Var}(X)} = \rho \frac{\sigma_Y}{\sigma_X} \quad (9.F.3)$$

Then

$$a = E(Y) - bE(X) = \mu_Y - \rho \frac{\sigma_Y}{\sigma_X} \mu_X \quad (9.F.4)$$

If we let \hat{Y} be the minimum mean square error prediction of Y based on a linear function of X , then we can write

$$\hat{Y} = \left[\mu_Y - \rho \frac{\sigma_Y}{\sigma_X} \mu_X \right] + \left[\rho \frac{\sigma_Y}{\sigma_X} \mu_X \right] X \quad (9.F.5)$$

or

$$\left[\frac{\hat{Y} - \mu_Y}{\sigma_Y} \right] = \rho \left[\frac{X - \mu_X}{\sigma_X} \right] \quad (9.F.6)$$

In terms of standardized variables \hat{Y}^* and X^* , we have simply $\hat{Y}^* = \rho X^*$.

Also, using Equations (9.F.3) and (9.F.4), we find

$$\min g(a, b) = \sigma_Y^2(1 - \rho^2) \quad (9.F.7)$$

which provides a proof that $-1 \leq \rho \leq +1$ since $g(a, b) \geq 0$.

If we compare Equation (9.F.7) with Equation (9.F.2), we see that the minimum mean square error obtained when we use a linear function of X to predict Y is reduced by a factor of $1 - \rho^2$ compared with that obtained by ignoring X and simply using the constant μ_Y for our prediction.

Let us now consider the more general problem of predicting Y with an arbitrary function of X . Once more our criterion will be to minimize the mean square error of prediction. We need to choose the function $h(X)$, say, that minimizes

$$E[Y - h(X)]^2 \quad (9.F.8)$$

Using Equation (9.E.5), we can write this as

$$E[Y - h(X)]^2 = E(E\{[Y - h(X)]^2 | X\}) \quad (9.F.9)$$

Using Equation (9.E.4), the inner expectation can be written as

$$E\{[Y - h(X)]^2 | X = x\} = E\{[Y - h(x)]^2 | X = x\} \quad (9.F.10)$$

For each value of x , $h(x)$ is a constant, and we can apply the result of Equation (9.F.1) to the conditional distribution of Y given $X = x$. Thus, for each x , the best choice of $h(x)$ is

$$h(x) = E(Y|X = x) \quad (9.F.11)$$

Since this choice of $h(x)$ minimizes the inner expectation in Equation (9.F.9), it must also provide the overall minimum of Equation (9.F.8). Thus

$$h(X) = E(Y|X) \quad (9.F.12)$$

is the best predictor of Y of all functions of X .

If X and Y have a bivariate normal distribution, it is well-known that

$$E(Y|X) = \mu_Y + \rho \frac{\sigma_Y}{\sigma_X}(X - \mu_X)$$

so that the solutions given in Equations (9.F.12) and (9.F.5) coincide. In this case, the linear predictor is the best of all functions.

More generally, if Y is to be predicted by a function of X_1, X_2, \dots, X_n , then it can be easily argued that the minimum square error predictor is given by

$$E(Y|X_1, X_2, \dots, X_n) \quad (9.F.13)$$

Appendix G: The Truncated Linear Process

Suppose $\{Y_t\}$ satisfies the general ARIMA(p, d, q) model with AR characteristic polynomial $\phi(x)$, MA characteristic polynomial $\theta(x)$, and constant term θ_0 . Then the **truncated linear process** representation for $\{Y_t\}$ is given by

$$Y_{t+\ell} = C_t(\ell) + I_t(\ell) \quad \text{for } \ell \geq 1 \quad (9.G.1)$$

where

$$I_t(\ell) = \sum_{j=0}^{\ell-1} \psi_j e_{t+\ell-j} \quad \text{for } \ell \geq 1 \quad (9.G.2)$$

$$C_t(\ell) = \sum_{i=0}^d A_i \ell^i + \sum_{i=1}^r \sum_{j=0}^{p_i-1} B_{ij} \ell^j (G_i)^{\ell} \quad (9.G.3)$$

and $A_i, B_{ij}, i = 1, 2, \dots, r, j = 1, 2, \dots, p_i$, are constant in ℓ and depend only on Y_t, Y_{t-1}, \dots .[†] As always, the ψ -weights are defined by the identity

$$\phi(x)(1-x)^d(1 + \psi_1 x + \psi_2 x^2 + \dots) = \theta(x) \quad (9.G.4)$$

or

$$\phi(x)(1 + \psi_1 x + \psi_2 x^2 + \dots) = \theta(x) \quad (9.G.5)$$

We shall show that the representation given by Equation (9.G.1) is valid by arguing that, for fixed t , $C_t(\ell)$ is essentially the **complementary function** of the defining difference equation, that is,

$$C_t(\ell) - \varphi_1 C_t(\ell-1) - \varphi_2 C_t(\ell-2) - \dots - \varphi_{p+d} C_t(\ell-p-d) = \theta_0 \quad \text{for } \ell \geq 0 \quad (9.G.6)$$

and that $I_t(\ell)$ is a **particular solution** (without θ_0):

$$\begin{aligned} I_t(\ell) - \varphi_1 I_t(\ell-1) - \varphi_2 I_t(\ell-2) - \dots - \varphi_{p+d} I_t(\ell-p-d) \\ = e_{t+\ell} - \theta_1 e_{t+\ell-1} - \theta_2 e_{t+\ell-2} - \dots - \theta_q e_{t+\ell-q} \quad \text{for } \ell > q \end{aligned} \quad (9.G.7)$$

Since $C_t(\ell)$ contains $p+d$ arbitrary constants (the A 's and the B 's), summing $C_t(\ell)$ and $I_t(\ell)$ yields the general solution of the ARIMA equation. Specific values for the A 's and B 's will be determined by initial conditions on the $\{Y_t\}$ process.

We note that A_d is not arbitrary. We have

$$A_d = \frac{\theta_0}{(1 - \varphi_1 - \varphi_2 - \dots - \varphi_p)d!} \quad (9.G.8)$$

The proof that $C_t(\ell)$ as given by Equation (9.G.2) is the complementary function and satisfies Equation (9.G.6) is a standard result from the theory of difference equations

[†] The only property of the $C_t(\ell)$ that we need is that it depends only on Y_t, Y_{t-1}, \dots .

(see, for example, Goldberg, 1958). We shall show that the particular solution $I_t(\ell)$ defined by Equation (9.G.2) does satisfy Equation (9.G.7).

For convenience of notation, we let $\varphi_j = 0$ for $j > p + d$. Consider the left-hand side of Equation (9.G.7). It can be written as:

$$\left. \begin{aligned} & (\psi_0 e_{t+\ell} + \psi_1 e_{t+\ell-1} + \cdots + \psi_{\ell-1} e_{t+1}) - \varphi_1 (\psi_0 e_{t+\ell-1} + \psi_1 e_{t+\ell-2} + \cdots \\ & \quad + \psi_{\ell-2} e_{t+1}) - \cdots - \varphi_{p+d} (\psi_0 e_{t+\ell-p-d} \\ & \quad + \psi_1 e_{t+\ell-p-d-1} + \cdots + \psi_{\ell-p-d-1} e_{t+1}) \end{aligned} \right\} \quad (9.G.9)$$

Now grouping together common e_t terms and picking off their coefficients, we obtain

Coefficient of $e_{t+\ell-1}$: ψ_0

Coefficient of $e_{t+\ell-2}$: $\psi_1 - \varphi_1 \psi_0$

Coefficient of $e_{t+\ell-3}$: $\psi_2 - \varphi_1 \psi_1 - \varphi_2 \psi_0$

\vdots

Coefficient of e_{t+1} : $\psi_{\ell-1} - \varphi_1 \psi_{\ell-2} - \varphi_2 \psi_{\ell-3} - \cdots - \varphi_{p+d} \psi_{\ell-p-d-1}$

If $\ell > q$, we can match these coefficients to the corresponding coefficients on the right-hand side of Equation (9.G.7) to obtain the relationships

$$\left. \begin{aligned} \psi_0 &= 1 \\ \psi_1 - \varphi_1 \psi_0 &= -\theta_1 \\ \psi_2 - \varphi_1 \psi_1 - \varphi_2 \psi_0 &= -\theta_2 \\ &\vdots \\ \psi_q - \varphi_1 \psi_{q-1} - \varphi_2 \psi_{q-2} - \cdots - \varphi_q \psi_0 &= -\theta_q \\ \psi_{\ell-1} - \varphi_1 \psi_{\ell-2} - \varphi_2 \psi_{\ell-3} - \cdots - \varphi_{p+d} \psi_{\ell-p-d-1} &= 0 \quad \text{for } \ell > q \end{aligned} \right\} \quad (9.G.10)$$

However, by comparing these relationships with Equation (9.G.5), we see that Equations (9.G.10) are precisely the equations defining the ψ -weights and thus Equation (9.G.7) is established as required.

Appendix H: State Space Models

Control theory engineers have developed and successfully used so-called **state space models** and **Kalman filtering** since Kalman published his seminal work in 1960. Recent references include Durbin and Koopman (2001) and Harvey et al. (2004).

Consider a general stationary and invertible ARMA(p, q) process $\{Z_t\}$. Put $m = \max(p, q + 1)$ and define the **state** of the process at time t as the column vector $Z(t)$ of length m whose j th element is the forecast $\hat{Z}(j)$ for $j = 0, 1, 2, \dots, m - 1$, based on Z_t, Z_{t-1}, \dots . Note that the lead element of $Z(t)$ is just $\hat{Z}(0) = Z_t$.

Recall the updating Equation (9.6.1) on page 207, which in the present context can

be written

$$\hat{Z}_{t+1}(\ell) = \hat{Z}_t(\ell+1) + \psi_\ell e_{t+1} \quad (9.H.1)$$

We shall use this expression directly for $\ell = 0, 1, 2, \dots, m-2$. For $\ell = m-1$, we have

$$\begin{aligned} \hat{Z}_{t+1}(m-1) &= \hat{Z}_t(m) + \psi_{m-1} e_{t+1} \\ &= \phi_1 \hat{Z}_t(m-1) + \phi_2 \hat{Z}_t(m-2) + \dots + \phi_p \hat{Z}_t(m-p) + \psi_{m-1} e_{t+1} \end{aligned} \quad (9.H.2)$$

where the last expression comes from Equation (9.3.34) on page 200, with $\mu = 0$.

The matrix formulation of Equations (9.H.1) and (9.H.2) relating $Z(t+1)$ to $Z(t)$ and e_{t+1} , called the **equations of state** (or **Akaike's Markovian representation**), is given as

$$Z(t+1) = FZ(t) + Ge_{t+1} \quad (9.H.3)$$

where

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ & & & & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ \phi_m & \phi_{m-1} & \cdot & \cdot & \cdot & \phi_1 \end{bmatrix} \quad (9.H.4)$$

and

$$G = \begin{bmatrix} 1 \\ \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{m-1} \end{bmatrix} \quad (9.H.5)$$

with $\phi_j = 0$ for $j > p$. Note that the simplicity of Equation (9.H.3) is obtained at the expense of having to deal with vector-valued processes. Because the state space formulation also usually allows for measurement error, we do not observe Z_t directly but only observe Y_t through the **observational equation**

$$Y_t = HZ(t) + \varepsilon_t \quad (9.H.6)$$

where $H = [1, 0, 0, \dots, 0]$ and $\{\varepsilon_t\}$ is another zero-mean white noise process independent of $\{e_t\}$. The special case of *no* measurement error is obtained by setting $\varepsilon_t = 0$ in Equation (9.H.6). Equivalently, this case is obtained by taking $\sigma_\varepsilon^2 = 0$ in subsequent equations. More general state space models allow F , G , and H to be more general, possibly also depending on time.

Evaluation of the Likelihood Function and Kalman Filtering

First a definition: The **covariance matrix** for a vector of random variables \mathbf{X} of dimension $n \times 1$ is defined to be the $n \times n$ matrix whose ij th element is the covariance between the i th and j th components of \mathbf{X} .

If $\mathbf{Y} = \mathbf{AX} + \mathbf{B}$, then it is easily shown that the covariance matrix for \mathbf{Y} is $\mathbf{AV}\mathbf{A}^T$, where \mathbf{V} is the covariance matrix for \mathbf{X} and the superscript T denotes matrix transpose.

Getting back to the Kalman filter, we let $\mathbf{Z}(t+1|t)$ denote the $m \times 1$ vector whose j th component is $E[\hat{Z}_{t+1}(j)|Y_t, Y_{t-1}, \dots, Y_1]$ for $j = 0, 1, 2, \dots, m-1$. Similarly, let $\mathbf{Z}(t|t)$ be the vector whose j th component is $E[\hat{Z}_t(j)|Y_t, Y_{t-1}, \dots, Y_1]$ for $j = 0, 1, 2, \dots, m-1$.

Then, since e_{t+1} is independent of Z_t, Z_{t-1}, \dots , and hence also of Y_t, Y_{t-1}, \dots , we see from Equation (9.H.3) that

$$\mathbf{Z}(t+1|t) = \mathbf{F}\mathbf{Z}(t|t) \quad (9.H.7)$$

Also letting $\mathbf{P}(t+1|t)$ be the covariance matrix for the “forecast error” $\mathbf{Z}(t+1) - \mathbf{Z}(t+1|t)$ and $\mathbf{P}(t|t)$ be the covariance matrix for the “forecast error” $\mathbf{Z}(t) - \mathbf{Z}(t|t)$, we have from Equation (9.H.3) that

$$\mathbf{P}(t+1|t) = \mathbf{F}[\mathbf{P}(t|t)]\mathbf{F}^T + \sigma_e^2 \mathbf{G}\mathbf{G}^T \quad (9.H.8)$$

From the observational equation (Equation (9.H.6)) and then replacing $t+1$ by t ,

$$Y(t+1|t) = \mathbf{HZ}(t+1|t) \quad (9.H.9)$$

where $Y(t+1|t) = E(Y_{t+1}|Y_t, Y_{t-1}, \dots, Y_1)$.

It can now be shown that the following relationships hold (see, for example, Harvey, 1981c):

$$\mathbf{Z}(t+1|t+1) = \mathbf{Z}(t+1|t) + \mathbf{K}(t+1)[Y_{t+1} - Y(t+1|t)] \quad (9.H.10)$$

where

$$\mathbf{K}(t+1) = \mathbf{P}(t+1|t)\mathbf{H}^T [\mathbf{HP}(t+1|t)\mathbf{H}^T + \sigma_e^2]^{-1} \quad (9.H.11)$$

and

$$\mathbf{P}(t+1|(t+1)) = \mathbf{P}(t+1|t) - \mathbf{K}(t+1)\mathbf{HP}(t+1|t) \quad (9.H.12)$$

Collectively, Equations (9.H.10), (9.H.11), and (9.H.12) are referred to as the **Kalman filter equations**. The quantity

$$err_{t+1} = Y_{t+1} - Y(t+1|t) \quad (9.H.13)$$

in Equation (9.H.10) is the prediction error and is independent of (or at least uncorrelated with) the past observations Y_t, Y_{t-1}, \dots . Since we are allowing for measurement error, err_{t+1} is not, in general, the same as e_{t+1} .

From Equations (9.H.13) and (9.H.6), we have

$$\nu_{t+1} = Var(err_{t+1}) = \mathbf{HP}(t+1|t)\mathbf{H}^T + \sigma_e^2 \quad (9.H.14)$$

Now consider the likelihood function for the observed series Y_1, Y_2, \dots, Y_n . From the definition of the conditional probability density function, we can write

$$f(y_1, y_2, \dots, y_n) = f(y_n|y_1, y_2, \dots, y_{n-1})f(y_1, y_2, \dots, y_{n-1})$$

or, by taking logs,

$$\log f(y_1, y_2, \dots, y_n) = \log f(y_1, y_2, \dots, y_{n-1}) + \log f(y_n|y_1, y_2, \dots, y_{n-1}) \quad (9.H.15)$$

Assume now that we are dealing with normal distributions, that is, that $\{e_t\}$ and $\{\varepsilon_t\}$ are normal white noise processes. Then it is known that the distribution of Y_n conditional on $Y_1 = y_1, Y_2 = y_2, \dots, Y_{n-1} = y_{n-1}$, is also normal with mean $y(n|n-1)$ and variance v_n . In the remainder of this section and the next, we write $y(n|n-1)$ for the observed value of $Y(n|n-1)$. The second term on the right-hand side of Equation (9.H.15) can then be written

$$\log f(y_n|y_1, y_2, \dots, y_{n-1}) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log v_n - \frac{1}{2} \frac{[y_n - y(n|n-1)]^2}{v_n}$$

Furthermore, the first term on the right-hand side of Equation (9.H.15) can be decomposed similarly again and again until we have

$$\log f(y_1, y_2, \dots, y_n) = \sum_{t=2}^n \log f(y_t|y_1, y_2, \dots, y_{t-1}) + \log f(y_1) \quad (9.H.16)$$

which then becomes the prediction error decomposition of the likelihood, namely

$$\log f(y_1, y_2, \dots, y_n) = -\frac{n}{2}\log 2\pi - \frac{1}{2} \sum_{t=1}^n v_t - \frac{1}{2} \sum_{t=1}^n \frac{[y_t - y(t|t-1)]^2}{v_t} \quad (9.H.17)$$

with $y(1|0) = 0$ and $v_1 = \text{Var}(Y_1)$.

The overall strategy for computing the likelihood for a given set of parameter values is to use the Kalman filter equations to generate recursively the prediction errors and their variances and then use the prediction error decomposition of the likelihood function. Only one point remains: We need initial values $Z(0|0)$ and $P(0|0)$ to get the recursions started.

The Initial State Covariance Matrix

The initial state vector $Z(0|0)$ will be a vector of zeros for a zero-mean process, and $P(0|0)$ is the covariance matrix for $Z(0) - Z(0|0) = Z(0)$. Now, because $Z(0)$ is the column vector with elements $[Z_0, \hat{Z}_0(1), \dots, \hat{Z}_0(m-1)]$, it is necessary for us to evaluate

$$\text{Cov}[\hat{Z}_0(i), \hat{Z}_0(j)] \quad \text{for } i, j = 0, 1, \dots, m-1$$

From the truncated linear process form, Equation (9.3.35) on page 200 with $C_t(\ell) = \hat{Z}_t(\ell)$, we may write, for $j > 0$

$$Z_j = \hat{Z}_0(j) + \sum_{k=-j}^{-1} \psi_{j+k} e_{-k} \quad (9.H.18)$$

Multiplying Equation (9.H.18) by Z_0 and taking expected values yields

$$\gamma_j = E(Z_0 Z_j) = E[\hat{Z}_0(0)(\hat{Z}_0(j))] \quad \text{for } j \geq 0 \quad (9.H.19)$$

Now multiply Equation (9.H.18) by itself with j replaced by i and take expected values. Recalling that the e 's are independent of past Z 's and assuming $0 < i \leq j$, we obtain

$$\gamma_{j-i} = Cov[\hat{Z}_0(i), \hat{Z}_0(j)] + \sigma_e^2 \sum_{k=0}^{i-1} \psi_k \psi_{k+j-i} \quad (9.H.20)$$

Combining Equations (9.H.19) and (9.H.20), we have as the required elements of $\mathbf{P}(0|0)$

$$Cov[\hat{Z}_0(i), \hat{Z}_0(j)] = \begin{cases} \gamma_i & 0 = i \leq j \leq m-1 \\ \gamma_{j-i} - \sigma_e^2 \sum_{k=0}^{i-1} \psi_k \psi_{k+j-i} & 1 \leq i \leq j \leq m-1 \end{cases} \quad (9.H.21)$$

where the ψ -weights are obtained from the recursion of Equation (4.4.7) on page 79, and γ_k , the autocovariance function for the $\{Z_t\}$ process, is obtained as in Appendix C on page 85.

The variance σ_e^2 can be removed from the problem by dividing σ_e^2 by σ_e^2 . The prediction error variance v_t is then replaced by $\sigma_e^2 v_t$ in the log-likelihood of Equation (9.H.17), and we set $\sigma_e^2 = 1$ in Equation (9.H.8). Dropping unneeded constants, we get the new log-likelihood

$$\ell = \sum_{t=1}^n \left\{ \log(\sigma_e^2 v_t) + \frac{[y_t - y(t|t-1)]^2}{v_t} \right\} \quad (9.H.22)$$

which can be minimized analytically with respect to σ_e^2 . We obtain

$$\sigma_e^2 = \sum_{t=1}^n \left\{ \frac{[y_t - y(t|t-1)]^2}{\sigma_e^2 v_t} \right\} \quad (9.H.23)$$

Substituting this back into Equation (9.H.22), we now find that

$$\ell = \sum_{t=1}^n \log v_t + n \log \sum_{t=1}^n \frac{[y_t - y(t|t-1)]^2}{v_t} \quad (9.H.24)$$

which must be minimized numerically with respect to $\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q$, and σ_e^2 . Having done so, we return to Equation (9.H.23) to estimate σ_e^2 . The function defined by Equation (9.H.24) is sometimes called the **concentrated log-likelihood function**.

CHAPTER 10

SEASONAL MODELS

In Chapter 3, we saw how seasonal deterministic trends might be modeled. However, in many areas in which time series are used, particularly business and economics, the assumption of any deterministic trend is quite suspect even though cyclical tendencies are very common in such series.

Here is an example: Levels of carbon dioxide (CO_2) are monitored at several sites around the world to investigate atmospheric changes. One of the sites is at Alert, Northwest Territories, Canada, near the Arctic Circle.

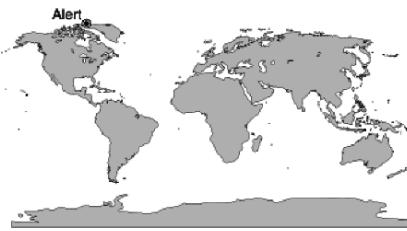
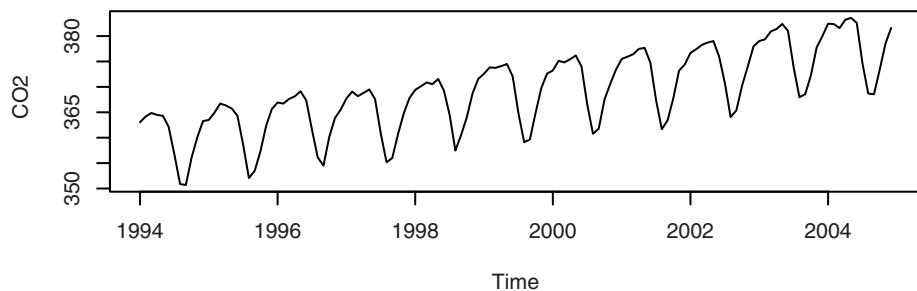


Exhibit 10.1 displays the monthly CO_2 levels from January 1994 through December 2004. There is a strong upward trend but also a seasonality that can be seen better in the more detailed Exhibit 10.2, where only the last few years are graphed using monthly plotting symbols.

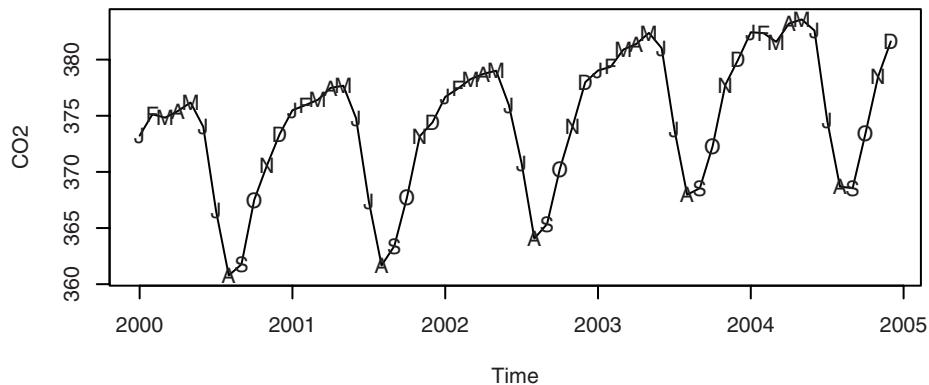
Exhibit 10.1 Monthly Carbon Dioxide Levels at Alert, NWT, Canada



```
> data(co2)
> win.graph(width=4.875, height=3, pointsize=8)
> plot(co2, ylab='CO2')
```

As we see in the displays, carbon dioxide levels are higher during the winter months and much lower in the summer. Deterministic seasonal models such as seasonal means plus linear time trend or sums of cosine curves at various frequencies plus linear time trend as we investigated in Chapter 3 could certainly be considered here. But we discover that such models do not explain the behavior of this time series. For this series and many others, it can be shown that the residuals from a seasonal means plus linear time trend model are highly autocorrelated at many lags.[†] In contrast, we will see that the **stochastic seasonal models** developed in this chapter do work well for this series.

Exhibit 10.2 Carbon Dioxide Levels with Monthly Symbols



```
> plot(window(co2,start=c(2000,1)),ylab='CO2')
> Month=c('J','F','M','A','M','J','J','A','S','O','N','D')
> points(window(co2,start=c(2000,1)),pch=Month)
```

10.1 Seasonal ARIMA Models

We begin by studying stationary models and then consider nonstationary generalizations in Section 10.3. We let s denote the known seasonal period; for monthly series $s = 12$ and for quarterly series $s = 4$.

Consider the time series generated according to

$$Y_t = e_t - \Theta e_{t-12}$$

Notice that

$$\begin{aligned} \text{Cov}(Y_t, Y_{t-1}) &= \text{Cov}(e_t - \Theta e_{t-12}, e_{t-1} - \Theta e_{t-13}) \\ &= 0 \end{aligned}$$

but that

[†] We ask you to verify this in Exercise 10.8.

$$\begin{aligned} \text{Cov}(Y_t, Y_{t-12}) &= \text{Cov}(e_t - \Theta e_{t-12}, e_{t-12} - \Theta e_{t-24}) \\ &= -\Theta \sigma_e^2 \end{aligned}$$

It is easy to see that such a series is stationary and has nonzero autocorrelations only at lag 12.

Generalizing these ideas, we define a **seasonal MA(Q) model of order Q with seasonal period s** by

$$Y_t = e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2s} - \cdots - \Theta_Q e_{t-Qs} \quad (10.1.1)$$

with **seasonal MA characteristic polynomial**

$$\Theta(x) = 1 - \Theta_1 x^s - \Theta_2 x^{2s} - \cdots - \Theta_Q x^{Qs} \quad (10.1.2)$$

It is evident that such a series is always stationary and that the autocorrelation function will be nonzero only at the seasonal lags of $s, 2s, 3s, \dots, Qs$. In particular,

$$\rho_{ks} = \frac{-\Theta_k + \Theta_1 \Theta_{k+1} + \Theta_2 \Theta_{k+2} + \cdots + \Theta_{Q-k} \Theta_Q}{1 + \Theta_1^2 + \Theta_2^2 + \cdots + \Theta_Q^2} \quad \text{for } k = 1, 2, \dots, Q \quad (10.1.3)$$

(Compare this with Equation (4.2.5) on page 65 for the nonseasonal MA process.) For the model to be invertible, the roots of $\Theta(x) = 0$ must all exceed 1 in absolute value.

It is useful to note that the seasonal MA(Q) model can also be viewed as a special case of a nonseasonal MA model of order $q = Qs$ but with all θ -values zero except at the seasonal lags $s, 2s, 3s, \dots, Qs$.

Seasonal autoregressive models can also be defined. Consider

$$Y_t = \Phi Y_{t-12} + e_t \quad (10.1.4)$$

where $|\Phi| < 1$ and e_t is independent of Y_{t-1}, Y_{t-2}, \dots . It can be shown that $|\Phi| < 1$ ensures stationarity. Thus it is easy to argue that $E(Y_t) = 0$; multiplying Equation (10.1.4) by Y_{t-k} , taking expectations, and dividing by γ_0 yields

$$\rho_k = \Phi \rho_{k-12} \quad \text{for } k \geq 1 \quad (10.1.5)$$

Clearly

$$\rho_{12} = \Phi \rho_0 = \Phi \quad \text{and} \quad \rho_{24} = \Phi \rho_{12} = \Phi^2$$

More generally,

$$\rho_{12k} = \Phi^k \quad \text{for } k = 1, 2, \dots \quad (10.1.6)$$

Furthermore, setting $k = 1$ and then $k = 11$ in Equation (10.1.5) and using $\rho_k = \rho_{-k}$ gives us

$$\rho_1 = \Phi \rho_{11} \quad \text{and} \quad \rho_{11} = \Phi \rho_1$$

which implies that $\rho_1 = \rho_{11} = 0$. Similarly, one can show that $\rho_k = 0$ except at the seasonal lags 12, 24, 36, \dots . At those lags, the autocorrelation function decays exponentially like an AR(1) model.

With this example in mind, we define a **seasonal AR(P) model of order P and seasonal period s** by

$$Y_t = \Phi_1 Y_{t-s} + \Phi_2 Y_{t-2s} + \cdots + \Phi_P Y_{t-Ps} + e_t \quad (10.1.7)$$

with **seasonal characteristic polynomial**

$$\Phi(x) = 1 - \Phi_1 x^s - \Phi_2 x^{2s} - \cdots - \Phi_P x^{Ps} \quad (10.1.8)$$

As always, we require e_t to be independent of Y_{t-1}, Y_{t-2}, \dots , and, for stationarity, that the roots of $\Phi(x) = 0$ be greater than 1 in absolute value. Again, Equation (10.1.7) can be seen as a special AR(p) model of order $p = Ps$ with nonzero ϕ -coefficients only at the seasonal lags $s, 2s, 3s, \dots, Ps$.

It can be shown that the autocorrelation function is nonzero only at lags $s, 2s, 3s, \dots$, where it behaves like a combination of decaying exponentials and damped sine functions. In particular, Equations (10.1.4), (10.1.5), and (10.1.6) easily generalize to the general seasonal AR(1) model to give

$$\rho_{ks} = \Phi^k \text{ for } k = 1, 2, \dots \quad (10.1.9)$$

with zero correlation at other lags.

10.2 Multiplicative Seasonal ARMA Models

Rarely shall we need models that incorporate autocorrelation *only* at the seasonal lags. By combining the ideas of seasonal and nonseasonal ARMA models, we can develop parsimonious models that contain autocorrelation for the seasonal lags but also for low lags of neighboring series values.

Consider a model whose MA characteristic polynomial is given by

$$(1 - \theta x)(1 - \Theta x^{12})$$

Multiplying out, we have $1 - \theta x - \Theta x^{12} + \theta\Theta x^{13}$. Thus the corresponding time series satisfies

$$Y_t = e_t - \theta e_{t-1} - \Theta e_{t-12} + \theta\Theta e_{t-13} \quad (10.2.1)$$

For this model, we can check that the autocorrelation function is nonzero only at lags 1, 11, 12, and 13. We find

$$\gamma_0 = (1 + \theta^2)(1 + \Theta^2)\sigma_e^2 \quad (10.2.2)$$

$$\rho_1 = -\frac{\theta}{1 + \theta^2} \quad (10.2.3)$$

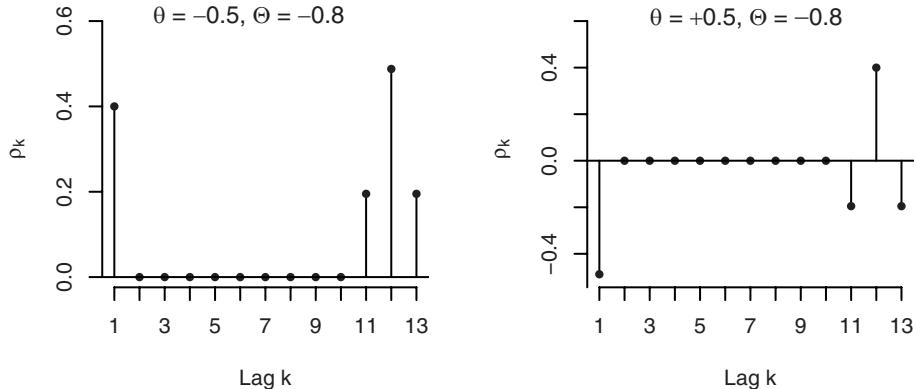
$$\rho_{11} = \rho_{13} = \frac{\theta\Theta}{(1 + \theta^2)(1 + \Theta^2)} \quad (10.2.4)$$

and

$$\rho_{12} = -\frac{\Theta}{1 + \Theta^2} \quad (10.2.5)$$

Exhibit 10.3 displays the autocorrelation functions for the model of Equation (10.2.1) with $\theta = \pm 0.5$ and $\Theta = -0.8$ as given by Equations (10.2.2)–(10.2.5).

Exhibit 10.3 Autocorrelations from Equations (10.2.2)–(10.2.5)



Of course, we could also introduce both short-term and seasonal autocorrelations by defining an MA model of order 12 with only θ_1 and θ_{12} nonzero. We shall see in the next section that the “multiplicative” model arises quite naturally for nonstationary models that entail differencing.

In general, then, we define a **multiplicative seasonal ARMA(p,q) $\times(P,Q)_s$ model with seasonal period s** as a model with AR characteristic polynomial $\phi(x)\Phi(x)$ and MA characteristic polynomial $\theta(x)\Theta(x)$, where

$$\left. \begin{aligned} \phi(x) &= 1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p \\ \Phi(x) &= 1 - \Phi_1 x^s - \Phi_2 x^{2s} - \cdots - \Phi_P x^{Ps} \end{aligned} \right\} \quad (10.2.6)$$

and

$$\left. \begin{aligned} \theta(x) &= 1 - \theta_1 x - \theta_2 x^2 - \cdots - \theta_q x^q \\ \Theta(x) &= 1 - \Theta_1 x^s - \Theta_2 x^{2s} - \cdots - \Theta_Q x^{Qs} \end{aligned} \right\} \quad (10.2.7)$$

The model may also contain a constant term θ_0 . Note once more that we have just a special ARMA model with AR order $p + Ps$ and MA order $q + Qs$, but the coefficients are not completely general, being determined by only $p + P + q + Q$ coefficients. If $s = 12$, $p + P + q + Q$ will be considerably smaller than $p + Ps + q + Qs$ and will allow a much more parsimonious model.

As another example, suppose $P = q = 1$ and $p = Q = 0$ with $s = 12$. The model is then

$$Y_t = \Phi Y_{t-12} + e_t - \theta e_{t-1} \quad (10.2.8)$$

Using our standard techniques, we find that

$$\gamma_1 = \Phi\gamma_{11} - \theta\sigma_e^2 \quad (10.2.9)$$

and

$$\gamma_k = \Phi\gamma_{k-12} \text{ for } k \geq 2 \quad (10.2.10)$$

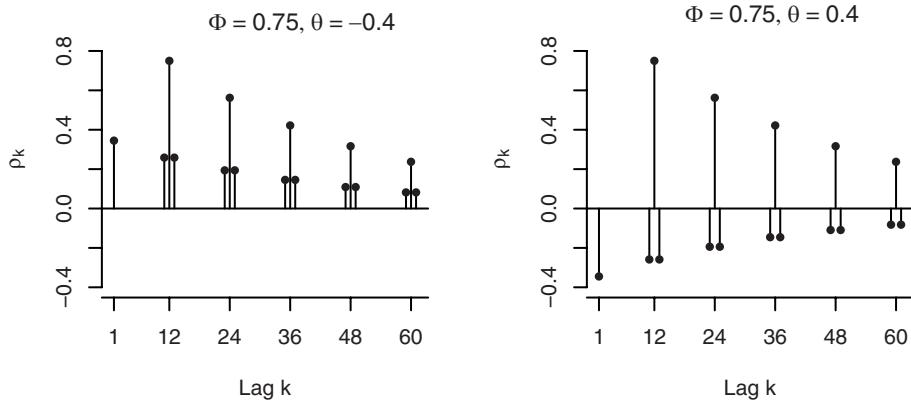
After considering the equations implied by various choices for k , we arrive at

$$\left. \begin{aligned} \gamma_0 &= \left[\frac{1 + \theta^2}{1 - \Phi^2} \right] \sigma_e^2 \\ \rho_{12k} &= \Phi^k \text{ for } k \geq 1 \\ \rho_{12k-1} &= \rho_{12k+1} = \left(-\frac{\theta}{1 + \theta^2} \Phi^k \right) \text{ for } k = 0, 1, 2, \dots \end{aligned} \right\} \quad (10.2.11)$$

with autocorrelations for all other lags equal to zero.

Exhibit 10.4 displays the autocorrelation functions for two of these seasonal ARIMA processes with period 12: one with $\Phi = 0.75$ and $\theta = -0.4$, the other with $\Phi = 0.75$ and $\theta = 0.4$. The shape of these autocorrelations is somewhat typical of the sample autocorrelation functions for numerous seasonal time series. The even simpler autocorrelation function given by Equations (10.2.3), (10.2.4), and (10.2.5) and displayed in Exhibit 10.3 also seems to occur frequently in practice (perhaps after differencing).

Exhibit 10.4 Autocorrelation Functions from Equation (10.2.11)



10.3 Nonstationary Seasonal ARIMA Models

An important tool in modeling nonstationary seasonal processes is the seasonal difference. The **seasonal difference** of period s for the series $\{Y_t\}$ is denoted $\nabla_s Y_t$ and is defined as

$$\nabla_s Y_t = Y_t - Y_{t-s} \quad (10.3.1)$$

For example, for monthly series we consider the changes from January to January, February to February, and so forth for successive years. Note that for a series of length n , the seasonal difference series will be of length $n - s$; that is, s data values are lost due to seasonal differencing.

As an example where seasonal differencing is appropriate, consider a process generated according to

$$Y_t = S_t + e_t \quad (10.3.2)$$

with

$$S_t = S_{t-s} + \varepsilon_t \quad (10.3.3)$$

where $\{e_t\}$ and $\{\varepsilon_t\}$ are independent white noise series. Here $\{S_t\}$ is a “seasonal random walk,” and if $\sigma_\varepsilon \ll \sigma_e$, $\{S_t\}$ would model a slowly changing seasonal component.

Due to the nonstationarity of $\{S_t\}$, clearly $\{Y_t\}$ is nonstationary. However, if we seasonally difference $\{Y_t\}$, as given in Equation (10.3.1), we find

$$\begin{aligned} \nabla_s Y_t &= S_t - S_{t-s} + e_t - e_{t-s} \\ &= \varepsilon_t + e_t - e_{t-s} \end{aligned} \quad (10.3.4)$$

An easy calculation shows that $\nabla_s Y_t$ is stationary and has the autocorrelation function of an $\text{MA}(1)_s$ model.

The model described by Equations (10.2.2) and (10.2.3) could also be generalized to account for a nonseasonal, slowly changing stochastic trend. Consider

$$Y_t = M_t + S_t + e_t \quad (10.3.5)$$

with

$$S_t = S_{t-s} + \varepsilon_t \quad (10.3.6)$$

and

$$M_t = M_{t-1} + \xi_t \quad (10.3.7)$$

where $\{e_t\}$, $\{\varepsilon_t\}$, and $\{\xi_t\}$ are mutually independent white noise series. Here we take both a seasonal difference and an ordinary nonseasonal difference to obtain[†]

[†] It should be noted that $\nabla_s Y_t$ will in fact be stationary and $\nabla \nabla_s Y_t$ will be noninvertible. We use Equations (10.2.5), (10.2.6), and (10.2.7) merely to help motivate multiplicative seasonal ARIMA models.

$$\begin{aligned}\nabla \nabla_s Y_t &= \nabla(M_t - M_{t-s} + \varepsilon_t + e_t - e_{t-s}) \\ &= (\xi_t + \varepsilon_t + e_t) - (\varepsilon_{t-1} + e_{t-1}) - (\xi_{t-s} + e_{t-s}) + e_{t-s-1}\end{aligned}\tag{10.3.8}$$

The process defined here is stationary and has nonzero autocorrelation only at lags 1, $s-1$, s , and $s+1$, which agrees with the autocorrelation structure of the multiplicative seasonal model ARMA(0,1) \times (0,1) with seasonal period s .

These examples lead to the definition of nonstationary seasonal models. A process $\{Y_t\}$ is said to be a **multiplicative seasonal ARIMA model** with nonseasonal (regular) orders p , d , and q , seasonal orders P , D , and Q , and seasonal period s if the differenced series

$$W_t = \nabla^d \nabla_s^D Y_t \tag{10.3.9}$$

satisfies an ARMA(p, q) \times (P, Q) $_s$ model with seasonal period s .[†] We say that $\{Y_t\}$ is an ARIMA(p, d, q) \times (P, D, Q) $_s$ model with seasonal period s .

Clearly, such models represent a broad, flexible class from which to select an appropriate model for a particular time series. It has been found empirically that many series can be adequately fit by these models, usually with a small number of parameters, say three or four.

10.4 Model Specification, Fitting, and Checking

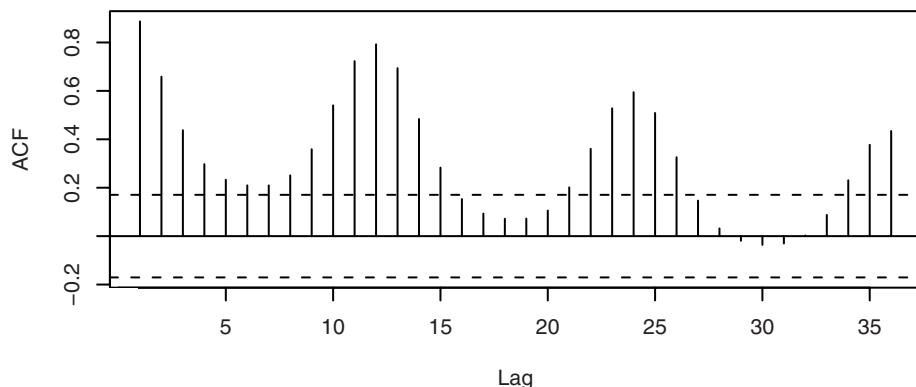
Model specification, fitting, and diagnostic checking for seasonal models follow the same general techniques developed in Chapters 6, 7, and 8. Here we shall simply highlight the application of these ideas specifically to seasonal models and pay special attention to the seasonal lags.

Model Specification

As always, a careful inspection of the time series plot is the first step. Exhibit 10.1 on page 227 displays monthly carbon dioxide levels in northern Canada. The upward trend alone would lead us to specify a nonstationary model. Exhibit 10.5 shows the sample autocorrelation function for that series. The seasonal autocorrelation relationships are shown quite prominently in this display. Notice the strong correlation at lags 12, 24, 36, and so on. In addition, there is substantial other correlation that needs to be modeled.

[†] Using the backshift operator notation of Appendix D, page 106, we may write the general ARIMA(p, d, q) \times (P, D, Q) $_s$ model as $\phi(B)\Phi(B)\nabla^d \nabla_s^D Y_t = \theta(B)\Theta(B)e_t$

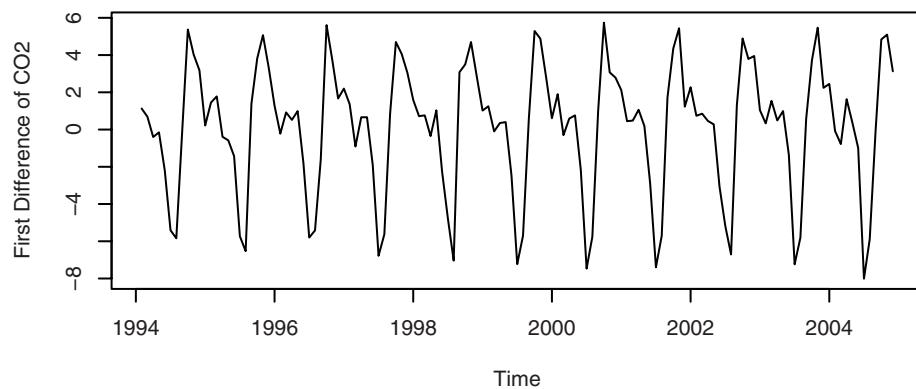
Exhibit 10.5 Sample ACF of CO₂ Levels



```
> acf(as.vector(co2), lag.max=36)
```

Exhibit 10.6 shows the time series plot of the CO₂ levels after we take a first difference.

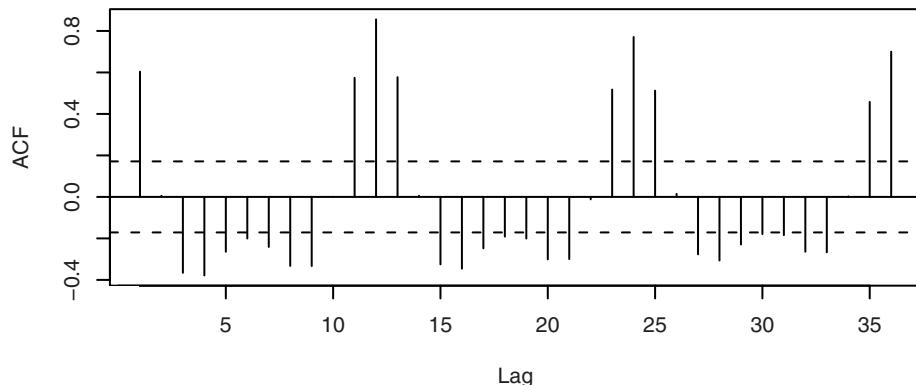
Exhibit 10.6 Time Series Plot of the First Differences of CO₂ Levels



```
> plot(diff(co2), ylab='First Difference of CO2', xlab='Time')
```

The general upward trend has now disappeared but the strong seasonality is still present, as evidenced by the behavior shown in Exhibit 10.7. Perhaps seasonal differencing will bring us to a series that may be modeled parsimoniously.

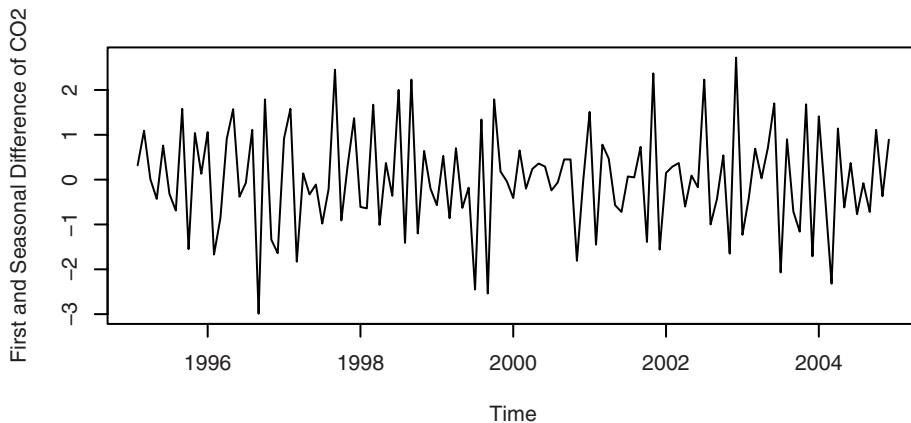
Exhibit 10.7 Sample ACF of First Differences of CO₂ Levels



```
> acf(as.vector(diff(co2)),lag.max=36)
```

Exhibit 10.8 displays the time series plot of the CO₂ levels after taking both a first difference and a seasonal difference. It appears that most, if not all, of the seasonality is gone now.

Exhibit 10.8 Time Series Plot of First and Seasonal Differences of CO₂



```
> plot(diff(diff(co2),lag=12),xlab='Time',
      ylab='First and Seasonal Difference of CO2')
```

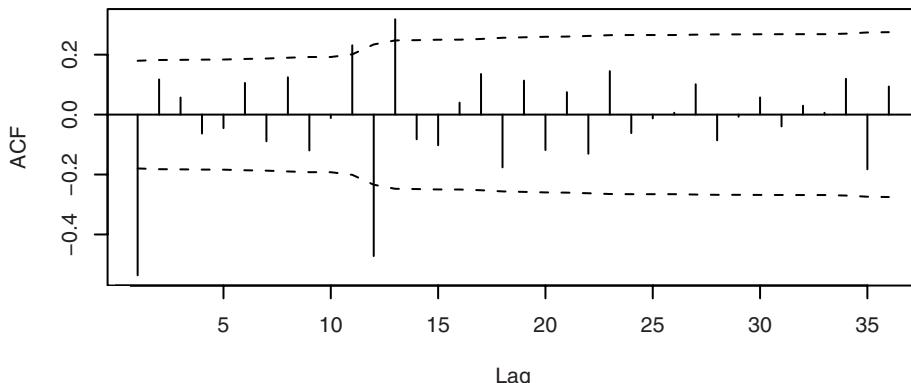
Exhibit 10.9 confirms that very little autocorrelation remains in the series after these two differences have been taken. This plot also suggests that a simple model which incorporates the lag 1 and lag 12 autocorrelations might be adequate.

We will consider specifying the multiplicative, seasonal ARIMA(0,1,1) \times (0,1,1)₁₂ model

$$\nabla_{12} \nabla Y_t = e_t - \theta e_{t-1} - \Theta e_{t-12} + \theta \Theta e_{t-13} \quad (10.4.10)$$

which incorporates many of these requirements. As usual, all models are tentative and subject to revision at the diagnostics stage of model building.

Exhibit 10.9 Sample ACF of First and Seasonal Differences of CO₂



```
> acf(as.vector(diff(diff(co2),lag=12)),lag.max=36,ci.type='ma')
```

Model Fitting

Having specified a tentative seasonal model for a particular time series, we proceed to estimate the parameters of that model as efficiently as possible. As we have remarked earlier, multiplicative seasonal ARIMA models are just special cases of our general ARIMA models. As such, all of our work on parameter estimation in Chapter 7 carries over to the seasonal case.

Exhibit 10.10 gives the maximum likelihood estimates and their standard errors for the ARIMA(0,1,1) \times (0,1,1)₁₂ model for CO₂ levels.

Exhibit 10.10 Parameter Estimates for the CO₂ Model

Coefficient	θ	Θ
Estimate	0.5792	0.8206
Standard error	0.0791	0.1137
$\hat{\sigma}_e^2 = 0.5446$: log-likelihood = -139.54, AIC = 283.08		

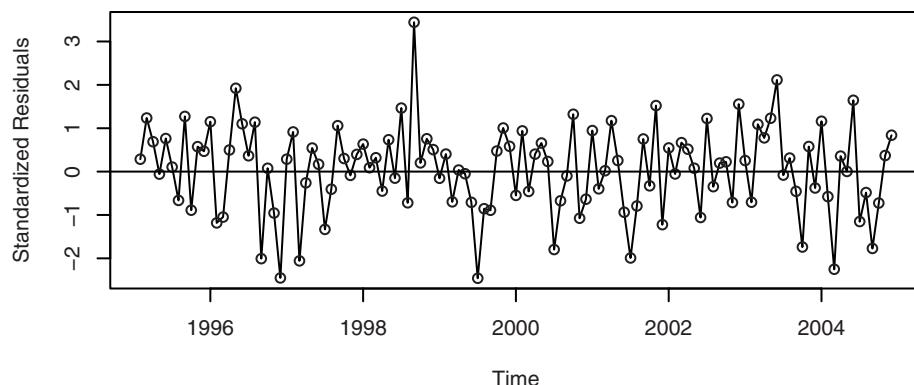
```
> m1.co2=arima(co2,order=c(0,1,1),seasonal=list(order=c(0,1,1),
  period=12))
> m1.co2
```

The coefficient estimates are all highly significant, and we proceed to check further on this model.

Diagnostic Checking

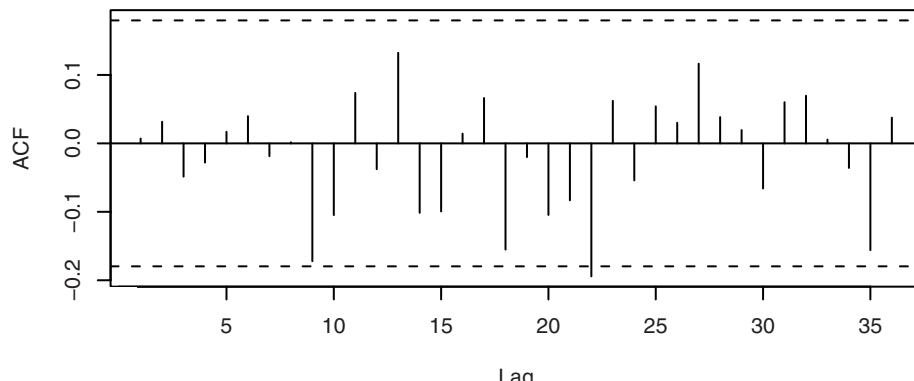
To check the estimated the ARIMA(0,1,1) \times (0,1,1)₁₂ model, we first look at the time series plot of the residuals. Exhibit 10.11 gives this plot for standardized residuals. Other than some strange behavior in the middle of the series, this plot does not suggest any major irregularities with the model, although we may need to investigate the model further for outliers, as the standardized residual at September 1998 looks suspicious. We investigate this further in Chapter 11.

Exhibit 10.11 Residuals from the ARIMA(0,1,1) \times (0,1,1)₁₂ Model



```
> plot(window(rstandard(m1.co2), start=c(1995,2)),
      ylab='Standardized Residuals', type='o')
> abline(h=0)
```

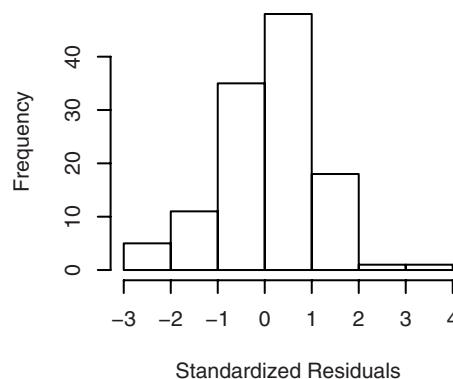
To look further, we graph the sample ACF of the residuals in Exhibit 10.12. The only “statistically significant” correlation is at lag 22, and this correlation has a value of only -0.17 , a very small correlation. Furthermore, we can think of no reasonable interpretation for dependence at lag 22. Finally, we should not be surprised that one autocorrelation out of the 36 displayed is statistically significant. This could easily happen by chance alone. Except for marginal significance at lag 22, the model seems to have captured the essence of the dependence in the series.

Exhibit 10.12 ACF of Residuals from the ARIMA(0,1,1) \times (0,1,1)₁₂ Model

```
> acf(as.vector(window(rstandard(m1.co2), start=c(1995, 2))),  
lag.max=36)
```

The Ljung-Box test for this model gives a chi-squared value of 25.59 with 22 degrees of freedom, leading to a p -value of 0.27—a further indication that the model has captured the dependence in the time series.

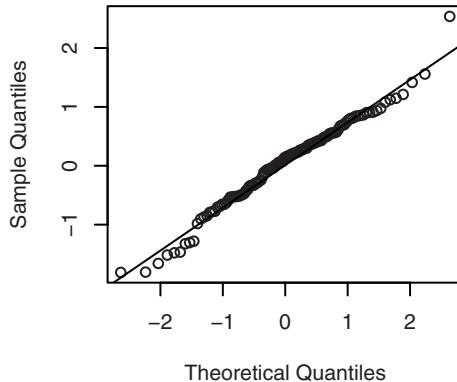
Next we investigate the question of normality of the error terms via the residuals. Exhibit 10.13 displays the histogram of the residuals. The shape is somewhat “bell-shaped” but certainly not ideal. Perhaps a quantile-quantile plot will tell us more.

Exhibit 10.13 Residuals from the ARIMA(0,1,1) \times (0,1,1)₁₂ Model

```
> win.graph(width=3, height=3, pointsize=8)  
> hist(window(rstandard(m1.co2), start=c(1995, 2)),  
xlab='Standardized Residuals')
```

Exhibit 10.14 displays the QQ-normal plot for the residuals.

Exhibit 10.14 Residuals: ARIMA(0,1,1)×(0,1,1)₁₂ Model



```
> win.graph(width=2.5, height=2.5, pointsize=8)
> qqnorm(window(rstandard(m1.co2), start=c(1995, 2)))
> qqline(window(rstandard(m1.co2), start=c(1995, 2)))
```

Here we again see the one outlier in the upper tail, but the Shapiro-Wilk test of normality has a test statistic of $W = 0.982$, leading to a p -value of 0.11, and normality is not rejected at any of the usual significance levels.

As one further check on the model, we consider overfitting with an ARIMA(0,1,2) × (0,1,1)₁₂ model with the results shown in Exhibit 10.15.

Exhibit 10.15 ARIMA(0,1,2)×(0,1,1)₁₂ Overfitted Model

Coefficient	θ_1	θ_2	Θ
Estimate	0.5714	0.0165	0.8274
Standard error	0.0897	0.0948	0.1224

$$\hat{\sigma}_e^2 = 0.5427; \text{log-likelihood} = -139.52, \text{AIC} = 285.05$$

```
> m2.co2=arima(co2, order=c(0,1,2), seasonal=list(order=c(0,1,1),
  period=12))
> m2.co2
```

When we compare these results with those reported in Exhibit 10.10 on page 237, we see that the estimates of θ_1 and Θ have changed very little—especially when the size of the standard errors is taken into consideration. In addition, the estimate of the new parameter, θ_2 , is not statistically different from zero. Note also that the estimate $\hat{\sigma}_e^2$ and the log-likelihood have not changed much while the AIC has actually increased.

The ARIMA(0,1,1) × (0,1,1)₁₂ model was popularized in the first edition of the seminal book of Box and Jenkins (1976) when it was found to characterize the logarithms of

a monthly airline passenger time series. This model has come to be known as the **airline model**. We ask you to analyze the original airline data in the exercises.

10.5 Forecasting Seasonal Models

Computing forecasts with seasonal ARIMA models is, as expected, most easily carried out recursively using the difference equation form for the model, as in Equations (9.3.28), (9.3.29) on page 199 and (9.3.40) on page 201. For example, consider the model ARIMA(0,1,1) \times (1,0,1)₁₂.

$$Y_t - Y_{t-1} = \Phi(Y_{t-12} - Y_{t-13}) + e_t - \theta e_{t-1} - \Theta e_{t-12} + \theta \Theta e_{t-13} \quad (10.5.1)$$

which we rewrite as

$$Y_t = Y_{t-1} + \Phi Y_{t-12} - \Phi Y_{t-13} + e_t - \theta e_{t-1} - \Theta e_{t-12} + \theta \Theta e_{t-13} \quad (10.5.2)$$

The one-step-ahead forecast from origin t is then

$$\hat{Y}_t(1) = Y_t + \Phi Y_{t-11} - \Phi Y_{t-12} - \theta e_t - \Theta e_{t-11} + \theta \Theta e_{t-12} \quad (10.5.3)$$

and the next one is

$$\hat{Y}_t(2) = \hat{Y}_t(1) + \Phi Y_{t-10} - \Phi Y_{t-11} - \Theta e_{t-10} + \theta \Theta e_{t-11} \quad (10.5.4)$$

and so forth. The noise terms $e_{t-13}, e_{t-12}, e_{t-11}, \dots, e_t$ (as residuals) will enter into the forecasts for lead times $\ell = 1, 2, \dots, 13$, but for $\ell > 13$ the autoregressive part of the model takes over and we have

$$\hat{Y}_t(\ell) = \hat{Y}_t(\ell-1) + \Phi \hat{Y}_t(\ell-12) - \Phi \hat{Y}_t(\ell-13) \text{ for } \ell > 13 \quad (10.5.5)$$

To understand the general nature of the forecasts, we consider several special cases.

Seasonal AR(1)₁₂

The seasonal AR(1)₁₂ model is

$$Y_t = \Phi Y_{t-12} + e_t \quad (10.5.6)$$

Clearly, we have

$$\hat{Y}_t(\ell) = \Phi \hat{Y}_t(\ell-12) \quad (10.5.7)$$

However, iterating back on ℓ , we can also write

$$\hat{Y}_t(\ell) = \Phi^{k+1} Y_{t+r-11} \quad (10.5.8)$$

where k and r are defined by $\ell = 12k + r + 1$ with $0 \leq r < 12$ and $k = 0, 1, 2, \dots$. In other words, k is the integer part of $(\ell-1)/12$ and $r/12$ is the fractional part of $(\ell-1)/12$. If our last observation is in December, then the next January value is forecast as Φ times the last observed January value, February is forecast as Φ times the last observed February

value, and so on. Two Januaries ahead is forecast as Φ^2 times the last observed January. Looking just at January values, the forecasts into the future will decay exponentially at a rate determined by the magnitude of Φ . All of the forecasts for each month will behave similarly but with different initial forecasts depending on the particular month under consideration.

Using Equation (9.3.38) on page 201 and the fact that the ψ -weights are nonzero only for multiple of 12, namely,

$$\psi_j = \begin{cases} \Phi^{j/12} & \text{for } j = 0, 12, 24, \dots \\ 0 & \text{otherwise} \end{cases} \quad (10.5.9)$$

we have that the forecast error variance can be written as

$$Var(e_t(\ell)) = \left[\frac{1 - \Phi^{2k+2}}{1 - \Phi^2} \right] \sigma_e^2 \quad (10.5.10)$$

where, as before, k is the integer part of $(\ell - 1)/12$.

Seasonal MA(1)₁₂

For the seasonal MA(1)₁₂ model, we have

$$Y_t = e_t - \Theta e_{t-12} + \theta_0 \quad (10.5.11)$$

In this case, we see that

$$\left. \begin{array}{l} \hat{Y}_t(1) = -\Theta e_{t-11} + \theta_0 \\ \hat{Y}_t(2) = -\Theta e_{t-10} + \theta_0 \\ \vdots \\ \hat{Y}_t(12) = -\Theta e_t + \theta_0 \end{array} \right\} \quad (10.5.12)$$

and

$$\hat{Y}_t(\ell) = \theta_0 \quad \text{for } \ell > 12 \quad (10.5.13)$$

Here we obtain different forecasts for the months of the first year, but from then on all forecasts are given by the process mean.

For this model, $\psi_0 = 1$, $\psi_{12} = -\Theta$, and $\psi_j = 0$ otherwise. Thus, from Equation (9.3.38) on page 201,

$$Var(e_t(\ell)) = \begin{cases} \sigma_e^2 & 1 \leq \ell \leq 12 \\ (1 + \Theta^2)\sigma_e^2 & 12 < \ell \end{cases} \quad (10.5.14)$$

ARIMA(0,0,0)×(0,1,1)₁₂

The ARIMA(0,0,0)×(0,1,1)₁₂ model is

$$Y_t - Y_{t-12} = e_t - \Theta e_{t-12} \quad (10.5.15)$$

or

$$Y_{t+\ell} = Y_{t+\ell-12} + e_{t+\ell} - \Theta e_{t+\ell-12}$$

so that

$$\left. \begin{aligned} \hat{Y}_t(1) &= Y_{t-11} - \Theta e_{t-11} \\ \hat{Y}_t(2) &= Y_{t-10} - \Theta e_{t-10} \\ &\vdots \\ \hat{Y}_t(12) &= Y_t - \Theta e_t \end{aligned} \right\} \quad (10.5.16)$$

and then

$$\hat{Y}_t(\ell) = \hat{Y}_t(\ell-12) \text{ for } \ell > 12 \quad (10.5.17)$$

It follows that all Januarys will forecast identically, all Februarys identically, and so forth.

If we invert this model, we find that

$$Y_t = (1 - \Theta)(Y_{t-12} + \Theta Y_{t-24} + \Theta^2 Y_{t-36} + \dots) + e_t$$

Consequently, we can write

$$\left. \begin{aligned} \hat{Y}_t(1) &= (1 - \Theta) \sum_{j=0}^{\infty} \Theta^j Y_{t-11-12j} \\ \hat{Y}_t(2) &= (1 - \Theta) \sum_{j=0}^{\infty} \Theta^j Y_{t-10-12j} \\ &\vdots \\ \hat{Y}_t(12) &= (1 - \Theta) \sum_{j=0}^{\infty} \Theta^j Y_{t-12j} \end{aligned} \right\} \quad (10.5.18)$$

From this representation, we see that the forecast for each January is an exponentially weighted moving average of all observed Januarys, and similarly for each of the other months.

In this case, we have $\psi_j = 1 - \Theta$ for $j = 12, 24, \dots$, and zero otherwise. The forecast error variance is then

$$\text{Var}(e_t(\ell)) = [1 + k(1 - \Theta)^2] \sigma_e^2 \quad (10.5.19)$$

where k is the integer part of $(\ell - 1)/12$.

ARIMA(0,1,1)×(0,1,1)₁₂

For the ARIMA(0,1,1)×(0,1,1)₁₂ model

$$Y_t = Y_{t-1} + Y_{t-12} - Y_{t-13} + e_t - \theta e_{t-1} - \Theta e_{t-12} + \theta \Theta e_{t-13} \quad (10.5.20)$$

the forecasts satisfy

$$\left. \begin{aligned} \hat{Y}_t(1) &= Y_t + Y_{t-11} - Y_{t-12} - \theta e_t - \Theta e_{t-11} + \theta \Theta e_{t-12} \\ \hat{Y}_t(2) &= \hat{Y}_t(1) + Y_{t-10} - Y_{t-11} - \Theta e_{t-10} + \theta \Theta e_{t-11} \\ &\vdots \\ \hat{Y}_t(12) &= \hat{Y}_t(11) + Y_t - Y_{t-1} - \Theta e_t + \theta \Theta e_{t-1} \\ \hat{Y}_t(13) &= \hat{Y}_t(12) + \hat{Y}_t(1) - Y_t + \theta \Theta e_t \end{aligned} \right\} \quad (10.5.21)$$

and

$$\hat{Y}_t(\ell) = \hat{Y}_t(\ell-1) + \hat{Y}_t(\ell-12) - \hat{Y}_t(\ell-13) \text{ for } \ell > 13 \quad (10.5.22)$$

To understand the general pattern of these forecasts, we can use the representation

$$\hat{Y}_t(\ell) = A_1 + A_2 \ell + \sum_{j=0}^6 \left[B_{1j} \cos\left(\frac{2\pi j \ell}{12}\right) + B_{2j} \sin\left(\frac{2\pi j \ell}{12}\right) \right] \quad (10.5.23)$$

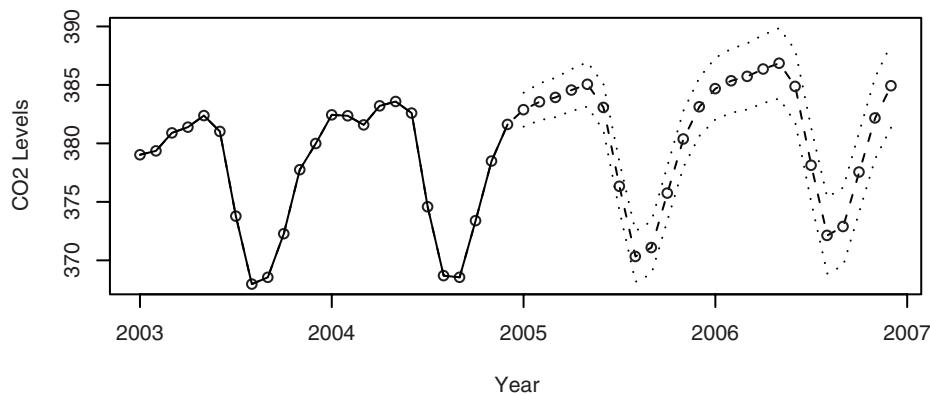
where the A 's and B 's are dependent on Y_t, Y_{t-1}, \dots , or, alternatively, determined from the initial forecasts $\hat{Y}_t(1), \hat{Y}_t(2), \dots, \hat{Y}_t(13)$. This result follows from the general theory of difference equations and involves the roots of $(1-x)(1-x^{12})=0$.

Notice that Equation (10.5.23) reveals that the forecasts are composed of a linear trend in the lead time plus a sum of periodic components. However, the coefficients A_i and B_{ij} are more dependent on recent data than on past data and will adapt to changes in the process as our forecast origin changes and the forecasts are updated. This is in stark contrast to forecasting with deterministic time trend plus seasonal components, where the coefficients depend rather equally on both recent and past data and remain the same for all future forecasts.

Prediction Limits

Prediction limits are obtained precisely as in the nonseasonal case. We illustrate this with the carbon dioxide time series. Exhibit 10.16 shows the forecasts and 95% forecast limits for a lead time of two years for the ARIMA(0,1,1) \times (0,1,1)₁₂ model that we fit. The last two years of observed data are also shown. The forecasts mimic the stochastic periodicity in the data quite well, and the forecast limits give a good feeling for the precision of the forecasts.

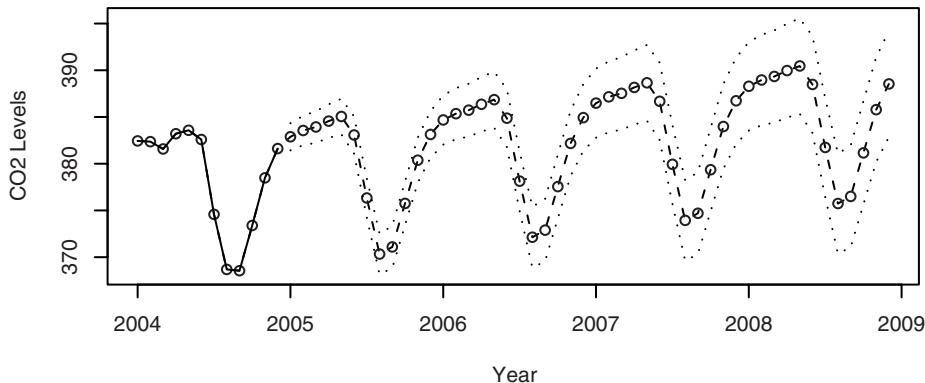
Exhibit 10.16 Forecasts and Forecast Limits for the CO₂ Model



```
> win.graph(width=4.875,height=3,pointsize=8)
> plot(m1.co2,n1=c(2003,1),n.ahead=24,xlab='Year',type='o',
      ylab='CO2 Levels')
```

Exhibit 10.17 displays the last year of observed data and forecasts out four years. At this lead time, it is easy to see that the forecast limits are getting wider, as there is more uncertainty in the forecasts.

Exhibit 10.17 Long-Term Forecasts for the CO₂ Model



```
> plot(m1.co2,n1=c(2004,1),n.ahead=48,xlab='Year',type='b',
      ylab='CO2 Levels')
```

10.6 Summary

Multiplicative seasonal ARIMA models provide an economical way to model time series whose seasonal tendencies are not as regular as we would have with a deterministic seasonal trend model which we covered in Chapter 3. Fortunately, these models are simply special ARIMA models so that no new theory is needed to investigate their properties. We illustrated the special nature of these models with a thorough modeling of an actual time series.

EXERCISES

- 10.1** Based on quarterly data, a seasonal model of the form

$$Y_t = Y_{t-4} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2}$$

has been fit to a certain time series.

- (a) Find the first four ψ -weights for this model.
- (b) Suppose that $\theta_1 = 0.5$, $\theta_2 = -0.25$, and $\sigma_e = 1$. Find forecasts for the next four quarters if data for the last four quarters are

Quarter	I	II	III	IV
Series	25	20	25	40
Residual	2	1	2	3

- (c) Find 95% prediction intervals for the forecasts in part (b).

- 10.2** An AR model has AR characteristic polynomial

$$(1 - 1.6x + 0.7x^2)(1 - 0.8x^{12})$$

- (a) Is the model stationary?
- (b) Identify the model as a certain seasonal ARIMA model.

- 10.3** Suppose that $\{Y_t\}$ satisfies

$$Y_t = a + bt + S_t + X_t$$

where S_t is deterministic and periodic with period s and $\{X_t\}$ is a seasonal ARIMA($p, 0, q$) $\times(P, 1, Q)_s$ series. What is the model for $W_t = Y_t - Y_{t-s}$?

- 10.4** For the seasonal model $Y_t = \Phi Y_{t-4} + e_e - \theta e_{t-1}$ with $|\Phi| < 1$, find γ_0 and ρ_k .
- 10.5** Identify the following as certain multiplicative seasonal ARIMA models:
 - (a) $Y_t = 0.5Y_{t-1} + Y_{t-4} - 0.5Y_{t-5} + e_t - 0.3e_{t-1}$.
 - (b) $Y_t = Y_{t-1} + Y_{t-12} + Y_{t-13} + e_t - 0.5e_{t-1} - e_{t-12} + 0.25e_{t-13}$.
- 10.6** Verify Equations (10.2.11) on page 232.

- 10.7** Suppose that the process $\{Y_t\}$ develops according to $Y_t = Y_{t-4} + e_t$ with $Y_t = e_t$ for $t = 1, 2, 3$, and 4 .
- Find the variance function for $\{Y_t\}$.
 - Find the autocorrelation function for $\{Y_t\}$.
 - Identify the model for $\{Y_t\}$ as a certain seasonal ARIMA model.
- 10.8** Consider the Alert, Canada, monthly carbon dioxide time series shown in Exhibit 10.1 on page 227. The data are in the file named co2.
- Fit a deterministic seasonal means plus linear time trend model to these data. Are any of the regression coefficients “statistically significant”?
 - What is the multiple R-squared for this model?
 - Now calculate the sample autocorrelation of the residuals from this model. Interpret the results.
- 10.9** The monthly airline passenger time series, first investigated in Box and Jenkins (1976), is considered a classic time series. The data are in the file named airline.
- Display the time series plots of both the original series and the logarithms of the series. Argue that taking logs is an appropriate transformation.
 - Display and interpret the time series plots of the first difference of the logged series.
 - Display and interpret the time series plot of the seasonal difference of the first difference of the logged series.
 - Calculate and interpret the sample ACF of the seasonal difference of the first difference of the logged series.
 - Fit the “airline model” ($\text{ARIMA}(0,1,1) \times (0,1,1)_{12}$) to the logged series.
 - Investigate diagnostics for this model, including autocorrelation and normality of the residuals.
 - Produce forecasts for this series with a lead time of two years. Be sure to include forecast limits.
- 10.10** Exhibit 5.8 on page 99 displayed the monthly electricity generated in the United States. We argued there that taking logarithms was appropriate for modeling. Exhibit 5.10 on page 100 showed the time series plot of the first differences for this series. The filename is electricity.
- Calculate the sample ACF of the first difference of the logged series. Is the seasonality visible in this display?
 - Plot the time series of seasonal difference and first difference of the logged series. Does a stationary model seem appropriate now?
 - Display the sample ACF of the series after a seasonal difference and a first difference have been taken of the logged series. What model(s) might you consider for the electricity series?

- 10.11** The quarterly earnings per share for 1960–1980 of the U.S. company Johnson & Johnson, are saved in the file named JJ.
- (a) Plot the time series and also the logarithm of the series. Argue that we should transform by logs to model this series.
 - (b) The series is clearly not stationary. Take first differences and plot that series. Does stationarity now seem reasonable?
 - (c) Calculate and graph the sample ACF of the first differences. Interpret the results.
 - (d) Display the plot of seasonal differences and the first differences. Interpret the plot. Recall that for quarterly data, a season is of length 4.
 - (e) Graph and interpret the sample ACF of seasonal differences with the first differences.
 - (f) Fit the model ARIMA(0,1,1) \times (0,1,1)₄, and assess the significance of the estimated coefficients.
 - (g) Perform all of the diagnostic tests on the residuals.
 - (h) Calculate and plot forecasts for the next two years of the series. Be sure to include forecast limits.
- 10.12** The file named boardings contains monthly data on the number of people who boarded transit vehicles (mostly light rail trains and city buses) in the Denver, Colorado, region for August 2000 through December 2005.
- (a) Produce the time series plot for these data. Be sure to use plotting symbols that will help you assess seasonality. Does a stationary model seem reasonable?
 - (b) Calculate and plot the sample ACF for this series. At which lags do you have significant autocorrelation?
 - (c) Fit an ARMA(0,3) \times (1,0)₁₂ model to these data. Assess the significance of the estimated coefficients.
 - (d) Overfit with an ARMA(0,4) \times (1,0)₁₂ model. Interpret the results.

CHAPTER 11

TIME SERIES REGRESSION MODELS

In this chapter, we introduce several useful ideas that incorporate external information into time series modeling. We start with models that include the effects of interventions on time series' normal behavior. We also consider models that assimilate the effects of outliers—observations, either in the observed series or in the error terms, that are highly unusual relative to normal behavior. Lastly, we develop methods to look for and deal with spurious correlation—correlation between series that is artificial and will not help model or understand the time series of interest. We will see that prewhitening of series helps us find meaningful relationships.

11.1 Intervention Analysis

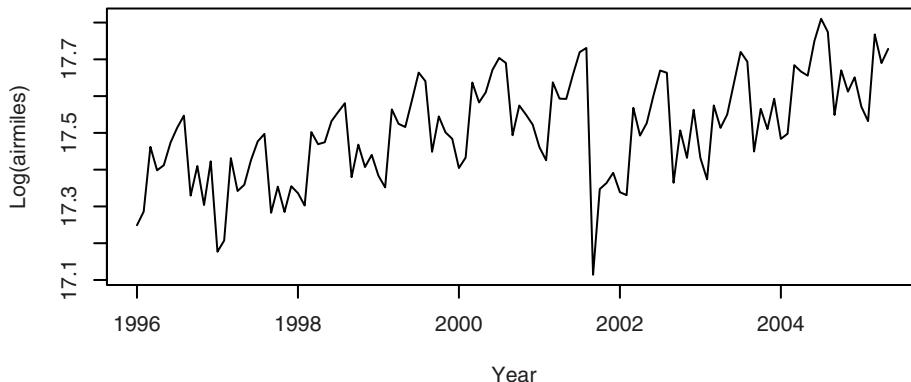
Exhibit 11.1 shows the time plot of the logarithms of monthly airline passenger-miles in the United States from January 1996 through May 2005. The time series is highly seasonal, displaying the fact that air traffic is generally higher during the summer months and the December holidays and lower in the winter months.[†] Also, air traffic was increasing somewhat linearly overall until it had a sudden drop in September 2001. The sudden drop in the number of air passengers in September 2001 and several months thereafter was triggered by the terrorist acts on September 11, 2001, when four planes were hijacked, three of which were crashed into the twin towers of the World Trade Center and the Pentagon and the fourth into a rural field in Pennsylvania. The terrorist attacks of September 2001 deeply depressed air traffic around that period, but air traffic gradually regained the losses as time went on. This is an example of an intervention that results in a change in the trend of a time series.

Intervention analysis, introduced by Box and Tiao (1975), provides a framework for assessing the effect of an intervention on a time series under study. It is assumed that the intervention affects the process by changing the mean function or trend of a time series. Interventions can be natural or man-made. For example, some animal population levels crashed to a very low level in a particular year because of extreme climate in that year. The postcrash annual population level may then be expected to be different from that in the precrash period. Another example is the increase of the speed limit from 65 miles per hour to 70 miles per hour on an interstate highway. This may make driving on

[†] In the exercises, we ask you to display the time series plot using seasonal plotting symbols on a full-screen graph, where the seasonality is quite easy to see.

the highway more dangerous. On the other hand, drivers may stay on the highway for a shorter length of time because of the faster speed, so the net effect of the increased speed limit change is unclear. The effect of the increase in speed limit may be studied by analyzing the mean function of some accident time series data; for example, the quarterly number of fatal car accidents on some segment of an interstate highway. (Note that the autocovariance function of the time series might also be changed by the intervention, but this possibility will not be pursued here.)

Exhibit 11.1 Monthly U.S. Airline Miles: January 1996 through May 2005



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(airmiles)
> plot(log(airmiles), ylab='Log(airmiles)', xlab='Year')
```

We first consider the simple case of a single intervention. The general model for the time series $\{Y_t\}$, perhaps after suitable transformation, is given by

$$Y_t = m_t + N_t \quad (11.1.1)$$

where m_t is the change in the mean function and N_t is modeled as some ARIMA process, possibly seasonal. The process $\{N_t\}$ represents the underlying time series were there no intervention. It is referred to as the natural or unperturbed process, and it may be stationary or nonstationary, seasonal or nonseasonal. Suppose the time series is subject to an intervention that takes place at time T . Before T , m_t is assumed to be identically zero. The time series $\{Y_t, t < T\}$ is referred to as the **preintervention data** and can be used to specify the model for the unperturbed process N_t .

Based on subject matter considerations, the effect of the intervention on the mean function can often be specified up to some parameters. A useful function in this specification is the **step function**

$$S_t^{(T)} = \begin{cases} 1, & \text{if } t \geq T \\ 0, & \text{otherwise} \end{cases} \quad (11.1.2)$$

that is 0 during the preintervention period and 1 throughout the postintervention period. The **pulse function**

$$P_t^{(T)} = S_t^{(T)} - S_{t-1}^{(T)} \quad (11.1.3)$$

equals 1 at $t = T$ and 0 otherwise. That is, $P_t^{(T)}$ is the indicator or dummy variable flagging the time that the intervention takes place. If the intervention results in an immediate and permanent shift in the mean function, the shift can be modeled as

$$m_t = \omega S_t^{(T)} \quad (11.1.4)$$

where ω is the unknown permanent change in the mean due to the intervention. Testing whether $\omega = 0$ or not is similar to testing whether the population means are the same with data in the form of two independent random samples from the two populations. However, the major difference here is that the pre- and postintervention data cannot generally be assumed to be independent and identically distributed. The inherent serial correlation in the data makes the problem more interesting but at the same time more difficult. If there is a **delay** of d time units before the intervention takes effect and d is known, then we can specify

$$m_t = \omega S_{t-d}^{(T)} \quad (11.1.5)$$

In practice, the intervention may affect the mean function gradually, with its full force reflected only in the long run. This can be modeled by specifying m_t as an AR(1)-type model with the error term replaced by a multiple of the lag 1 of $S_t^{(T)}$:

$$m_t = \delta m_{t-1} + \omega S_{t-1}^{(T)} \quad (11.1.6)$$

with the initial condition $m_0 = 0$. After some algebra, it can be shown that

$$m_t = \begin{cases} \omega \frac{1-\delta^{t-T}}{1-\delta}, & \text{for } t > T \\ 0, & \text{otherwise} \end{cases} \quad (11.1.7)$$

Often δ is selected in the range $1 > \delta > 0$. In that case, m_t approaches $\omega/(1 - \delta)$ for large t , which is the ultimate change (gain or loss) for the mean function. Half of the ultimate change is attained when $1 - \delta^{t-T} = 0.5$; that is, when $t = T + \log(0.5)/\log(\delta)$. The duration $\log(0.5)/\log(\delta)$ is called the **half-life** of the intervention effect, and the shorter it is, the quicker the ultimate change is felt by the system. Exhibit 11.2 displays the half-life as a function of δ , which shows that the half-life increases with δ . Indeed, the half-life becomes infinitely large when δ approaches 1.

Exhibit 11.2 Half-life based on an AR(1) Process with Step Function Input

δ	0.2	0.4	0.6	0.8	0.9	1
Half-life	0.43	0.76	1.46	3.11	6.58	∞

It is interesting to note the limiting case when $\delta = 1$. Then $m_t = \omega(T - t)$ for $t \geq T$ and 0 otherwise. The time sequence plot of m_t displays the shape of a ramp with slope ω . This specification implies that the intervention changes the mean function linearly in the postintervention period. This ramp effect (with a one time unit delay) is shown in Exhibit 11.3 (c).

Short-lived intervention effects may be specified using the pulse dummy variable

$$P_t^{(T)} = \begin{cases} 1, & \text{if } t = T \\ 0, & \text{otherwise} \end{cases} \quad (11.1.8)$$

For example, if the intervention impacts the mean function only at $t = T$, then

$$m_t = \omega P_t^{(T)} \quad (11.1.9)$$

Intervention effects that die out gradually may be specified via the AR(1)-type specification

$$m_t = \delta m_{t-1} + \omega P_t^{(T)} \quad (11.1.10)$$

That is, $m_t = \omega\delta^{T-t}$ for $t \geq T$ so that the mean changes immediately by an amount ω and subsequently the change in the mean decreases geometrically by the common factor of δ ; see Exhibit 11.4 (a). Delayed changes can be incorporated by lagging the pulse function. For example, if the change in the mean takes place after a delay of one time unit and the effect dies out gradually, we can specify

$$m_t = \delta m_{t-1} + \omega P_{t-1}^{(T)} \quad (11.1.11)$$

Again, we assume the initial condition $m_0 = 0$.

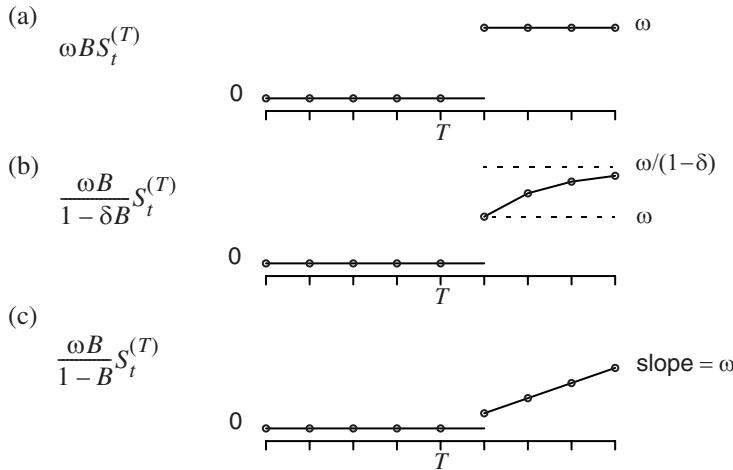
It is useful to write[†] the preceding model in terms of the backshift operator B , where $Bm_t = m_{t-1}$ and $BP_t^{(T)} = P_{t-1}^{(T)}$. Then $(1 - \delta B)m_t = \omega BP_t^{(T)}$. Or, we can write

$$m_t = \frac{\omega B}{1 - \delta B} P_t^{(T)} \quad (11.1.12)$$

Recall $(1 - B)S_t^{(T)} = P_t^{(T)}$, which can be rewritten as $S_t^{(T)} = \frac{1}{1 - B} P_t^{(T)}$.

[†] The remainder of this chapter makes use of the backshift operator introduced in Appendix D on page 106. You may want to review that appendix before proceeding further.

**Exhibit 11.3 Some Common Models for Step Response Interventions
(All are shown with a delay of 1 time unit)**



Several specifications can be combined to model more sophisticated intervention effects.

For example,

$$m_t = \frac{\omega_1 B}{1 - \delta B} P_t^{(T)} + \frac{\omega_2 B}{1 - B} P_t^{(T)} \quad (11.1.13)$$

depicts the situation displayed in Exhibit 11.4 (b) where ω_1 and ω_2 are both greater than zero, and

$$m_t = \omega_0 P_t^{(T)} + \frac{\omega_1 B}{1 - \delta B} P_t^{(T)} + \frac{\omega_2 B}{1 - B} P_t^{(T)} \quad (11.1.14)$$

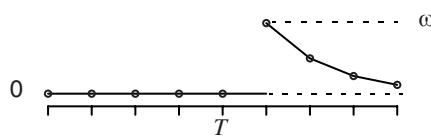
may model situations like Exhibit 11.4 (c) with ω_1 and ω_2 both negative. This last case may model the interesting situation where a special sale may cause strong rush buying, initially so much so that the sale is followed by depressed demand. More generally, we can model the change in the mean function by an ARMA-type specification

$$m_t = \frac{\omega(B)}{\delta(B)} P_t^{(T)} \quad (11.1.15)$$

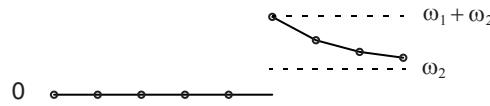
where $\omega(B)$ and $\delta(B)$ are some polynomials in B . Because $(1 - B)S_t^{(T)} = P_t^{(T)}$, the model for m_t can be specified in terms of either the pulse or step dummy variable.

**Exhibit 11.4 Some Common Models for Pulse Response Interventions
(All are shown with a delay of 1 time unit)**

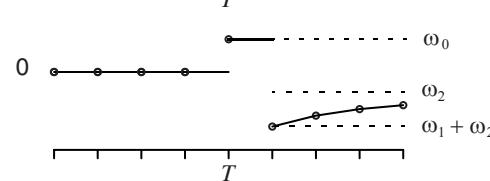
(a) $\frac{\omega B}{1 - \delta B} P_t^{(T)}$



(b) $\left[\frac{\omega_1 B}{1 - \delta B} + \frac{\omega_2 B}{1 - B} \right] P_t^{(T)}$



(c) $\left[\omega_0 + \frac{\omega_1 B}{1 - \delta B} + \frac{\omega_2 B}{1 - B} \right] P_t^{(T)}$



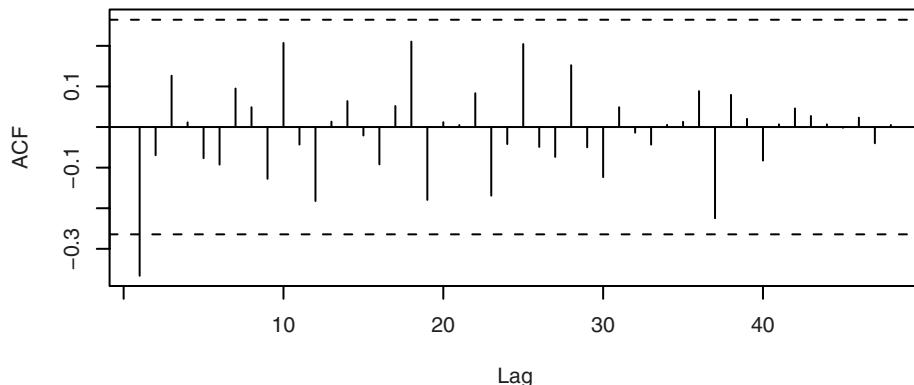
Estimation of the parameters of an intervention model may be carried out by the method of maximum likelihood estimation. Indeed, $Y_t - m_t$ is a seasonal ARIMA process so that the likelihood function equals the joint pdf of $Y_t - m_t$, $t = 1, 2, \dots, n$, which can be computed by methods studied in Chapter 7 or else by the state space modeling methods of Appendix H on page 222.

We now revisit the monthly passenger-airmiles data. Recall that the terrorist acts in September 2001 had lingering depressing effects on air traffic. The intervention may be specified as an AR(1) process with the pulse input at September 2001. But the unexpected turn of events in September 2001 had a strong instantaneous chilling effect on air traffic. Thus, we model the intervention effect (the 9/11 effect) as

$$m_t = \omega_0 P_t^{(T)} + \frac{\omega_1}{1 - \omega_2 B} P_t^{(T)}$$

where T denotes September 2001. In this specification, $\omega_0 + \omega_1$ represents the instantaneous 9/11 effect, and, for $k \geq 1$, $\omega_1(\omega_2)^k$ gives the 9/11 effect k months afterward. It remains to specify the seasonal ARIMA structure of the underlying unperturbed process. Based on the preintervention data, an $\text{ARIMA}(0,1,1) \times (0,1,0)_{12}$ model was tentatively specified for the unperturbed process; see Exhibit 11.5.

Exhibit 11.5 Sample ACF for $(1-B)(1-B^{12}) \log(\text{Air Passenger Miles})$ Over the Preintervention Period



```
> acf(as.vector(diff(diff(window(log(airmiles), end=c(2001,8)), 12))), lag.max=48)
```

Model diagnostics of the fitted model suggested that a seasonal MA(1) coefficient was needed and the existence of some *additive* outliers occurring in December 1996, January 1997, and December 2002. (Outliers will be discussed in more detail later; here additive outliers may be regarded as interventions of unknown nature that have a pulse response function.) Hence, the model is specified as an ARIMA(0,1,1) \times (0,1,1)₁₂ plus the 9/11 intervention and three additive outliers. The fitted model is summarized in Exhibit 11.6.

**Exhibit 11.6 Estimation of Intervention Model for Logarithms of Air Miles
(Standard errors are shown below the estimates)**

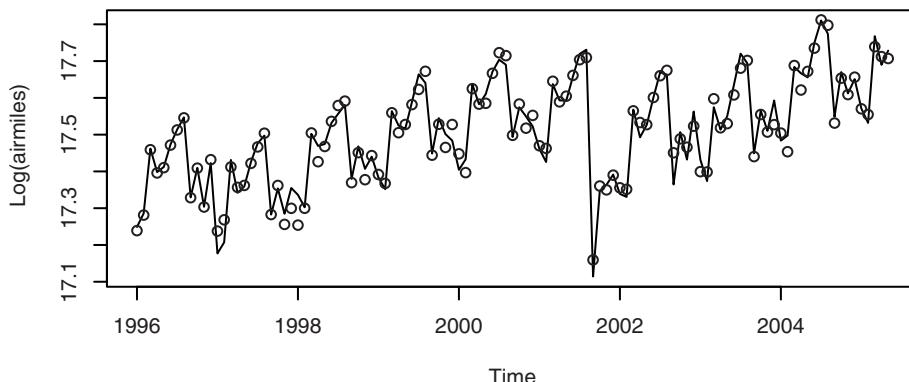
θ	Θ	Dec96	Jan97	Dec02	ω_0	ω_1	ω_2
0.383	0.650	0.099	-0.069	0.081	-0.095	-0.27	0.814
(0.093)	(0.119)	(0.023)	(0.022)	(0.020)	(0.046)	(0.044)	(0.098)

σ^2 estimated as 0.000672: log-likelihood = 219.99, AIC= -423.98

```
> air.m1=arimax(log(airmiles), order=c(0,1,1),
  seasonal=list(order=c(0,1,1), period=12),
  xtransf=data.frame(I911=1*(seq(airmiles)==69),
  I1911=1*(seq(airmiles)==69)), transfer=list(c(0,0), c(1,0)),
  xreg=data.frame(Dec96=1*(seq(airmiles)==12),
  Jan97=1*(seq(airmiles)==13), Dec02=1*(seq(airmiles)==84)),
  method='ML')
> air.m1
```

Model diagnostics suggested that the fitted model above provides a good fit to the data. The open circles in the time series plot shown in Exhibit 11.7 represent the fitted values from the final estimated model. They indicate generally good agreement between the model and the data.

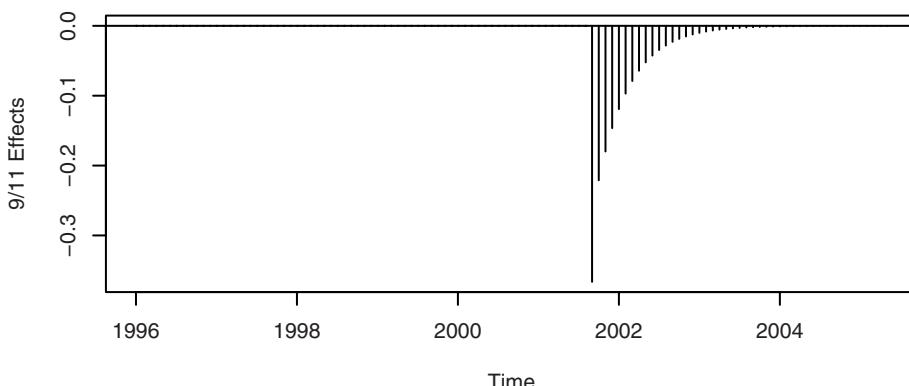
Exhibit 11.7 Logs of Air Passenger Miles and Fitted Values



```
> plot(log(airmiles), ylab='Log(airmiles)')
> points(fitted(air.m1))
```

The fitted model estimates that the 9/11 intervention reduced air traffic by $31\% = \{1 - \exp(-0.0949 - 0.2715)\} \times 100\%$ in September 2001, and air traffic k months later was lowered by $\{1 - \exp(-0.2715 \times 0.8139^k)\} \times 100\%$. Exhibit 11.8 graphs the estimated 9/11 effects on air traffic, which indicate that air traffic regained its losses toward the end of 2003.

Exhibit 11.8 The Estimated 9/11 Effects for the Air Passenger Series



```
> Nine11p=1*(seq(airmiles)==69)
> plot(ts(Nine11p*(-0.0949) +
```

```
filter(Nine11p,filter=.8139,method='recursive', side=1)*
(-0.2715),frequency=12,start=1996),ylab='9/11 Effects',
type='h'); abline(h=0)
```

11.2 Outliers

Outliers refer to atypical observations that may arise because of measurement and/or copying errors or because of abrupt, short-term changes in the underlying process. For time series, two kinds of outliers can be distinguished, namely **additive** outliers and **innovative** outliers. These two kinds of outliers are often abbreviated as AO and IO, respectively. An additive outlier occurs at time T if the underlying process is perturbed additively at time T so that the data equal

$$Y'_t = Y_t + \omega_A P_t^{(T)} \quad (11.2.1)$$

where $\{Y_t\}$ is the unperturbed process. Henceforth in this section, Y' denotes the observed process that may be affected by some outliers and Y the unperturbed process should there be no outliers. Thus, $Y'_T = Y_T + \omega_A$ but $Y'_t = Y_t$ otherwise, so the time series is only affected at time T if it has an additive outlier at T . An additive outlier can also be treated as an intervention that has a pulse response at T so that $m_t = \omega_A P_t^{(T)}$.

On the other hand, an innovative outlier occurs at time t if the error (also known as an innovation) at time t is perturbed (that is, the errors equal $e'_t = e_t + \omega_I P_t^{(T)}$, where e_t is a zero-mean white noise process). So, $e'_T = e_T + \omega_I$ but $e'_t = e_t$ otherwise. Suppose that the unperturbed process is stationary and admits an MA(∞) representation

$$Y_t = e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots$$

Consequently, the perturbed process can be written

$$\begin{aligned} Y'_t &= e'_t + \psi_1 e'_{t-1} + \psi_2 e'_{t-2} + \dots \\ &= [e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots] + \psi_{t-T} \omega_I \end{aligned}$$

or

$$Y'_t = Y_t + \psi_{t-T} \omega_I \quad (11.2.2)$$

where $\psi_0 = 1$ and $\psi_j = 0$ for negative j . Thus, an innovative outlier at T perturbs all observations on and after T , although with diminishing effect, as the observation is further away from the origin of the outlier.

To detect whether an observation is an AO or IO, we use the AR(∞) representation of the unperturbed process to define the residuals:

$$a_t = Y'_t - \pi_1 Y'_{t-1} - \pi_2 Y'_{t-2} - \dots \quad (11.2.3)$$

For simplicity, we assume the process has zero mean and that the parameters are known. In practice, the unknown parameter values are replaced by their estimates from the possibly perturbed data. Under the null hypothesis of no outliers and for large samples, this

has a negligible effect on the properties of the test procedures described below. If the series has exactly one IO at time T , then the residual $a_T = \omega_I + e_T$ but $a_t = e_t$ otherwise. So ω_I can be estimated by $\tilde{\omega}_I = a_T$ with variance equal to σ^2 . Thus, a test statistic for testing for an IO at T is

$$\lambda_{1,T} = \frac{a_T}{\sigma} \quad (11.2.4)$$

which has (approximately) a standard normal distribution under the null hypothesis that there are no outliers in the time series. When T is known beforehand, the observation in question is declared an outlier if the corresponding standardized residual exceeds 1.96 in magnitude at the 5% significance level. In practice, there is often no prior knowledge about T , and the test is applied to all observations. In addition, σ will need to be estimated. A simple conservative procedure is to use the Bonferroni rule for controlling the overall error rate of multiple tests. Let

$$\lambda_1 = \max_{1 \leq t \leq n} |\lambda_{1,t}| \quad (11.2.5)$$

be attained at $t = T$. Then the T th observation is deemed an IO if λ_1 exceeds the upper $0.025/n \times 100$ percentile of the standard normal distribution. This procedure guarantees that there is at most a 5% probability of a false detection of an IO. Note that an outlier will inflate the maximum likelihood estimate of σ , so if there is no adjustment for outliers, the power of most tests is usually reduced. A robust estimate of the noise standard deviation may be used in lieu of the maximum likelihood estimate to increase the power of the test. For example, σ can be more robustly estimated by the mean absolute residual times $\sqrt{2/\pi}$.

The detection of an AO is more complex. Suppose that the process admits an AO at T and is otherwise free of outliers. Then it can be shown that

$$a_t = -\omega_A \pi_{t-T} + e_t \quad (11.2.6)$$

where $\pi_0 = -1$ and $\pi_j = 0$ for negative j . Hence, $a_t = e_t$ for $t < T$, $a_T = \omega_A + e_T$, $a_{T+1} = -\omega_A \pi_1 + e_{T+1}$, $a_{T+2} = -\omega_A \pi_2 + e_{T+2}$, and so forth. A least squares estimator of ω_A is

$$\tilde{\omega}_{T,A} = -\rho^2 \sum_{t=1}^n \pi_{t-T} a_t \quad (11.2.7)$$

where $\rho^2 = (1 + \pi_1^2 + \pi_2^2 + \dots + \pi_{n-T}^2)^{-1}$, with the variance of the estimate being equal to $\rho^2 \sigma^2$. We can then define

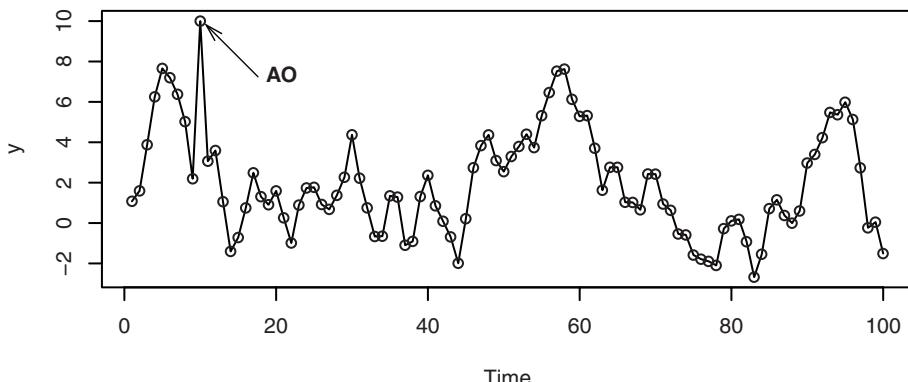
$$\lambda_{2,T} = \frac{\tilde{\omega}_{T,A}}{\rho \sigma} \quad (11.2.8)$$

as the test statistic for testing the null hypothesis that the time series has no outliers versus the alternative hypothesis of an AO at T . As before, ρ and σ will need to be estimated. The test statistic $\lambda_{2,T}$ is approximately distributed as $N(0,1)$ under the null hypothesis. Again, T is often unknown, and the test is applied repeatedly to each time point. The Bonferroni rule may again be applied to control the overall error rate. Furthermore, the nature of an outlier is not known beforehand. In the case where an outlier

is detected at T , it may be classified to be an IO if $|\lambda_{1,T}| > |\lambda_{2,T}|$ and an AO otherwise. See Chang et al. (1988) for another approach to classifying the nature of an outlier. When an outlier is found, it can be incorporated into the model, and the outlier-detection procedure can then be repeated with the refined model until no more outliers are found.

As a first example, we simulated a time series of length $n = 100$ from the ARIMA(1,0,1) model with $\phi = 0.8$ and $\theta = -0.5$. We then changed the 10th observation from -2.13 to 10 (that is, $\omega_A = 12.13$); see Exhibit 11.9. Based on the sample ACF, PACF and EACF, an AR(1) model was tentatively identified. Based on the Bonferroni rule, the 9th, 10th, and 11th observations were found to be possible additive outliers with the corresponding robustified test statistics being -3.54 , 9.55 , and -5.20 . The test for IO revealed that the 10th and 11th observations may be IO, with the corresponding robustified test statistics being 7.11 and -6.64 . Because among the tests for AO and IO the largest magnitude occurs for the test for AO at $T = 10$, the 10th observation was tentatively marked as an AO. Note that the nonrobustified test statistic for AO at $T = 10$ equals 7.49 , which is substantially less than the more robust test value of 9.55 , showing that robustifying the estimate of the noise standard deviation does increase the power of the test. After incorporating the AO in the model, no more outliers were found. However, the lag 1 residual ACF was significant, suggesting the need for an MA(1) component. Hence, an ARIMA(1,0,1) + AO at $T = 10$ model was fitted to the data. This model was found to have no additional outliers and passed all model diagnostic checks.

Exhibit 11.9 Simulated ARIMA(1,0,1) Process with an Additive Outlier



> The extensive R code for the simulation and analysis of this example may be found in the R code script file for Chapter 11.

For a real example, we return to the seasonal ARIMA(0,1,1) \times (0,1,1)₁₂ model that we fitted to the carbon dioxide time series in Chapter 10. The time series plot of the standardized residuals from this model, shown in Exhibit 10.11 on page 238, showed a suspiciously large standardized residual in September 1998. Calculation shows that there is no evidence of an additive outlier, as $\lambda_{2,t}$ is not significantly large for any t . However, the robustified $\lambda_1 = \max_{1 \leq t \leq n} |\lambda_{1,t}| = 3.7527$, which is attained at $t = 57$, cor-

responding to September 1998. The Bonferroni critical value with $\alpha = 5\%$ and $n = 132$ is 3.5544. So our observed λ_1 is large enough to claim significance for an innovation outlier in September 1998. Exhibit 11.10 shows the results of fitting the ARIMA(0,1,1) \times (0,1,1)₁₂ model with an IO at $t = 57$ to the CO₂ time series. These results should be compared with the earlier results shown in Exhibit 10.10 on page 237, where the outlier was not taken into account. Notice that the estimates of θ and Θ have not changed very much, the AIC is better (that is, smaller), and the IO effect is highly significant. Diagnostics based on this model turn out to be excellent, no further outliers are detected, and we have a very adequate model for this seasonal time series.

Exhibit 11.10 ARIMA(0,1,1) \times (0,1,1)₁₂ Model with IO at $t = 57$ for CO₂ Series

Coefficient	θ	Θ	IO-57
Estimate	0.5925	0.8274	2.6770
Standard Error	0.0775	0.1016	0.7246

$$\hat{\sigma}_e^2 = 0.4869; \text{log-likelihood} = -133.08, \text{AIC} = 272.16$$

```
> m1.co2=arima(co2,order=c(0,1,1),seasonal=list(order=c(0,1,1),
  period=12)); m1.co2
> detectAO(m1.co2); detectIO(m1.co2)
> m4.co2=arimax(co2,order=c(0,1,1),seasonal=list(order=c(0,1,1),
  period=12),io=c(57)); m4.co2
```

11.3 Spurious Correlation

A main purpose of building a time series model is for forecasting, and the ARIMA model does this by exploiting the autocorrelation pattern in the data. Often, the time series under study may be related to, or led by, some other covariate time series. For example, Stige et al. (2006) found that pasture production in Africa is generally related to some climatic indices. In such cases, better understanding of the underlying process and/or more accurate forecasts may be achieved by incorporating relevant covariates into the time series model.

Let $Y = \{Y_t\}$ be the time series of the response variable and $X = \{X_t\}$ be a covariate time series that we hope will help explain or forecast Y . To explore the correlation structure between X and Y and their lead-led relationship, we define the cross-covariance function $\gamma_{t,s}(X, Y) = \text{Cov}(X_t, Y_s)$ for each pair of integers t and s . Stationarity of a univariate time series can be easily extended to the case of multivariate time series. For example, X and Y are jointly (weakly) stationary if their means are constant and the covariance $\gamma_{t,s}(X, Y)$ is a function of the time difference $t - s$. For jointly stationary processes, the **cross-correlation function** between X and Y at lag k can then be defined by $\rho_k(X, Y) = \text{Corr}(X_t, Y_{t-k}) = \text{Corr}(X_{t+k}, Y_t)$. Note that if $Y = X$, the cross-correlation becomes the autocorrelation of Y at lag k . The coefficient $\rho_0(Y, X)$ measures the contemporaneous linear association between X and Y , whereas $\rho_k(X, Y)$ measures the linear association between X_t and that of Y_{t-k} . Recall that the autocorrelation function is an

even function, that is, $\rho_k(Y, Y) = \rho_{-k}(Y, Y)$. (This is because $\text{Corr}(Y_t, Y_{t-k}) = \text{Corr}(Y_{t-k}, Y_t) = \text{Corr}(Y_t, Y_{t+k})$, by stationarity.) However, the cross-correlation function is generally not an even function since $\text{Corr}(X_t, Y_{t-k})$ need not equal $\text{Corr}(X_t, Y_{t+k})$.

As an illustration, consider the regression model

$$Y_t = \beta_0 + \beta_1 X_{t-d} + e_t \quad (11.3.1)$$

where the X 's are independent, identically distributed random variables with variance σ_X^2 and the e 's are also white noise with variance σ_e^2 and are independent of the X 's. It can be checked that the cross-correlation function (CCF) $\rho_k(X, Y)$ is identically zero except for lag $k = -d$, where

$$\rho_{-d}(X, Y) = \frac{\beta_1 \sigma_X}{\sqrt{\beta_1^2 \sigma_X^2 + \sigma_e^2}} \quad (11.3.2)$$

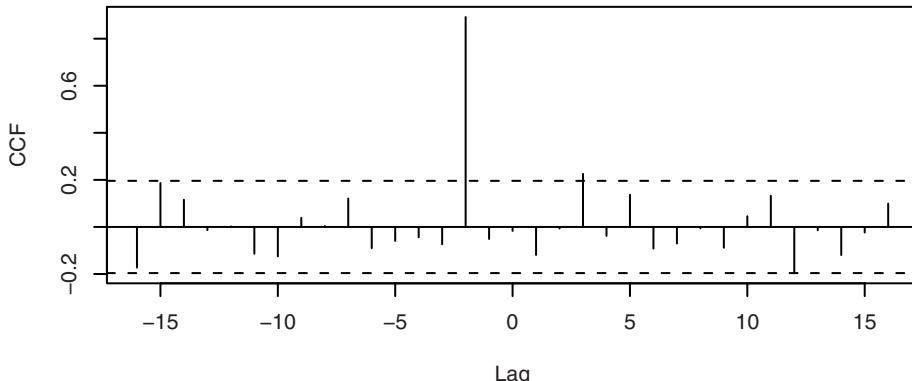
In this case, the theoretical CCF is nonzero only at lag $-d$, reflecting the fact that X is “leading” Y by d units of time. The CCF can be estimated by the **sample cross-correlation function** (sample CCF) defined by

$$r_k(X, Y) = \frac{\sum (X_t - \bar{X})(Y_{t-k} - \bar{Y})}{\sqrt{\sum (X_t - \bar{X})^2} \sqrt{\sum (Y_{t-k} - \bar{Y})^2}} \quad (11.3.3)$$

where the summations are done over all data where the summands are available. The sample CCF becomes the sample ACF when $Y = X$. The covariate X is independent of Y if and only if $\beta_1 = 0$, in which case the sample autocorrelation $r_k(X, Y)$ is approximately normally distributed with zero mean and variance $1/n$, where n is the sample size—the number of pairs of (X_t, Y_t) available. Sample cross-correlations that are larger than $1.96/\sqrt{n}$ in magnitude are then deemed significantly different from zero.

We have simulated 100 pairs of (X_t, Y_t) from the model of Equation (11.3.1) with $d = 2$, $\beta_0 = 0$, and $\beta_1 = 1$. The X 's and e 's are generated as normal random variables distributed as $N(0, 1)$ and $N(0, 0.25)$, respectively. Theoretically, the CCF should then be zero except at lag -2 , where it equals $\rho_{-2}(X, Y) = 1/\sqrt{1+0.25} = 0.8944$. Exhibit 11.11 shows the sample CCF of the simulated data, which is significant at lags -2 and 3 . But the sample CCF at lag 3 is quite small and only marginally significant. Such a false alarm is not unexpected as the exhibit displays a total of 33 sample CCF values out of which we may expect $33 \times 0.05 = 1.65$ false alarms on average.

Exhibit 11.11 Sample Cross-Correlation from Equation (11.3.1) with $d = 2$



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> set.seed(12345); X=rnorm(105); Y=zlag(X, 2)+.5*rnorm(105)
> X=ts(X[-(1:5)], start=1, freq=1); Y=ts(Y[-(1:5)], start=1, freq=1)
> ccf(X, Y, ylab='CCF')
```

Even though X_{t-2} correlates with Y_t , the regression model considered above is rather restrictive, as X and Y are each white noise series. For stationary time series, the response variable and the covariate are each generally autocorrelated, and the error term of the regression model is also generally autocorrelated. Hence a more useful regression model is given by

$$Y_t = \beta_0 + \beta_1 X_{t-d} + Z_t \quad (11.3.4)$$

where Z_t may follow some ARIMA(p, d, q) model. Even if the processes X and Y are independent of each other ($\beta_1 = 0$), the autocorrelations in Y and X have the unfortunate consequence of implying that the sample CCF is no longer approximately $N(0, 1/n)$. Under the assumption that both X and Y are stationary and that they are independent of each other, it turns out that the sample variance tends to be different from $1/n$. Indeed, it may be shown that the variance of $\sqrt{n}r_k(X, Y)$ is approximately

$$1 + 2 \sum_{k=1}^{\infty} \rho_k(X)\rho_k(Y) \quad (11.3.5)$$

where $\rho_k(X)$ is the autocorrelation of X at lag k and $\rho_k(Y)$ is similarly defined for the Y -process. For refinement of this asymptotic result, see Box et al. (1994, p. 413). Suppose X and Y are both AR(1) processes with AR(1) coefficients ϕ_X and ϕ_Y , respectively. Then $r_k(X, Y)$ is approximately normally distributed with zero mean, but the variance is now approximately equal to

$$\frac{1 + \phi_X\phi_Y}{n(1 - \phi_X\phi_Y)} \quad (11.3.6)$$

When both AR(1) coefficients are close to 1, the ratio of the sampling variance of $r_k(X, Y)$ to the nominal value of $1/n$ approaches infinity. Thus, the unquestioned use of the $1/n$ rule in deciding the significance of the sample CCF may lead to many more false positives than the nominal 5% error rate, even though the response and covariate time series are independent of each other. Exhibit 11.12 shows some numerical results for the case where $\phi_X = \phi_Y = \phi$.

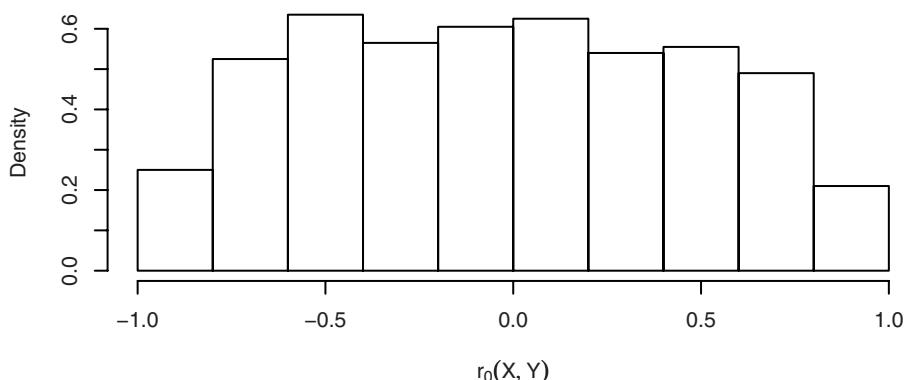
Exhibit 11.12 Asymptotic Error Rates of a Nominal 5% Test of Independence for a Pair of AR(1) Processes

$\phi = \phi_X = \phi_Y$	0.00	0.15	0.30	0.45	0.60	0.75	0.90
Error Rate	5%	6%	7%	11%	18%	30%	53%

```
> phi=seq(0,.95,.15)
> rejection=2*(1-pnorm(1.96*sqrt((1-phi^2)/(1+phi^2))))
> M=signif(rbind(phi,rejection),2)
> rownames(M)=c('phi', 'Error Rate')
> M
```

The problem of inflated variance of the sample cross-correlation coefficients becomes more acute for nonstationary data. In fact, the sample cross-correlation coefficients may no longer be approximately normally distributed even with a large sample size. Exhibit 11.13 displays the histogram of 1000 simulated lag zero cross-correlations between two independent IMA(1,1) processes each of size 500. An MA(1) coefficient of $\theta = 0.8$ was used for both simulated processes. Note that the distribution of $r_0(X, Y)$ is far from normal and widely dispersed between -1 and 1 . See Phillips (1998) for a relevant theoretical discussion.

Exhibit 11.13 Histogram of 1000 Sample Lag Zero Cross-Correlations of Two Independent IMA(1,1) Processes Each of Size 500

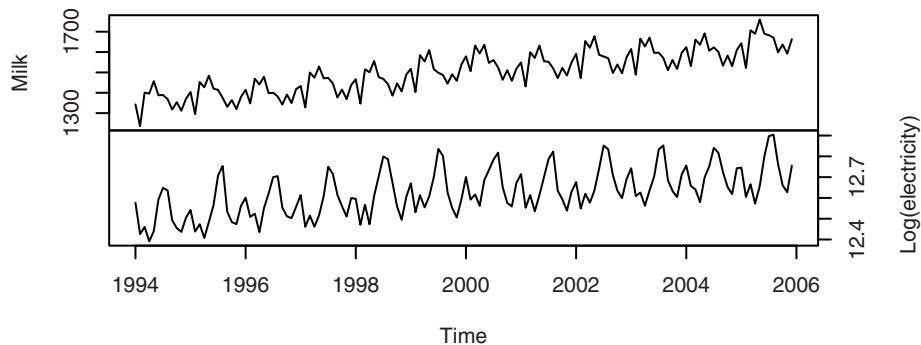


```
> set.seed(23457)
> correlation.v=NULL; B=1000; n=500
> for (i in 1:B) {x=cumsum(arima.sim(model=list(ma=.8),n=n))
> y=cumsum(arima.sim(model=list(ma=.8),n=n))
> correlation.v=c(correlation.v,ccf(x,y,lag.max=1,
+ plot=F)$acf[2])}
> hist(correlation.v,prob=T,xlab=expression(r[0](X,Y)))
```

These results provide insight into why we sometimes obtain nonsense (spurious) correlation between time series variables. The phenomenon of spurious correlation was first studied systematically by Yule (1926).

As an example, the monthly milk production and the logarithms of monthly electricity production in the United States from January 1994 to December 2005 are shown in Exhibit 11.14. Both series have an upward trend and are highly seasonal.

Exhibit 11.14 Monthly Milk Production and Logarithms of Monthly Electricity Production in the U.S.

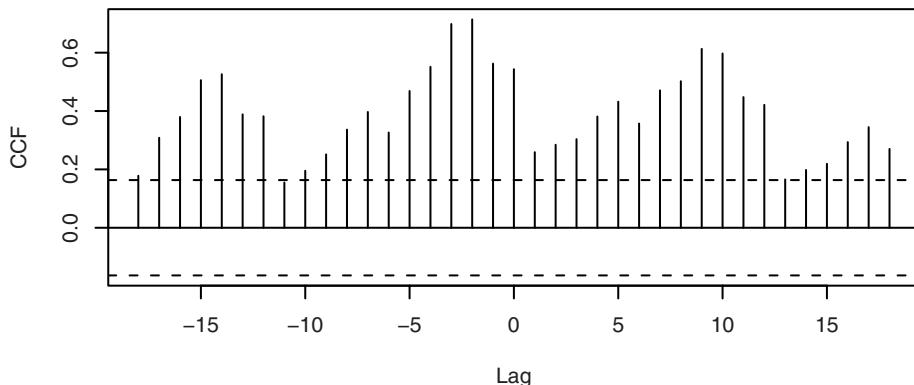


```
> data(milk); data(electricity)
> milk.electricity=ts.intersect(milk,log(electricity))
> plot(milk.electricity,yax.flip=T)
```

Calculation shows that these series have a cross-correlation coefficient at lag zero of 0.54, which is “statistically significantly different from zero” as judged against the standard error criterion of $1.96/\sqrt{n} = 0.16$. Exhibit 11.15 displays the strong cross-correlations between these two variables at a large number of lags.

Needless to say, it is difficult to come up with a plausible reason for the relationship between monthly electricity production and monthly milk production. The nonstationarity in the milk production series and in the electricity series is more likely the cause of the spurious correlations found between the two series. The following section contains further discussion of this example.

Exhibit 11.15 Sample Cross-Correlation Between Monthly Milk Production and Logarithm of Monthly Electricity Production in the U.S.



```
> ccf(as.vector(milk.electricity[,1]),
      as.vector(milk.electricity[,2]), ylab='CCF')
```

11.4 Prewitthing and Stochastic Regression

In the preceding section, we found that with strongly autocorrelated data it is difficult to assess the dependence between the two processes. Thus, it is pertinent to disentangle the linear association between X and Y , say, from their autocorrelation. A useful device for doing this is prewhitening. Recall that, for the case of stationary X and Y that are independent of each other, the variance of $r_k(X, Y)$ is approximately

$$\frac{1}{n} \left[1 + 2 \sum_{k=1}^{\infty} \rho_k(X)\rho_k(Y) \right] \quad (11.4.1)$$

An examination of this formula reveals that the approximate variance is $1/n$ if either one (or both) of X or Y is a white noise process. In practice, the data may be nonstationary, but they may be transformed to approximately white noise by replacing the data by the residuals from a fitted ARIMA model. For example, if X follows an ARIMA(1,1,0) model with no intercept term, then

$$\tilde{X}_t = X_t - X_{t-1} - \phi(X_{t-1} - X_{t-2}) = 1 - (1 + \phi B) + \phi B^2]X_t \quad (11.4.2)$$

is white noise. More generally, if X_t follows some invertible ARIMA(p, d, q) model, then it admits an AR(∞) representation

$$\tilde{X}_t = (1 - \pi_1 B - \pi_2 B^2 - \dots)X_t = \pi(B)X_t$$

where the \tilde{X} 's are white noise. The process of transforming the X 's to the \tilde{X} 's via the filter $\pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \dots$ is known as **whitening** or **prewhitening**. We now can

study the CCF between X and Y by prewhitening the Y and X using the *same* filter based on the X process and then computing the CCF of \tilde{Y} and \tilde{X} ; that is, the prewhitened Y and X . Since prewhitening is a *linear* operation, any linear relationships between the original series will be preserved after prewhitening. Note that we have abused the terminology, as \tilde{Y} need not be white noise because the filter $\pi(B)$ is tailor-made only to transform X to a white noise process—not Y . We assume, furthermore, that \tilde{Y} is stationary. This approach has two advantages: (i) the statistical significance of the sample CCF of the prewhitened data can be assessed using the cutoff $1.96/\sqrt{n}$, and (ii) the theoretical counterpart of the CCF so estimated is proportional to certain regression coefficients.

To see (ii), consider a more general regression model relating X to Y and, without loss of generality, assume both processes have zero mean:

$$Y_t = \sum_{j=-\infty}^{\infty} \beta_j X_{t-j} + Z_t \quad (11.4.3)$$

where X is independent of Z and the coefficients β are such that the process is well-defined. In this model, the coefficients β_k could be nonzero for any integer k . However, in real applications, the doubly infinite sum is often a finite sum so that the model simplifies to

$$Y_t = \sum_{j=m_1}^{m_2} \beta_j X_{t-j} + Z_t, \quad (11.4.4)$$

which will be assumed below even though we retain the doubly infinite summation notation for ease of exposition. If the summation ranges only over a finite set of *positive* indices, then X leads Y and the covariate X serves as a useful **leading indicator** for future Y 's. Applying the filter $\pi(B)$ to both sides of this model, we get

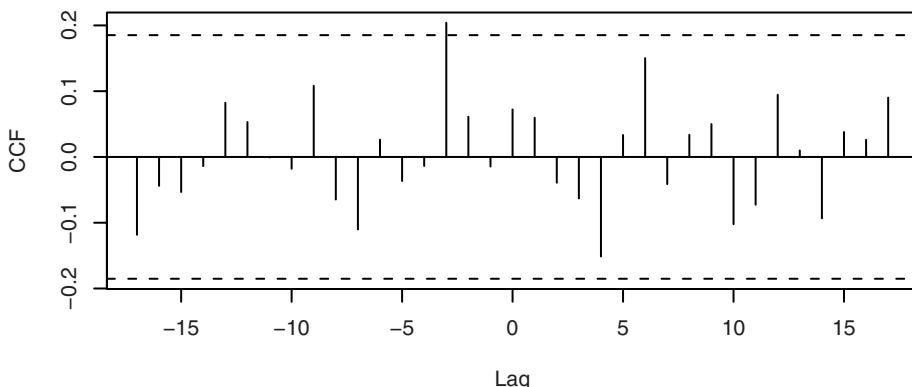
$$\tilde{Y}_t = \sum_{k=-\infty}^{\infty} \beta_k \tilde{X}_{t-k} + \tilde{Z}_t \quad (11.4.5)$$

where $\tilde{Z}_t = Z_t - \pi_1 Z_{t-1} - \pi_2 Z_{t-2} - \dots$. The prewhitening procedure thus orthogonalizes the various lags of X in the original regression model. Because \tilde{X} is a white noise sequence and \tilde{X} is independent of \tilde{Z} , the theoretical cross-correlation coefficient between \tilde{X} and \tilde{Y} at lag k equals $\beta_{-k}(\sigma_{\tilde{X}}/\sigma_{\tilde{Y}})$. In other words, the theoretical cross-correlation of the prewhitened processes at lag k is proportional to the regression coefficient β_{-k} .

For a quick preliminary analysis, an approximate prewhitening can be done easily by first differencing the data (if needed) and then fitting an approximate AR model with the order determined by minimizing the AIC. For example, for the milk production and electricity consumption data, both are highly seasonal and contain trends. Consequently, they can be differenced with both regular differencing and seasonal differencing, and then the prewhitening can be carried out by filtering both differenced series by an AR model fitted to the differenced milk data. Exhibit 11.16 shows the sample CCF between the prewhitened series. None of the cross-correlations are now significant except for lag -3 , which is just marginally significant. The lone significant cross-correlation is likely a false alarm since we expect about 1.75 false alarms out of the 35 sample cross-correla-

tions examined. Thus, it seems that milk production and electricity consumption are in fact largely uncorrelated, and the strong cross-correlation pattern found between the raw data series is indeed spurious.

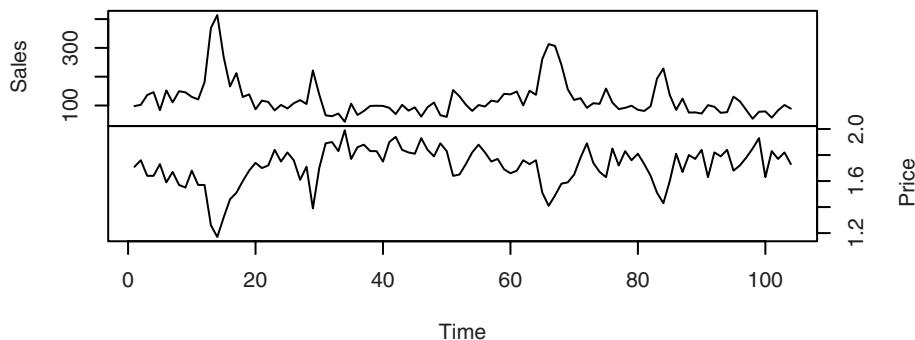
Exhibit 11.16 Sample CCF of Prewhitened Milk and Electricity Production



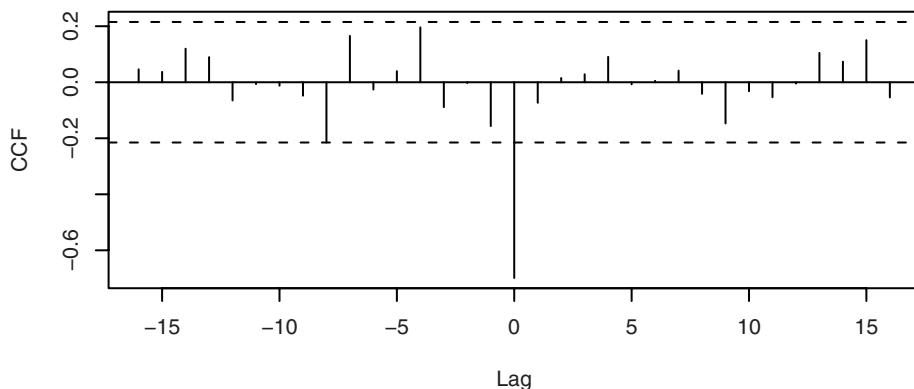
```
> me.dif=ts.intersect(diff(diff(milk,12)),
  diff(diff(log(electricity),12)))
> prewhiten(as.vector(me.dif[,1]),as.vector(me.dif[,2]),
  ylab='CCF')
```

The model defined by Equation (11.3.4) on page 262 is known variously as the **transfer-function model**, the distributed-lag model, or the dynamic regression model. The specification of which lags of the covariate enter into the model is often done by inspecting the sample cross-correlation function based on the prewhitened data. When the model appears to require a fair number of lags of the covariate, the regression coefficients may be parsimoniously specified via an ARMA specification similar to the case of intervention analysis; see Box et al. (1994, Chapter 11) for some details. We illustrate the method below with two examples where only one lag of the covariate appears to be needed. The specification of the stochastic noise process Z_t can be done by examining the residuals from an ordinary least squares (OLS) fit of Y on X using the techniques learned in earlier chapters.

Our first example of this section is a sales and price dataset of a certain potato chip from Bluebird Foods Ltd., New Zealand. The data consist of the log-transformed weekly unit sales of large packages of standard potato chips sold and the weekly average price over a period of 104 weeks from September 20, 1998 through September 10, 2000; see Exhibit 11.17. The logarithmic transformation is needed because the sales data are highly skewed to the right. These data are clearly nonstationary. Exhibit 11.18 shows that, after differencing and using prewhitened data, the CCF is significant only at lag 0, suggesting a strong contemporaneous negative relationship between lag 1 of price and sales. Higher prices are associated with lower sales.

Exhibit 11.17 Weekly Log(Sales) and Price for Bluebird Potato Chips

```
> data(bluebird)
> plot(bluebird,yax.flip=T)
```

Exhibit 11.18 Sample Cross Correlation Between Prewhitened Differenced Log(Sales) and Price of Bluebird Potato Chips

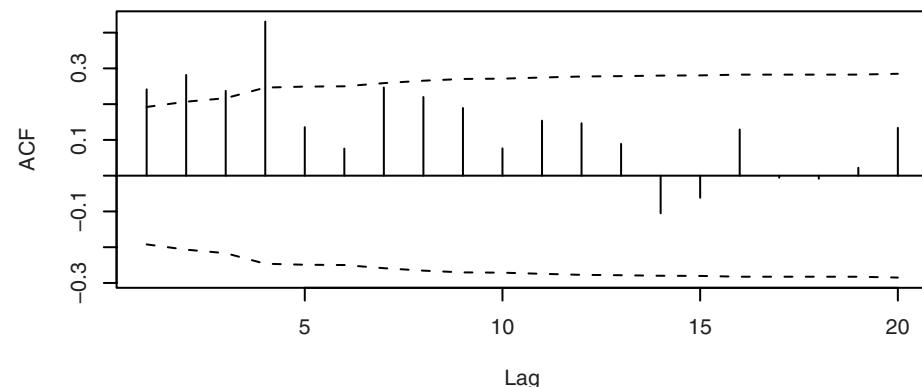
```
> prewhiten(y=diff(bluebird) [,1],x=diff(bluebird) [,2],ylab='CCF')
```

Exhibit 11.19 reports the estimates from the OLS regression of $\log(\text{sales})$ on price. The residuals are, however, autocorrelated, as can be seen from their sample ACF and PACF displayed in Exhibits 11.20 and 11.21, respectively. Indeed, the sample autocorrelations of the residuals are significant for the first four lags, whereas the sample partial autocorrelations are significant at lags 1, 2, 4, and 14.

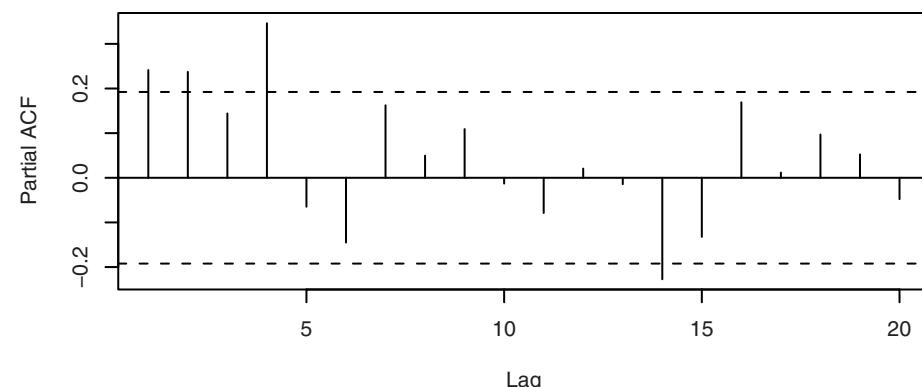
Exhibit 11.19 OLS Regression Estimates of Log(Sales) on Price

	Estimate	Std. Error	t value	Pr(>)
Intercept	15.90	0.2170	73.22	< 0.0001
Price	-2.489	0.1260	-19.75	< 0.0001

```
> sales=bluebird[,1]; price=bluebird[,2]
> chip.m1=lm(sales~price,data=bluebird)
> summary(chip.m1)
```

Exhibit 11.20 Sample ACF of Residuals from OLS Regression of Log(Sales) on Price

```
> acf(residuals(chip.m1), ci.type='ma')
```

Exhibit 11.21 Sample PACF of Residuals from OLS Regression of Log(Sales) on Price

```
> pacf(residuals(chip.m1))
```

The sample EACF of the residuals, shown in Exhibit 11.22, contains a triangle of zeros with a vertex at (1,4), thereby suggesting an ARMA(1,4) model. Hence, we fit a regression model of log(sales) on price with an ARMA(1,4) error.

Exhibit 11.22 The Sample EACF of the Residuals from the OLS Regression of Log(Sales) on Price

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	0	0	x	x	0	0	0	0	0	0
1	x	0	0	x	0	0	0	0	0	0	0	0	0	0
2	x	x	0	x	0	0	0	0	0	0	0	0	0	0
3	x	x	0	x	0	0	0	0	0	0	0	0	0	0
4	0	x	x	0	0	0	0	0	0	0	0	0	0	0
5	x	x	x	0	x	0	0	0	0	0	0	0	0	0
6	x	x	0	x	x	x	0	0	0	0	0	0	0	0
7	x	0	x	0	0	0	0	0	0	0	0	0	0	0

```
> eacf(residuals(chip.m1))
```

It turns out that the estimates of the AR(1) coefficient and the MA coefficients θ_1 and θ_3 are not significant, and hence a model fixing these coefficients to be zero was subsequently fitted and reported in Exhibit 11.23.

Exhibit 11.23 Maximum Likelihood Estimates of a Regression Model of Log(sales) on Price with a Subset MA(4) for the Errors

Parameter	θ_1	θ_2	θ_3	θ_4	Intercept	Price
Estimate	0	-0.2884	0	-0.5416	15.86	-2.468
Standard Error	0	0.0794	0	0.01167	0.1909	0.1100

σ^2 estimated as 0.02623: log likelihood = 41.02, AIC = -70.05

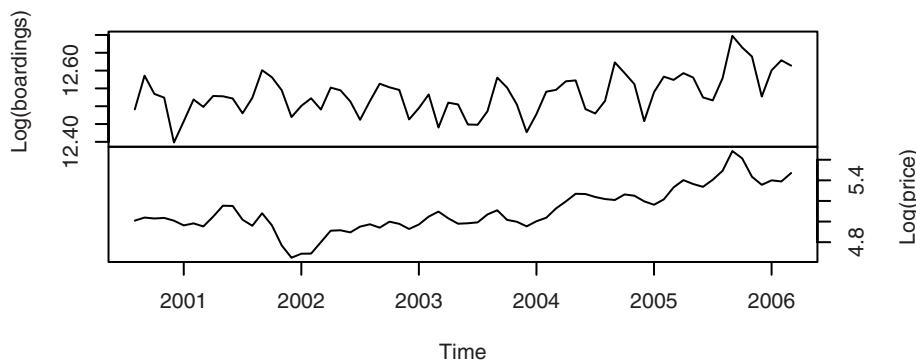
```
> chip.m2=arima(sales,order=c(1,0,4),xreg=data.frame(price))
> chip.m2
> chip.m3=arima(sales,order=c(1,0,4),xreg=data.frame(price),
+                 fixed=c(NA,0,NA,0,NA,NA,NA)); chip.m3
> chip.m4=arima(sales,order=c(0,0,4),xreg=data.frame(price),
+                 fixed=c(0,NA,0,NA,NA,NA)); chip.m4
```

Note that the regression coefficient estimate on Price is similar to that from the OLS regression fit earlier, but the standard error of the estimate is about 10% lower than that from the simple OLS regression. This illustrates the general result that the simple OLS estimator is consistent but the associated standard error is generally not trustworthy.

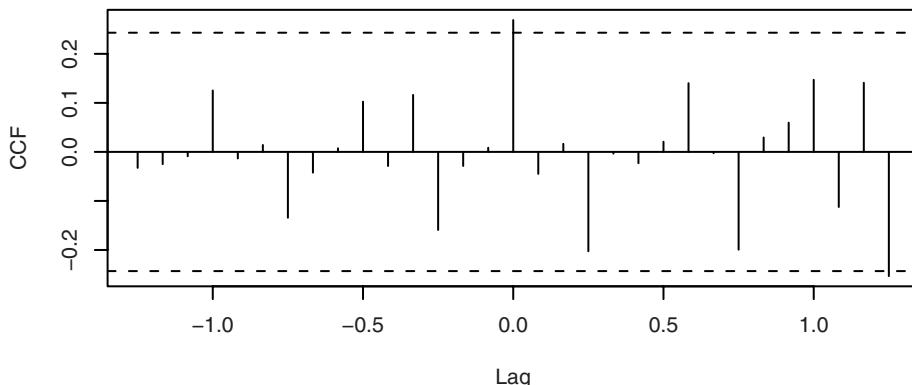
The residuals from this fitted model by and large pass various model diagnostic tests except that the residual ACF is significant at lag 14. As a result, some Box-Ljung test statistics have p -values bordering on 0.05 when 14 or more lags of the residual auto-correlations are included in the test. Even though the significant ACF at lag 14 may suggest a quarterly effect, we do not report a more complex model including lag 14 because (1) 14 weeks do not exactly make a quarter and (2) adding a seasonal MA(1) component of period 14 only results in marginal improvement in terms of model diagnostics.

For a second example, we study the impact of higher gasoline price on public transportation usage. The dataset consists of the monthly number of boardings on public transportation in the Denver, Colorado, region together with the average monthly gasoline prices in Denver from August 2000 through March 2006. Both variables are skewed to the right and hence are log-transformed. As we shall see below, the logarithmic transformation also makes the final fitted model more interpretable. The time series plots, shown in Exhibit 11.24, display the increasing trends for both variables and the seasonal fluctuation in the number of boardings. Based on the sample ACF and PACF, an ARIMA(2,1,0) model was fitted to the gasoline price data. This fitted model was then used to filter the boardings data before computing their sample CCF which is shown in Exhibit 11.25. The sample CCF is significant at lags 0 and 15, suggesting positive contemporaneous correlation between gasoline price and public transportation usage. The significant CCF at lag 15, however, is unlikely to be real, as it is hard to imagine why the number of boardings might lead the gasoline price with a lag of 15 months. In this case, the quick preliminary approach of prewhitening the series by fitting a long AR model, however, showed that none of the CCFs are significant. It turns out that even after differencing the data, the AIC selects an AR(16) model. The higher order selected coupled with the relatively short time span may substantially weaken the power to detect correlations between the two variables. Incidentally, this example warns against simply relying on the AIC to select a high-order AR model to do prewhitening, especially with relatively short time series data.

Exhibit 11.24 Logarithms of Monthly Public Transit Boardings and Gasoline Prices in Denver, August 2000 through March 2006



```
> data(boardings)
> plot(boardings, yax.flip=T)
```

Exhibit 11.25 Sample CCF of Prewhitened Log(Boardings) and Log(Price)

```
> m1=arima(boardings[,2],order=c(2,1,0))
> prewhiten(x=boardings[,2],y=boardings[,1],x.model=m1)
```

Based on the sample ACF, PACF, and EACF of the residuals from a linear model of boardings on gasoline price, a seasonal ARIMA(2,0,0) \times (1,0,0)₁₂ model was tentatively specified for the error process in the regression model. However, the ϕ_2 coefficient estimate was not significant, and hence the AR order was reduced to $p = 1$. Using the outlier detection techniques discussed in Section 11.2, we found an additive outlier for March 2003 and an innovative outlier for March 2004. Because the test statistic for the additive outlier had a larger magnitude than that of the innovative outlier (-4.09 vs. 3.65), we incorporated the additive outlier in the model.[†] Diagnostics of the subsequent fitted model reveals that the residual ACF was significant at lag 3, which suggests the error process is a seasonal ARIMA(1,0,3) \times (1,0,0)₁₂ + outlier process. As the estimates of the coefficients θ_1 and θ_2 were found to be insignificant, they were suppressed from the final fitted model that is reported in Exhibit 11.26.

Diagnostics of the final fitted model suggest a good fit to the data. Also, no further outliers were detected. A 95% confidence interval for the regression coefficient on Log(Price) is (0.0249, 0.139). Note the interpretation of the fitted model: a 100% increase in the price of gasoline will lead to about an 8.2% increase in public transportation usage.

[†] Subsequent investigation revealed that a 30 inch snowstorm in March 2003 completely shut down Denver for one full day. It remained partially shut down for a few more days.

Exhibit 11.26 Maximum Likelihood Estimates of the Regression Model of Log(Boardings) on Log(Price) with ARMA Errors

Parameter	ϕ_1	θ_3	Φ_1	Intercept	Log(Price)	Outlier
Estimate	0.8782	0.3836	0.8987	12.12	0.0819	-0.0643
Standard Error	0.0645	0.1475	0.0395	0.1638	0.0291	0.0109

σ^2 estimated as 0.0004094: log-likelihood = 158.02, AIC = -304.05

```

> log.boardings=boardings[,1]
> log.price=boardings[,2]
> boardings.m1=arima(log.boardings,order=c(1,0,0),
+   seasonal=list(order=c(1,0,0),period=12),
+   xreg=data.frame(log.price))
> boardings.m1
> detectAO(boardings.m1); detectIO(boardings.m1)
> boardings.m2=arima(log.boardings,order=c(1,0,3),
+   seasonal=list(order=c(1,0,0),period=12),
+   xreg=data.frame(log.price,outlier=c(rep(0,31),1,rep(0,36))),
+   fixed=c(NA,0,0,rep(NA,5)))
> boardings.m2
> detectAO(boardings.m2); detectIO(boardings.m2)
> tsdiag(boardings.m2,tol=.15,gof.lag=24)

```

It is also of interest to note that dropping the outlier term from the model results in a new regression estimate on Log(Price) of 0.0619 with a standard error of 0.0372. Thus, when the outlier is not properly modeled, the regression coefficient ceases to be significant at the 5% level. As demonstrated by this example, the presence of an outlier can adversely affect inference in time series modeling.

11.5 Summary

In this chapter, we used information from other events or other time series to help model the time series of main interest. We began with the so-called intervention models, which attempt to incorporate known external events that we believe have a significant effect on the time series of interest. Various simple but useful ways of modeling the effects of these interventions were discussed. Outliers are observations that deviate rather substantially from the general pattern of the data. Models were developed to detect and incorporate outliers in time series. The material in the section on spurious correlation illustrates how difficult it is to assess relationships between two time series, but methods involving prewhitening were shown to help in this regard. Several substantial examples were used to illustrate the methods and techniques discussed.

EXERCISES

- 11.1** Produce a time series plot of the air passenger miles over the period January 1996 through May 2005 using seasonal plotting symbols. Display the graph full-screen and discuss the seasonality that is displayed. The data are in the file named airmiles.
- 11.2** Show that the expression given for m_t in Equation (11.1.7) on page 251 satisfies the “AR(1)” recursion given in Equation (11.1.6) with the initial condition $m_0 = 0$.
- 11.3** Find the “half-life” for the intervention effect specified in Equation (11.1.6) on page 251 when $\delta = 0.7$.
- 11.4** Show that the “half-life” for the intervention effect specified in Equation (11.1.6) on page 251 increases without bound as δ increases to 1.
- 11.5** Show that for the intervention effect specified by Equation (11.1.6) on page 251

$$\lim_{\delta \rightarrow 1} m_t = \begin{cases} \omega(T-t), & \text{for } t \geq T \\ 0, & \text{otherwise} \end{cases}$$

- 11.6** Consider the intervention effect displayed in Exhibit 11.3, (b), page 253.
- (a) Show that the jump at time $T+1$ is of height ω as displayed.
 - (b) Show that, as displayed, the intervention effect tends to $\omega/(1-\delta)$ as t increases without bound.
- 11.7** Consider the intervention effect displayed in Exhibit 11.3, (c), page 253. Show that the effect increases linearly starting at time $T+1$ with slope ω as displayed.
- 11.8** Consider the intervention effect displayed in Exhibit 11.4, (a), page 254.
- (a) Show that the jump at time $T+1$ is of height ω as displayed.
 - (b) Show that, as displayed, the intervention effect tends to go back to 0 as t increases without bound.
- 11.9** Consider the intervention effect displayed in Exhibit 11.4, (b), page 254.
- (a) Show that the jump at time $T+1$ is of height $\omega_1 + \omega_2$ as displayed.
 - (b) Show that, as displayed, the intervention effect tends to ω_2 as t increases without bound.
- 11.10** Consider the intervention effect displayed in Exhibit 11.4, (c), page 254.
- (a) Show that the jump at time T is of height ω_0 as displayed.
 - (a) Show that the jump at time $T+1$ is of height $\omega_1 + \omega_2$ as displayed.
 - (b) Show that, as displayed, the intervention effect tends to ω_2 as t increases without bound.
- 11.11** Simulate 100 pairs of (X_t, Y_t) from the model of Equation (11.3.1) on page 261 with $d = 3$, $\beta_0 = 0$, and $\beta_1 = 1$. Use $\sigma_X = 2$ and $\sigma_e = 1$. Display and interpret the sample CCF between these two series.
- 11.12** Show that when the X and Y are independent AR(1) time series with parameters ϕ_X and ϕ_Y respectively, Equation (11.3.5) on page 262 reduces to give Equation (11.3.6).
- 11.13** Show that for the process defined by Equation (11.4.5) on page 266, the cross-correlation between \tilde{X} and \tilde{Y} at lag k is given by $\beta_{-k}(\sigma_{\tilde{X}}/\sigma_{\tilde{Y}})$.

11.14 Simulate an AR time series with $\phi = 0.7$, $\mu = 0$, $\sigma_e = 1$, and of length $n = 48$. Plot the time series, and inspect the sample ACF and PACF of the series.

(a) Now add a step function response of $\omega = 1$ unit height at time $t = 36$ to the simulated series. The series now has a theoretical mean of zero from $t = 1$ to 35 and a mean of 1 from $t = 36$ on. Plot the new time series and calculate the sample ACF and PACF for the new series. Compare these with the results for the original series.

(b) Repeat part (a) but with an impulse response at time $t = 36$ of unit height, $\omega = 1$. Plot the new time series, and calculate the sample ACF and PACF for the new series. Compare these with the results for the original series. See if you can detect the additive outlier at time $t = 36$ assuming that you do not know where the outlier might occur.

11.15 Consider the air passenger miles time series discussed in this chapter. The file is named airmiles. Use only the *preintervention* data (that is, data prior to September 2001) for this exercise.

(a) Verify that the sample ACF for the twice differenced series of the logarithms of the preintervention data is as shown in Exhibit 11.5 on page 255.

(b) The plot created in part (a) suggests an ARIMA(0,1,1) \times (0,1,0)₁₂. Fit this model and assess its adequacy. In particular, verify that additive outliers are detected in December 1996, January 1997, and December 2002.

(c) Now fit an ARIMA(0,1,1) \times (0,1,0)₁₂ + three outliers model and assess its adequacy.

(d) Finally, fit an ARIMA(0,1,1) \times (0,1,1)₁₂ + three outliers model and assess its adequacy.

11.16 Use the logarithms of the Denver region public transportation boardings and Denver gasoline price series. The data are in the file named boardings.

(a) Display the time series plot of the monthly boardings using seasonal plotting symbols. Interpret the plot.

(b) Display the time series plot of the monthly average gasoline prices using seasonal plotting symbols. Interpret the plot.

11.17 The data file named deere1 contains 82 consecutive values for the amount of deviation (in 0.000025 inch units) from a specified target value that an industrial machining process at Deere & Co. produced under certain specified operating conditions. These data were first used in Exercise 6.33, page 146, where we observed an obvious outlier at time $t = 27$.

(a) Fit an AR(2) model using the original data including the outlier.

(b) Test the fitted AR(2) model of part (a) for both AO and IO outliers.

(c) Now fit the AR(2) model incorporating a term in the model for the outlier.

(d) Assess the fit of the model in part (c) using all of our diagnostic tools. In particular, compare the properties of this model with the one obtained in part (a).

- 11.18** The data file named `days` contains accounting data from the Winegard Co. of Burlington, Iowa. The data are the number of days until Winegard receives payment for 130 consecutive orders from a particular distributor of Winegard products. (The name of the distributor must remain anonymous for confidentiality reasons.) These data were first investigated in Exercise 6.39, page 147, but several outliers were observed. When the observed outliers were replaced by more typical values, an MA(2) model was suggested.
- (a) Fit an MA(2) model to the original data, and test the fitted model for both AO and IO outliers.
 - (b) Now fit the MA(2) model incorporating the outliers into the model.
 - (c) Assess the fit of the model obtained in part (b). In particular, are any more outliers indicated?
 - (d) Fit another MA(2) model incorporating any additional outliers found in part (c), and assess the fit of this model.
- 11.19** The data file named `bluebirdlite` contains weekly sales and price data for Bluebird Lite potato chips. Carry out an analysis similar to that for Bluebird Standard potato chips that was begun on page 267.
- 11.20** The file named `units` contains annual unit sales of a certain product from a widely known international company over the years 1983 through 2005. (The name of the company must remain anonymous for proprietary reasons.)
- (a) Plot the time series of units and describe the general features of the plot.
 - (b) Use ordinary least squares regression to fit a straight line in time to the series.
 - (c) Display the sample PACF of the residuals from this model, and specify an ARIMA model for the residuals.
 - (d) Now fit the model $\text{unit sales} = \text{AR}(2) + \text{time}$. Interpret the output. In particular, compare the estimated regression coefficient on the time variable obtained here with the one you obtained in part (b).
 - (e) Perform a thorough analysis of the residuals from this last model.
 - (f) Repeat parts (d) and (e) using the logarithms of unit sales as the response variable. Compare these results with those obtained in parts (d) and (e).
- 11.21** In Chapters 5–8, we investigated an IMA(1,1) model for the logarithms of monthly oil prices. Exhibit 8.3 on page 178 suggested that there may be several outliers in this series. Investigate the IMA(1,1) model for this series for outliers using the techniques developed in this chapter. Be sure to compare your results with those obtained earlier that ignored the outliers. The data are in the file named `oil`.

CHAPTER 12

TIME SERIES MODELS OF HETROSCEDASTICITY

The models discussed so far concern the conditional mean structure of time series data. However, more recently, there has been much work on modeling the conditional variance structure of time series data—mainly motivated by the needs for financial modeling. Let $\{Y_t\}$ be a time series of interest. The conditional variance of Y_t given the past Y values, Y_{t-1}, Y_{t-2}, \dots , measures the uncertainty in the deviation of Y_t from its conditional mean $E(Y_t|Y_{t-1}, Y_{t-2}, \dots)$. If $\{Y_t\}$ follows some ARIMA model, the (one-step-ahead) conditional variance is always equal to the noise variance for any present and past values of the process. Indeed, the constancy of the conditional variance is true for predictions of any fixed number of steps ahead for an ARIMA process. In practice, the (one-step-ahead) conditional variance may vary with the current and past values of the process, and, as such, the conditional variance is itself a random process, often referred to as the conditional variance process. For example, daily returns of stocks are often observed to have larger conditional variance following a period of violent price movement than a relatively stable period. The development of models for the conditional variance process with which we can predict the variability of future values based on current and past data is the main concern of the present chapter. In contrast, the ARIMA models studied in earlier chapters focus on how to predict the conditional mean of future values based on current and past data.

In finance, the conditional variance of the return of a financial asset is often adopted as a measure of the risk of the asset. This is a key component in the mathematical theory of pricing a financial asset and the *VaR* (Value at Risk) calculations; see, for example, Tsay (2005). In an efficient market, the expected return (conditional mean) should be zero, and hence the return series should be white noise. Such series have the simplest autocorrelation structure. Thus, for ease of exposition, we shall assume in the first few sections of this chapter that the data are returns of some financial asset and are white noise; that is, serially uncorrelated data. By doing so, we can concentrate initially on studying how to model the conditional variance structure of a time series. By the end of the chapter, we discuss some simple schemes for simultaneously modeling the conditional mean and conditional variance structure by combining an ARIMA model with a model of conditional heteroscedasticity.

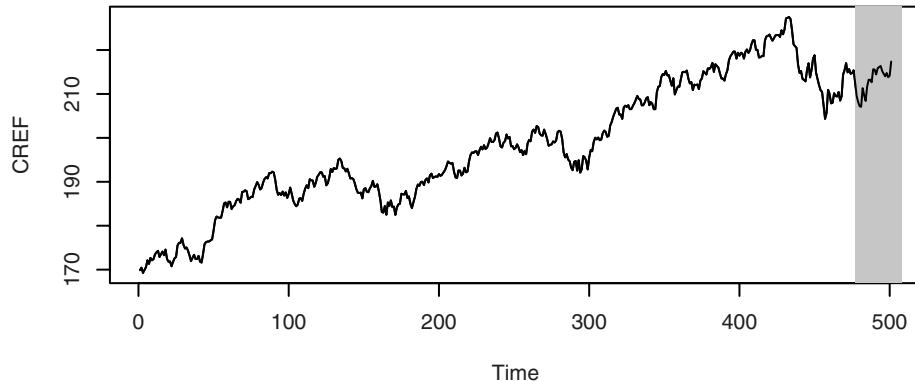
12.1 Some Common Features of Financial Time Series

As an example of financial time series, we consider the daily values of a unit of the CREF stock fund over the period from August 26, 2004 to August 15, 2006. The CREF stock fund is a fund of several thousand stocks and is not openly traded in the stock market.[†] Since stocks are not traded over weekends or on holidays, only on so-called *trading days*, the CREF data do not change over weekends and holidays. For simplicity, we will analyze the data as if they were equally spaced. Exhibit 12.1 shows the time series plot of the CREF data. It shows a generally increasing trend with a hint of higher variability with higher level of the stock value. Let $\{p_t\}$ be the time series of, say, the daily price of some financial asset. The (continuously compounded) return on the t th day is defined as

$$r_t = \log(p_t) - \log(p_{t-1}) \quad (12.1.1)$$

Sometimes the returns are then multiplied by 100 so that they can be interpreted as percentage changes in the price. The multiplication may also reduce numerical errors as the raw returns could be very small numbers and render large rounding errors in some calculations.

Exhibit 12.1 Daily CREF Stock Values: August 26, 2004 to August 15, 2006



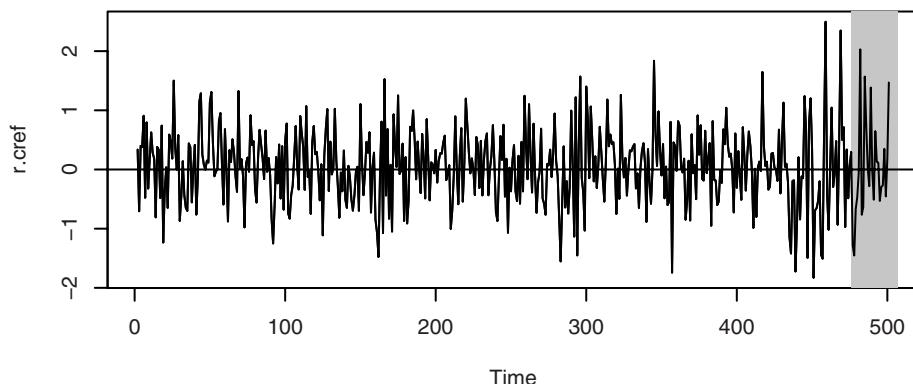
```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(CREF); plot(CREF)
```

Exhibit 12.2 plots the CREF return series (sample size = 500). The plot shows that the returns were more volatile over some time periods and became very volatile toward the end of the study period. This observation may be more clearly seen by plotting the time sequence plot of the absolute or squared returns; see Exercise 12.1, page 316.

[†] CREF stands for College Retirement Equities Fund—a group of stock and bond funds crucial to many college faculty retirement plans.

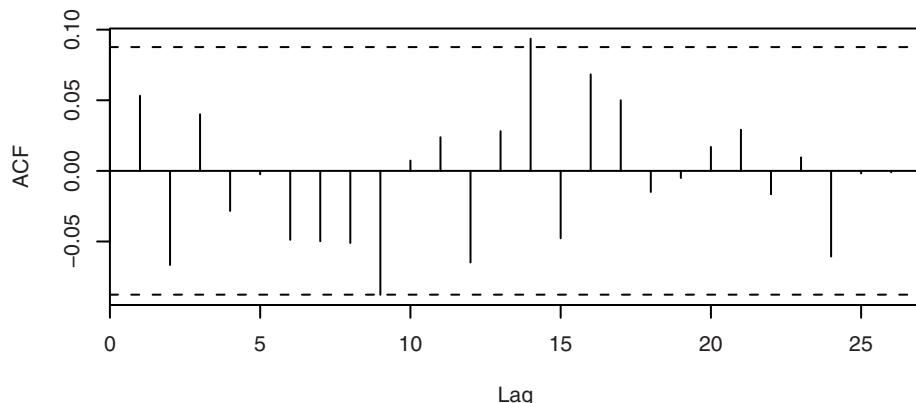
These results might be triggered by the instability in the Middle East due to a war in southern Lebanon from July 12 to August 14, 2006, the period that is shaded in gray in Exhibits 12.1 and 12.2. This pattern of alternating quiet and volatile periods of substantial duration is referred to as *volatility clustering* in the literature. Volatility in a time series refers to the phenomenon where the conditional variance of the time series varies over time. The study of the dynamical pattern in the volatility of a time series (that is, the conditional variance process of the time series) constitutes the main subject of this chapter.

Exhibit 12.2 Daily CREF Stock Returns: August 26, 2004 to August 15, 2006

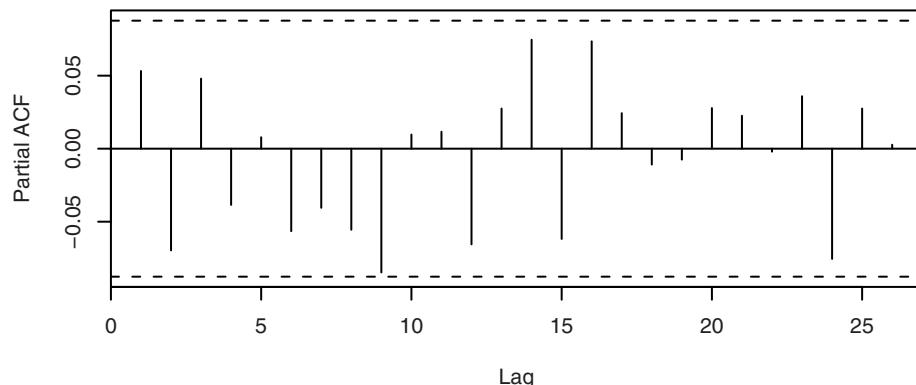


```
> r.cref=diff(log(CREF))*100  
> plot(r.cref); abline(h=0)
```

The sample ACF and PACF of the daily CREF returns (multiplied by 100), shown in Exhibits 12.3 and 12.4, suggest that the returns have little serial correlation at all. The sample EACF (not shown) also suggests that a white noise model is appropriate for these data. The average CREF return equals 0.0493 with a standard error of 0.02885. Thus the mean of the return process is not statistically significantly different from zero. This is expected based on the efficient-market hypothesis alluded to in the introduction to this chapter.

Exhibit 12.3 Sample ACF of Daily CREF Returns: 8/26/04 to 8/15/06

```
> acf(r.cref)
```

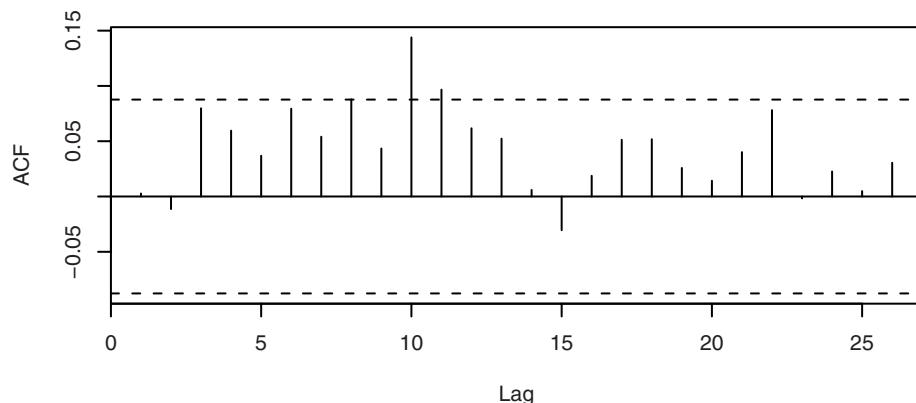
Exhibit 12.4 Sample PACF of Daily CREF Returns: 8/26/04 to 8/15/06

```
> pacf(r.cref)
```

However, the volatility clustering observed in the CREF return data gives us a hint that they may not be independently and identically distributed—otherwise the variance would be constant over time. This is the first occasion in our study of time series models where we need to distinguish between series values being uncorrelated and series values being independent. If series values are truly independent, then nonlinear instantaneous transformations such as taking logarithms, absolute values, or squaring preserves independence. However, the same is not true of correlation, as correlation is only a measure of linear dependence. Higher-order serial dependence structure in data can be explored by studying the autocorrelation structure of the absolute returns (of lesser sampling vari-

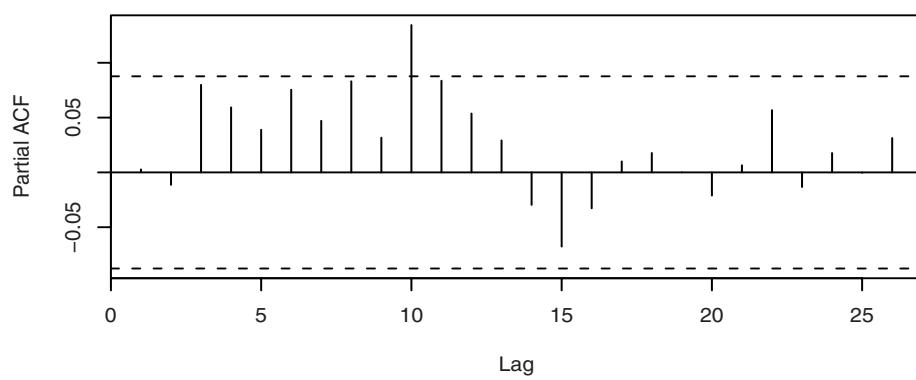
ability with less mathematical tractability) or that of the squared returns (of greater sampling variability but with more manageability in terms of statistical theory). If the returns are independently and identically distributed, then so are the absolute returns (as are the squared returns), and hence they will be white noise as well. Hence, if the absolute or squared returns admit some significant autocorrelations, then these autocorrelations furnish some evidence against the hypothesis that the returns are independently and identically distributed. Indeed, the sample ACF and PACF of the absolute returns and those of the squared returns in Exhibits 12.5 through 12.8 display some significant autocorrelations and hence provide some evidence that the daily CREF returns are not independently and identically distributed.

Exhibit 12.5 Sample ACF of the Absolute Daily CREF Returns

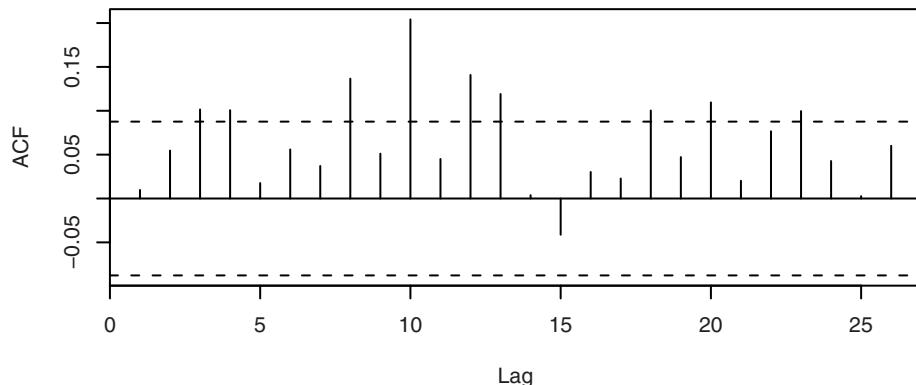


```
> acf(abs(r.cref))
```

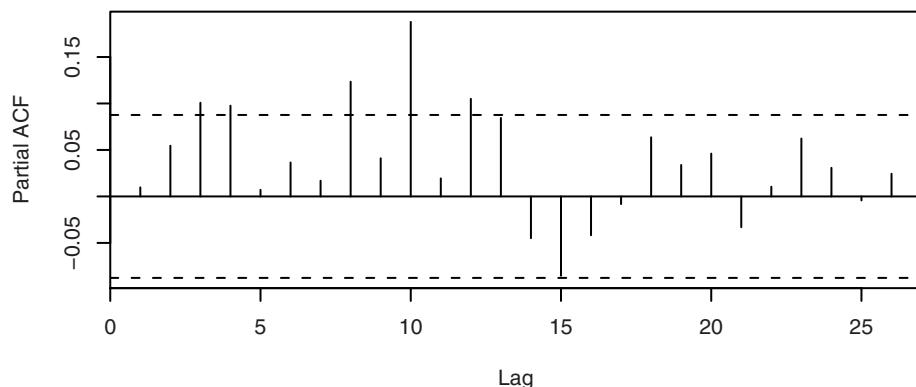
Exhibit 12.6 Sample PACF of the Absolute Daily CREF Returns



```
> pacf(abs(r.cref))
```

Exhibit 12.7 Sample ACF of the Squared Daily CREF Returns

```
> acf(r.cref^2)
```

Exhibit 12.8 Sample PACF of the Squared Daily CREF Returns

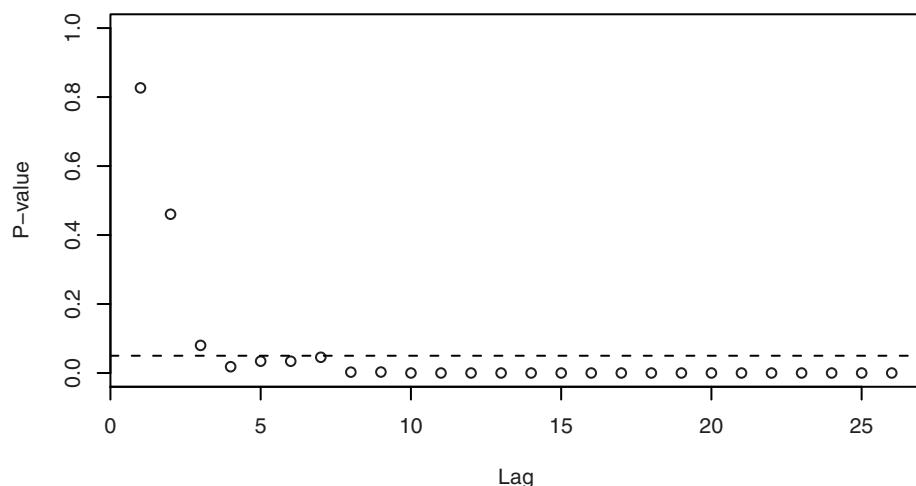
```
> pacf(r.cref^2)
```

These visual tools are often supplemented by formally testing whether the squared data are autocorrelated using the Box-Ljung test. Because no model fitting is required, the degrees of freedom of the approximating chi-square distribution for the Box-Ljung statistic equals the number of correlations used in the test. Hence, if we use m autocorrelations of the squared data in the test, the test statistic is approximately chi-square distributed with m degrees of freedom, if there is no ARCH. This approach can be extended to the case when the conditional mean of the process is non-zero and if an ARMA model is adequate in describing the autocorrelation structure of the data. In which case, the first m autocorrelations of the squared residuals from this model can be used to test for the presence of ARCH. The corresponding Box-Ljung statistic will have a

chi-square distribution with m degrees of freedom under the assumption of no ARCH effect, see McLeod and Li (1983) and Li(2004). Below, we shall refer to the test for ARCH effects using the Box-Ljung statistic with the squared residuals or data as the McLeod- Li test.

In practice, it is useful to apply the McLeod-Li test for ARCH using a number of lags and plot the p -values of the test. Exhibit 12.9 shows that the McLeod-Li tests are all significant at the 5% significance level when more than 3 lags are included in the test. This is broadly consistent with the visual pattern in Exhibit 12.7 and formally shows strong evidence for ARCH in this data.

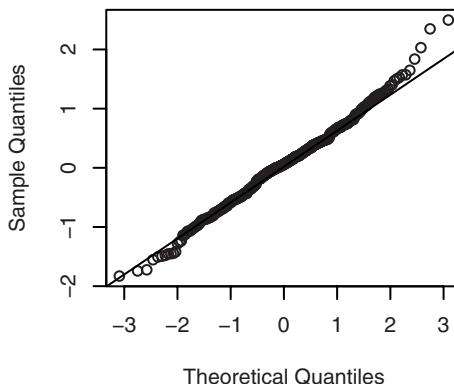
Exhibit 12.9 McLeod-Li Test Statistics for Daily CREF Returns



```
> win.graph(width=4.875, height=3, pointsize=8)
> McLeod.Li.test(y=r.cref)
```

The distributional shape of the CREF returns can be explored by constructing a QQ normal scores plot—see Exhibit 12.10. The QQ plot suggests that the distribution of returns may have a tail thicker than that of a normal distribution and may be somewhat skewed to the right. Indeed, the Shapiro-Wilk test statistic for testing normality equals 0.9932 with p -value equal to 0.024, and hence we reject the normality hypothesis at the usual significance levels.

Exhibit 12.10 QQ Normal Plot of Daily CREF Returns



```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(r.cref); qqline(r.cref)
```

The *skewness* of a random variable, say Y , is defined by $E(Y-\mu)^3/\sigma^3$, where μ and σ are the mean and standard deviation of Y , respectively. It can be estimated by the sample skewness

$$g_1 = \sum_{i=1}^n (Y_i - \bar{Y})^3 / (n\hat{\sigma}^3) \quad (12.1.2)$$

where $\hat{\sigma}^2 = \Sigma(Y_i - \bar{Y})^2/n$ is the sample variance. The sample skewness of the CREF returns equals 0.116. The thickness of the tail of a distribution relative to that of a normal distribution is often measured by the (excess) *kurtosis*, defined as $E(Y - \mu)^4/\sigma^4 - 3$. For normal distributions, the kurtosis is always equal to zero. A distribution with positive kurtosis is called a *heavy-tailed* distribution, whereas it is called *light-tailed* if its kurtosis is negative. The kurtosis can be estimated by the sample kurtosis

$$g_2 = \sum_{i=1}^n (Y_i - \bar{Y})^4 / (n\hat{\sigma}^4) - 3 \quad (12.1.3)$$

The sample kurtosis of the CREF returns equals 0.6274. An alternative definition of kurtosis modifies the formula and uses $E(r_t - \mu)^4/\sigma^4$; that is, it does not subtract three from the ratio. We shall always use the former definition for kurtosis.

Another test for normality is the Jarque-Bera test, which is based on the fact that a normal distribution has zero skewness and zero kurtosis. Assuming independently and identically distributed data Y_1, Y_2, \dots, Y_n , the Jarque-Bera test statistic is defined as

$$JB = \frac{n g_1^2}{6} + \frac{n g_2^2}{24} \quad (12.1.4)$$

where g_1 is the sample skewness and g_2 is the sample kurtosis. Under the null hypothesis of normality, the Jarque-Bera test statistic is approximately distributed as χ^2 with two degrees of freedom. In fact, under the normality assumption, each summand defining the Jarque-Bera statistic is approximately χ^2 with 1 degree of freedom. The Jarque-Bera test rejects the normality assumption if the test statistic is too large. For the CREF returns, $JB = 500 \times 0.116^2/6 + 500 \times 0.6274^2/24 = 1.12 + 8.20 = 9.32$ with a p -value equal to 0.011. Recall that the upper 5 percentage point of a χ^2 distribution with unit degree of freedom equals 3.84. Hence, the data appear not to be skewed but do have a relatively heavy tail. In particular, the normality assumption is inconsistent with the CREF return data—a conclusion that is also consistent with the finding of the Shapiro-Wilk test.

In summary, the CREF return data are found to be serially uncorrelated but admit a higher-order dependence structure, namely volatility clustering, and a heavy-tailed distribution. It is commonly observed that such characteristics are rather prevalent among financial time series data. The GARCH models introduced in the next sections attempt to provide a framework for modeling and analyzing time series that display some of these characteristics.

12.2 The ARCH(1) Model

Engle (1982) first proposed the autoregressive conditional heteroscedasticity (ARCH) model for modeling the changing variance of a time series. As discussed in the previous section, the return series of a financial asset, say $\{r_t\}$, is often a serially uncorrelated sequence with zero mean, even as it exhibits volatility clustering. This suggests that the conditional variance of r_t given past returns is not constant. The conditional variance, also referred to as the *conditional volatility*, of r_t will be denoted by $\sigma_{t|t-1}^2$, with the subscript $t-1$ signifying that the conditioning is upon returns through time $t-1$. When r_t is available, the squared return r_t^2 provides an unbiased estimator of $\sigma_{t|t-1}^2$. A series of large squared returns may foretell a relatively volatile period. Conversely, a series of small squared returns may foretell a relatively quiet period. The ARCH model is formally a regression model with the conditional volatility as the response variable and the past lags of the squared return as the covariates. For example, the ARCH(1) model assumes that the return series $\{r_t\}$ is generated as follows:

$$r_t = \sigma_{t|t-1} \varepsilon_t \quad (12.2.1)$$

$$\sigma_{t|t-1}^2 = \omega + \alpha r_{t-1}^2 \quad (12.2.2)$$

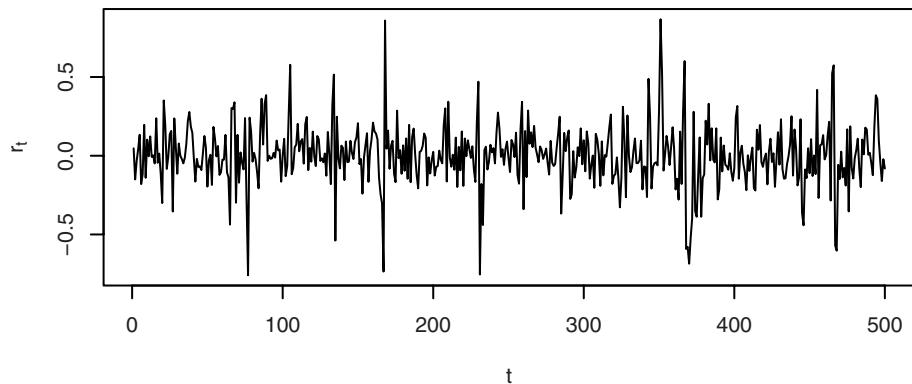
where α and ω are unknown parameters, $\{\varepsilon_t\}$ is a sequence of *independently and identically distributed* random variables each with zero mean and unit variance (also known as the *innovations*), and ε_t is independent of r_{t-j} , $j = 1, 2, \dots$. The innovation ε_t is presumed to have unit variance so that the conditional variance of r_t equals $\sigma_{t|t-1}^2$. This follows from

$$\begin{aligned}
 E(r_t^2 | r_{t-j}, j = 1, 2, \dots) &= E(\sigma_{t|t-1}^2 \varepsilon_t^2 | r_{t-j}, j = 1, 2, \dots) \\
 &= \sigma_{t|t-1}^2 E(\varepsilon_t^2 | r_{t-j}, j = 1, 2, \dots) \\
 &= \sigma_{t|t-1}^2 E(\varepsilon_t^2) \\
 &= \sigma_{t|t-1}^2
 \end{aligned} \tag{12.2.3}$$

The second equality follows because $\sigma_{t|t-1}$ is known given the past returns, the third equality holds because ε_t is independent of past returns, and the last equality results from the assumption that the variance of ε_t equals 1.

Exhibit 12.11 shows the time series plot of a simulated series of size 500 from an ARCH(1) model with $\omega = 0.01$ and $\alpha = 0.9$. Volatility clustering is evident in the data as larger fluctuations cluster together, although the series is able to recover from large fluctuations quickly because of the very short memory in the conditional variance process.[†]

Exhibit 12.11 Simulated ARCH(1) Model with $\omega = 0.01$ and $\alpha_1 = 0.9$



```

> set.seed(1235678); library(tseries)
> garch01.sim=garch.sim(alpha=c(.01,.9),n=500)
> plot(garch01.sim,type='l',ylab=expression(r[t]), xlab='t')

```

While the ARCH model resembles a regression model, the fact that the conditional variance is not directly observable (and hence is called a latent variable) introduces some subtlety in the use of ARCH models in data analysis. For example, it is not obvious how to explore the regression relationship graphically. To do so, it is pertinent to replace the conditional variance by some observable in Equation (12.2.2). Let

[†] The R package named `tseries` is required for this chapter. We assume that the reader has downloaded and installed it.

$$\eta_t = r_t^2 - \sigma_{t|t-1}^2 \quad (12.2.4)$$

It can be verified that $\{\eta_t\}$ is a serially uncorrelated series with zero mean. Moreover, η_t is uncorrelated with past returns. Substituting $\sigma_{t|t-1}^2 = r_t^2 - \eta_t$ into Equation (12.2.2) it is obvious that

$$r_t^2 = \omega + \alpha r_{t-1}^2 + \eta_t \quad (12.2.5)$$

Thus, the squared return series satisfies an AR(1) model under the assumption of an ARCH(1) model for the return series! Based on this useful observation, an ARCH(1) model may be specified if an AR(1) specification for the squared returns is warranted by techniques learned from earlier chapters.

Besides its value in terms of data analysis, the deduced AR(1) model for the squared returns can be exploited to gain theoretical insights on the parameterization of the ARCH model. For example, because the squared returns must be nonnegative, it makes sense to always restrict the parameters ω and α to be nonnegative. Also, if the return series is stationary with variance σ^2 , then taking expectation on both sides of Equation (12.2.5) yields

$$\sigma^2 = \omega + \alpha \sigma^2 \quad (12.2.6)$$

That is, $\sigma^2 = \omega/(1 - \alpha)$ and hence $0 \leq \alpha < 1$. Indeed, it can be shown (Ling and McAleer, 2002) that the condition $0 \leq \alpha < 1$ is necessary and sufficient for the (weak) stationarity of the ARCH(1) model. At first sight, it seems that the concepts of stationarity and conditional heteroscedasticity may be incompatible. However, recall that weak stationarity of a process requires that the mean of the process be constant and the covariance of the process at any two epochs be finite and identical whenever the lags of the two epochs are the same. In particular, the variance is constant for a weakly stationary process. The condition $0 \leq \alpha < 1$ implies that there exists an initial distribution for r_0 such that r_t defined by Equations (12.2.1) and (12.2.2) for $t \geq 1$ is weakly stationary in the sense above. It is interesting to observe that weak stationarity does not preclude the possibility of a nonconstant conditional variance process, as is the case for the ARCH(1) model! It can be checked that the ARCH(1) process is white noise. Hence, it is an example of a white noise that admits a nonconstant conditional variance process as defined by Equation (12.2.2) that varies with the lag one of the squared process.

A satisfying feature of the ARCH(1) model is that, even if the innovation η_t has a normal distribution, the stationary distribution of an ARCH(1) model with $1 > \alpha > 0$ has fat tails; that is, its kurtosis, $E(r_t^4)/\sigma^4 - 3$, is greater than zero. (Recall that the kurtosis of a normal distribution is always equal to 0, and a distribution with positive kurtosis is said to be fat-tailed, while one with a negative kurtosis is called a light-tailed distribution.) To see the validity of this claim, consider the case where the $\{\varepsilon_t\}$ are independently and identically distributed as standard normal variables. Raising both sides of Equation (12.2.1) on page 285 to the fourth power and taking expectations gives

$$\begin{aligned}
E(r_t^4) &= E[E(\sigma_{t|t-1}^4 \varepsilon_t^4 | r_{t-j}, j = 1, 2, 3, \dots)] \\
&= E[\sigma_{t|t-1}^4 E(\varepsilon_t^4 | r_{t-j}, j = 1, 2, 3, \dots)] \\
&= E[\sigma_{t|t-1}^4 E(\varepsilon_t^4)] \\
&= 3E(\sigma_{t|t-1}^4)
\end{aligned} \tag{12.2.7}$$

The first equality follows from the iterated-expectation formula, which, in the simple case of two random variables X, Y , states that $E[E(X|Y)] = E(X)$. [See Equation (9.E.5) on page 218 for a review.] The second equality results from the fact that $\sigma_{t|t-1}$ is known given past returns. The third equality is a result of the independence between ε_t and past returns, and the final equality follows from the normality assumption. It remains to calculate $E(\sigma_{t|t-1}^4)$. Now, it is unclear whether the preceding expectation exists as a finite number. For the moment, assume it does and, assuming stationarity, let it be denoted by τ . Below, we shall derive a condition for this assumption to be valid. Raising both sides of Equation (12.2.2) to the second power and taking expectation yields

$$\tau = \omega^2 + 2\omega\alpha\sigma^2 + \alpha^2 3\tau \tag{12.2.8}$$

which implies

$$\tau = \frac{\omega^2 + 2\omega\alpha\sigma^2}{1 - 3\alpha^2} \tag{12.2.9}$$

This equality shows that a necessary (and, in fact, also sufficient) condition for the finiteness of τ is that $0 \leq \alpha < 1/\sqrt{3}$, in which case the ARCH(1) process has finite fourth moment. Incidentally, this shows that a stationary ARCH(1) model need not have finite fourth moments. The existence of finite higher moments will further restrict the parameter range—a feature also shared by higher-order analogues of the ARCH model and its variants. Returning to the calculation of the kurtosis of an ARCH(1) process, it can be shown by tedious algebra that Equation (12.2.1) implies that $\tau > \sigma^4$ and hence $E(r_t^4) > 3\sigma^4$. Thus the kurtosis of a stationary ARCH(1) process is greater than zero. This verifies our earlier statement that an ARCH(1) process has fat tails even with normal innovations. In other words, the fat tail is a result of the volatility clustering as specified by Equation (12.2.2).

A main use of the ARCH model is to predict the future conditional variances. For example, one might be interested in forecasting the h -step-ahead conditional variance

$$\sigma_{t+h|t}^2 = E(r_{t+h}^2 | r_t, r_{t-1}, \dots) \tag{12.2.10}$$

For $h = 1$, the ARCH(1) model implies that

$$\sigma_{t+1|t}^2 = \omega + \alpha r_t^2 = (1 - \alpha)\sigma^2 + \alpha r_t^2 \tag{12.2.11}$$

which is a weighted average of the long-run variance and the current squared return. Similarly, using the iterated expectation formula, we have

$$\begin{aligned}
\sigma_{t+h|t}^2 &= E(r_{t+h}^2 | r_t, r_{t-1}, \dots) \\
&= E[E(\sigma_{t+h|t+h-1}^2 \varepsilon_{t+h}^2 | r_{t+h-1}, r_{t+h-2}, \dots) | r_t, r_{t-1}, \dots] \\
&= E[\sigma_{t+h|t+h-1}^2 E(\varepsilon_{t+h}^2) | r_t, r_{t-1}, \dots] \\
&= E(\sigma_{t+h|t+h-1}^2 | r_t, r_{t-1}, \dots) \\
&= \omega + \alpha E(r_{t+h-1}^2 | r_t, r_{t-1}, \dots) \\
&= \omega + \alpha \sigma_{t+h-1|t}^2
\end{aligned} \tag{12.2.12}$$

where we adopt the convention that $\sigma_{t+h|t}^2 = r_{t+h}^2$ for $h < 0$. The formula above provides a recursive recipe for computing the h -step-ahead conditional variance.

12.3 GARCH Models

The forecasting formulas derived in the previous section show both the strengths and weaknesses of an ARCH(1) model, as the forecasting of the future conditional variances only involves the most recent squared return. In practice, one may expect that the accuracy of forecasting may improve by including all past squared returns with lesser weight for more distant volatilities. One approach is to include further lagged squared returns in the model. The ARCH(q) model, proposed by Engle (1982), generalizes Equation (12.2.2) on page 285, by specifying that

$$\sigma_{t|t-1}^2 = \omega + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \cdots + \alpha_q r_{t-q}^2 \tag{12.3.1}$$

Here, q is referred to as the ARCH order. Another approach, proposed by Bollerslev (1986) and Taylor (1986), introduces p lags of the conditional variance in the model, where p is referred to as the GARCH order. The combined model is called the generalized autoregressive conditional heteroscedasticity, GARCH(p, q), model.

$$\begin{aligned}
\sigma_{t|t-1}^2 &= \omega + \beta_1 \sigma_{t-1|t-2}^2 + \cdots + \beta_p \sigma_{t-p|t-p-1}^2 + \alpha_1 r_{t-1}^2 \\
&\quad + \alpha_2 r_{t-2}^2 + \cdots + \alpha_q r_{t-q}^2
\end{aligned} \tag{12.3.2}$$

In terms of the backshift B notation, the model can be expressed as

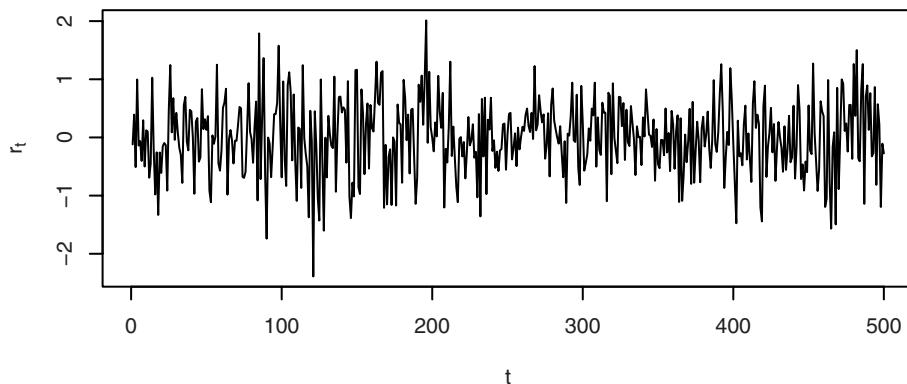
$$(1 - \beta_1 B - \cdots - \beta_p B^p) \sigma_{t|t-1}^2 = \omega + (\alpha_1 B + \cdots + \alpha_q B^q) r_t^2 \tag{12.3.3}$$

We note that in some of the literature, the notation GARCH(p, q) is written as GARCH(q, p); that is, the orders are switched. It can be rather confusing but true that the two different sets of conventions are used in different software! A reader must find out which convention is used by the software on hand before fitting or interpreting a GARCH model.

Because conditional variances must be nonnegative, the coefficients in a GARCH model are often constrained to be nonnegative. However, the nonnegative parameter constraints are not necessary for a GARCH model to have nonnegative conditional variances with probability 1; see Nelson and Cao (1992) and Tsai and Chan (2006). Allowing the parameter values to be negative may increase the dynamical patterns that can be captured by the GARCH model. We shall return to this issue later. Henceforth, within this section, we shall assume the nonnegative constraint for the GARCH parameters.

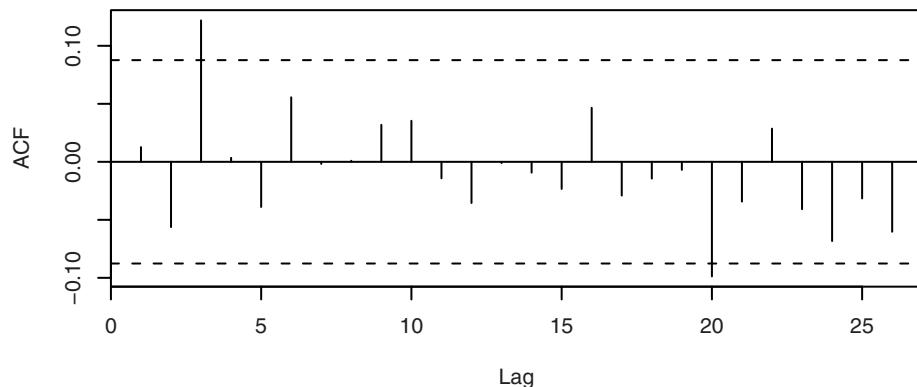
Exhibit 12.12 shows the time series plot of a time series, of size 500, simulated from a GARCH(1,1) model with standard normal innovations and parameter values $\omega = 0.02$, $\alpha = 0.05$, and $\beta = 0.9$. Volatility clustering is evident in the plot, as large (small) fluctuations are usually succeeded by large (small) fluctuations. Moreover, the inclusion of the lag 1 of the conditional variance in the model successfully enhances the memory in the volatility.

Exhibit 12.12 Simulated GARCH(1,1) Process

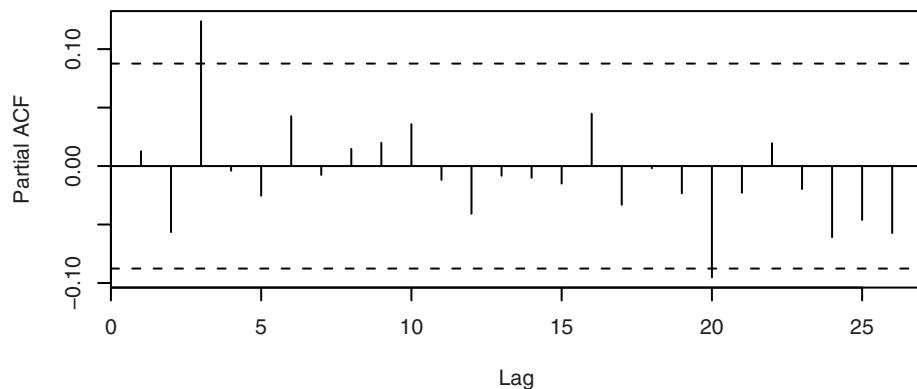


```
> set.seed(1234567)
> garch11.sim=garch.sim(alpha=c(0.02,0.05),beta=.9,n=500)
> plot(garch11.sim,type='l',ylab=expression(r[t]), xlab='t')
```

Except for lags 3 and 20, which are mildly significant, the sample ACF and PACF of the simulated data, shown in Exhibits 12.13 and 12.14, do not show significant correlations. Hence, the simulated process seems to be basically serially uncorrelated as it is.

Exhibit 12.13 Sample ACF of Simulated GARCH(1,1) Process

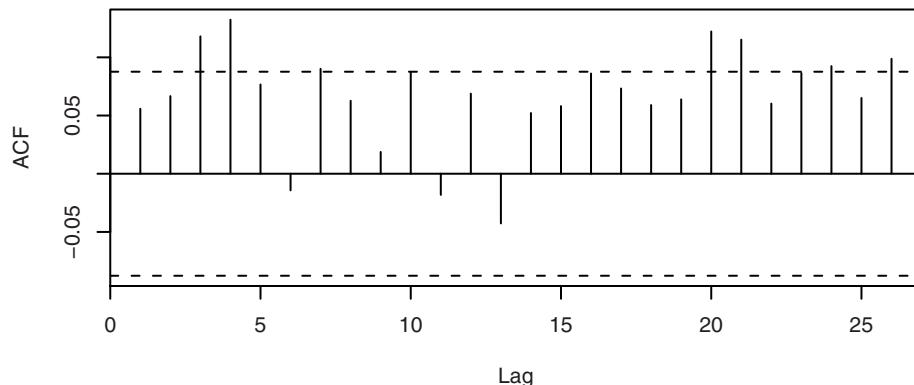
```
> acf(garch11.sim)
```

Exhibit 12.14 Sample PACF of Simulated GARCH(1,1) Process

```
> pacf(garch11.sim)
```

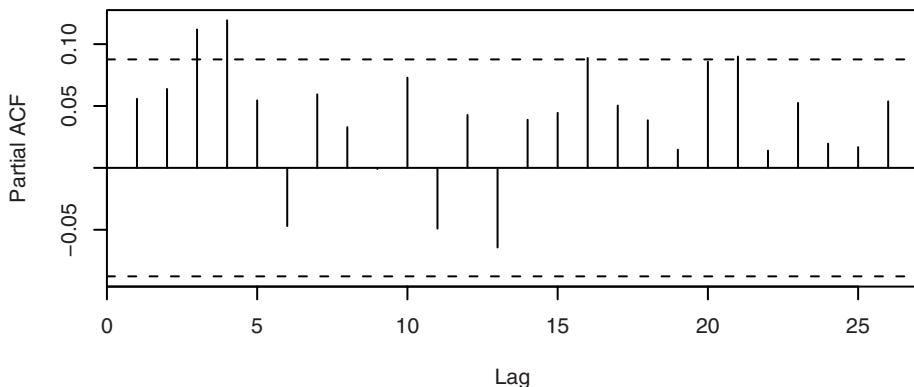
Exhibits 12.15 through 12.18 show the sample ACF and PACF of the absolute values and the squares of the simulated data.

Exhibit 12.15 Sample ACF of the Absolute Values of the Simulated GARCH(1,1) Process



```
> acf(abs(garch11.sim))
```

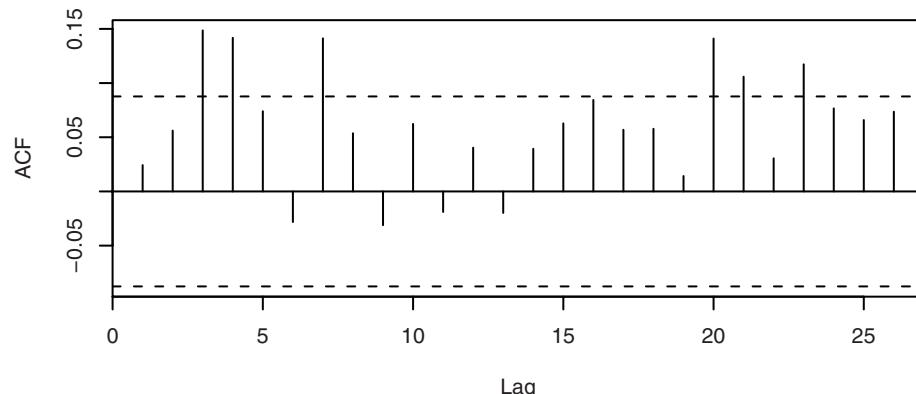
Exhibit 12.16 Sample PACF of the Absolute Values of the Simulated GARCH(1,1) Process



```
> pacf(abs(garch11.sim))
```

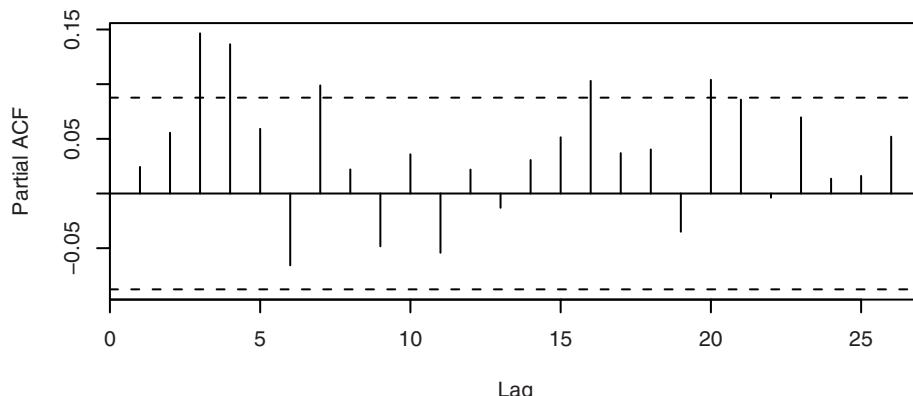
These plots indicate the existence of significant autocorrelation patterns in the absolute and squared data and indicate that the simulated process is in fact serially dependent. Interestingly, the lag 1 autocorrelations are not significant in any of these last four plots.

Exhibit 12.17 Sample ACF of the Squared Values of the Simulated GARCH(1,1) Process



```
> acf(garch11.sim^2)
```

Exhibit 12.18 Sample PACF of the Squared Values of the Simulated GARCH(1,1) Process



```
> pacf(garch11.sim^2)
```

For model identification of the GARCH orders, it is again advantageous to express the model for the conditional variances in terms of the squared returns. Recall the definition $\eta_t = r_t^2 - \sigma_{t|t-1}^2$. Similar to the ARCH(1) model, we can show that $\{\eta_t\}$ is a serially uncorrelated sequence. Moreover, η_t is uncorrelated with past squared returns. Substituting the expression $\sigma_{t|t-1}^2 = r_t^2 - \eta_t$ into Equation (12.3.2) yields

$$\begin{aligned} r_t^2 = \omega + (\beta_1 + \alpha_1)r_{t-1}^2 + \cdots + (\beta_{\max(p,q)} + \alpha_{\max(p,q)})r_{t-\max(p,q)}^2 \\ + \eta_t - \beta_1 \eta_{t-1} - \cdots - \beta_p \eta_{t-p} \end{aligned} \quad (12.3.4)$$

where $\beta_k = 0$ for all integers $k > p$ and $\alpha_k = 0$ for $k > q$. This shows that the GARCH(p,q) model for the return series implies that the model for the squared returns is an ARMA($\max(p,q), p$) model. Thus, we can apply the model identification techniques for ARMA models to the squared return series to identify p and $\max(p,q)$. Notice that if q is smaller than p , it will be masked in the model identification. In such cases, we can first fit a GARCH(p,p) model and then estimate q by examining the significance of the resulting ARCH coefficient estimates.

As an illustration, Exhibit 12.19 shows the sample EACF of the squared values from the simulated GARCH(1,1) series.

Exhibit 12.19 Sample EACF for the Squared Simulated GARCH(1,1) Series

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	o	o	x	x	o	o	x	o	o	o	o	o	o	o
1	x	o	o	o	x	o	x	x	o	o	o	o	o	o
2	x	o	o	o	o	o	x	o	o	o	o	o	o	o
3	x	x	x	o	o	x	o	o	o	o	o	o	o	o
4	x	x	o	x	x	o	o	o	o	o	o	o	o	o
5	x	o	x	x	o	o	o	o	o	o	o	o	o	o
6	x	o	x	x	o	x	o	o	o	o	o	o	o	o
7	x	x	x	x	x	x	o	o	o	o	o	o	o	o

> eacf((garch11.sim)^2)

The pattern in the EACF table is not very clear, although an ARMA(2,2) model seems to be suggested. The fuzziness of the signal in the EACF table is likely caused by the larger sampling variability when we deal with higher moments. Shin and Kang (2001) argued that, to a first-order approximation, a power transformation preserves the theoretical autocorrelation function and hence the order of a stationary ARMA process. Their result suggests that the GARCH order may also be identified by studying the absolute returns. Indeed, the sample EACF table for the absolute returns, shown in Exhibit 12.20, more convincingly suggests an ARMA(1,1) model, and therefore a GARCH(1,1) model for the original data, although there is also a hint of a GARCH(2,2) model.

Exhibit 12.20 Sample EACF for Absolute Simulated GARCH(1,1) Series

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	o	o	x	x	o	o	x	o	o	o	o	o	o	o
1	x	o	o	o	x	o	o	o	o	o	o	o	o	o
2	x	x	o	o	o	o	o	o	o	o	o	o	o	o
3	x	x	o	o	o	x	o	o	o	o	o	o	o	o
4	x	x	o	x	o	x	o	o	o	o	o	o	o	o
5	x	o	x	x	x	o	o	o	o	o	o	o	o	o
6	x	o	x	x	x	x	o	o	o	o	o	o	o	o
7	x	x	x	x	x	o	x	o	o	o	o	o	o	o

```
> eacf(abs(garch11.sim))
```

For the absolute CREF daily return data, the sample EACF table is reported in Exhibit 12.21, which suggests a GARCH(1,1) model. The corresponding EACF table for the squared CREF returns (not shown) is, however, less clear and may suggest a GARCH(2,2) model.

Exhibit 12.21 Sample EACF for the Absolute Daily CREF Returns

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	o	o	o	o	o	o	o	o	o	x	x	o	o	o
1	x	o	o	o	o	o	o	o	o	o	o	o	o	o
2	x	o	o	o	o	o	o	o	o	o	o	o	o	o
3	x	o	x	o	o	o	o	o	o	o	o	o	o	o
4	x	o	x	o	o	o	o	o	o	o	o	o	o	o
5	x	x	x	x	o	o	o	o	o	o	o	o	o	o
6	x	x	x	x	o	o	o	o	o	o	o	o	o	o
7	x	x	x	x	o	o	o	o	o	o	o	o	o	o

```
> eacf(abs(r.cref))
```

Furthermore, the parameter estimates of the fitted ARMA model for the absolute data may yield initial estimates for maximum likelihood estimation of the GARCH model. For example, Exhibit 12.22 reports the estimated parameters of the fitted ARMA(1,1) model for the absolute simulated GARCH(1,1) process.

Exhibit 12.22 Parameter Estimates with ARMA(1,1) Model for the Absolute Simulated GARCH(1,1) Series

Coefficient	ar1	ma1	Intercept
Estimate	0.9821	-0.9445	0.5077
s.e.	0.0134	0.0220	0.0499
> arima(abs(garch11.sim), order=c(1, 0, 1))			

Using Equation (12.3.4), it can be seen that β is estimated by 0.9445, α is estimated by $0.9821 - 0.9445 = 0.03763$, and ω can be estimated as the variance of the original data times the estimate of $1 - \alpha - \beta$, which equals 0.0073. Amazingly, these estimates turn out to be quite close to the maximum likelihood estimates reported in the next section!

We now derive the condition for a GARCH model to be weakly stationary. Assume for the moment that the return process is weakly stationary. Taking expectations on both sides of Equation (12.3.4) gives an equation for the unconditional variance σ^2

$$\sigma^2 = \omega + \sigma^2 \sum_{i=1}^{\max(p, q)} (\beta_i + \alpha_i) \quad (12.3.5)$$

so that

$$\sigma^2 = \frac{\omega}{1 - \sum_{i=1}^{\max(p, q)} (\beta_i + \alpha_i)} \quad (12.3.6)$$

which is finite if

$$\sum_{i=1}^{\max(p, q)} (\beta_i + \alpha_i) < 1 \quad (12.3.7)$$

This condition can be shown to be necessary and sufficient for the weak stationarity of a GARCH(p, q) model. (Recall that we have implicitly assumed that $\alpha_1 \geq 0, \dots, \alpha_p \geq 0$, and $\beta_1 \geq 0, \dots, \beta_q \geq 0$.) Henceforth, we assume $p = q$ for ease of notation.

As in the case of an ARCH(1) model, finiteness of higher moments of the GARCH model requires further stringent conditions on the coefficients; see Ling and McAleer (2002). Also, the stationary distribution of a GARCH model is generally fat-tailed even if the innovations are normal.

In terms of forecasting the h -step-ahead conditional variance $\sigma_{t+h|t}^2$, we can repeat the arguments used in the preceding section to derive the recursive formula that for $h > p$

$$\sigma_{t+h|t}^2 = \omega + \sum_{i=1}^p (\alpha_i + \beta_i) \sigma_{t+h-i|t}^2 \quad (12.3.8)$$

More generally, for arbitrary $h \geq 1$, the formula is more complex, as

$$\sigma_{t+h|t}^2 = \omega + \sum_{i=1}^p \alpha_i \sigma_{t+h-i|t}^2 + \sum_{i=1}^p \beta_i \hat{\sigma}_{t+h-i|t+h-i-1}^2 \quad (12.3.9)$$

where

$$\sigma_{t+h|t}^2 = r_{t+h}^2 \text{ for } h < 0 \quad (12.3.10)$$

and

$$\hat{\sigma}_{t+h-i|t+h-i-1}^2 = \begin{cases} \sigma_{t+h-i|t}^2 & \text{for } h-i-1 > 0 \\ \sigma_{t+h-i|t+h-i-1}^2 & \text{otherwise} \end{cases} \quad (12.3.11)$$

The computation of the conditional variances may be best illustrated using the GARCH(1,1) model. Suppose that there are n observations r_1, r_2, \dots, r_n and

$$\sigma_{t|t-1}^2 = \omega + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1|t-2}^2 \quad (12.3.12)$$

To compute the conditional variances for $2 \leq t \leq n$, we need to set the initial value $\sigma_{1|0}^2$. This may be set to the stationary unconditional variance $\sigma^2 = \omega/(1 - \alpha_1 - \beta_1)$ under the stationarity assumption or simply as r_1^2 . Thereafter, we can compute $\sigma_{t|t-1}^2$ by the formula defining the GARCH model. It is interesting to observe that

$$\sigma_{t|t-1}^2 = (1 - \alpha_1 - \beta_1)\sigma^2 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1|t-2}^2 \quad (12.3.13)$$

so that the estimate of the one-step-ahead conditional volatility is a weighted average of the long-run variance, the current squared return, and the current estimate of the conditional volatility. Further, the MA(∞) representation of the conditional variance implies that

$$\sigma_{t|t-1}^2 = \sigma^2 + \alpha_1(r_{t-1}^2 + \beta_1 r_{t-2}^2 + \beta_1^2 r_{t-3}^2 + \beta_1^3 r_{t-4}^2 + \dots) \quad (12.3.14)$$

an infinite moving average of past squared returns. The formula shows that the squared returns in the distant past receive exponentially diminishing weights. In contrast, simple moving averages of the squared returns are sometimes used to estimate the conditional variance. These, however, suffer much larger bias.

If $\alpha_1 + \beta_1 = 1$, then the GARCH(1,1) model is nonstationary and instead is called an IGARCH(1,1) model with the letter I standing for integrated. In such a case, we shall drop the subscript from the notation and let $\alpha = 1 - \beta$. Suppose that $\omega = 0$. Then

$$\sigma_{t|t-1}^2 = (1 - \beta)(r_{t-1}^2 + \beta r_{t-2}^2 + \beta^2 r_{t-3}^2 + \beta^3 r_{t-4}^2 + \dots), \quad (12.3.15)$$

an exponentially weighted average of the past squared returns. The famed Riskmetrics software in finance employs the IGARCH(1,1) model with $\beta = 0.94$ for estimating conditional variances; see Andersen et al. (2006).

12.4 Maximum Likelihood Estimation

The likelihood function of a GARCH model can be readily derived for the case of normal innovations. We illustrate the computation for the case of a stationary GARCH(1,1) model. Extension to the general case is straightforward. Given the parameters ω , α , and β , the conditional variances can be computed recursively by the formula

$$\sigma_{t|t-1}^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1|t-2}^2 \quad (12.4.1)$$

for $t \geq 2$, with the initial value, $\sigma_1^2|0$, set under the stationarity assumption as the stationary unconditional variance $\sigma^2 = \omega/(1 - \alpha - \beta)$. We use the conditional pdf

$$f(r_t|r_{t-1}, \dots, r_1) = \frac{1}{\sqrt{2\pi\sigma_{t|t-1}^2}} \exp[-r_t^2/(2\sigma_{t|t-1}^2)] \quad (12.4.2)$$

and the joint pdf

$$f(r_n, \dots, r_1) = f(r_{n-1}, \dots, r_1)f(r_n|r_{n-1}, \dots, r_1) \quad (12.4.3)$$

Iterating this last formula and taking logs gives the following formula for the log-likelihood function:

$$L(\omega, \alpha, \beta) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^n \left\{ \log(\sigma_{t-1|t-2}^2) + r_t^2/\sigma_{t|t-1}^2 \right\} \quad (12.4.4)$$

There is no closed-form solution for the maximum likelihood estimators of ω , α , and β , but they can be computed by maximizing the log-likelihood function numerically. The maximum likelihood estimators can be shown to be approximately normally distributed with the true parameter values as their means. Their covariances may be collected into a matrix denoted by Λ , which can be obtained as follows. Let

$$\theta = \begin{bmatrix} \omega \\ \alpha \\ \beta \end{bmatrix} \quad (12.4.5)$$

be the vector of parameters. Write the i th component of θ as θ_i so that $\theta_1 = \omega$, $\theta_2 = \alpha$, and $\theta_3 = \beta$. The diagonal elements of Λ are the approximate variances of the estimators, whereas the off-diagonal elements are their approximate covariances. So, the first diagonal element of Λ is the approximate variance of $\hat{\omega}$, the (1,2)th element of Λ is the approximate covariance between $\hat{\omega}$ and $\hat{\alpha}$, and so forth. We now outline the computation of Λ . Readers not interested in the mathematical details may skip the rest of this paragraph. The 3×3 matrix Λ is approximately equal to the inverse matrix of the 3×3 matrix whose (i, j) th element equals

$$\frac{1}{2} \sum_{t=1}^n \frac{1}{\sigma_{t|t-1}^4} \frac{\partial \sigma_{t|t-1}^2}{\partial \theta_i} \frac{\partial \sigma_{t|t-1}^2}{\partial \theta_j} \quad (12.4.6)$$

The partial derivatives in this expression can be obtained recursively by differentiating Equation (12.4.1). For example, differentiating both sides of Equation (12.4.1) with respect to ω yields the recursive formula

$$\frac{\partial \sigma_{t|t-1}^2}{\partial \omega} = 1 + \beta \frac{\partial \sigma_{t-1|t-2}^2}{\partial \omega} \quad (12.4.7)$$

Other partial derivatives can be computed similarly.

Recall that, in the previous section, the simulated GARCH(1,1) series was identified to be either a GARCH(1,1) model or a GARCH(2,2) model. The model fit of the GARCH(2,2) model is reported in Exhibit 12.23, where the estimate of ω is denoted by a_0 , that of α_1 by a_1 , that of β_1 by b_1 , and so forth. Note that none of the coefficients is significant, although a_2 is close to being significant. The model fit for the GARCH(1,1) model is given in Exhibit 12.24.

Exhibit 12.23 Estimates for GARCH(2,2) Model of a Simulated GARCH(1,1) Series

Coefficient	Estimate	Std. Error	t-value	Pr(> t)
a0	1.835e-02	1.515e-02	1.211	0.2257
a1	4.09e-15	4.723e-02	8.7e-14	1.0000
a2	1.136e-01	5.855e-02	1.940	0.0524
b1	3.369e-01	3.696e-01	0.911	0.3621
b2	5.100e-01	3.575e-01	1.426	0.1538

```
> g1=garch(garch11.sim,order=c(2,2))
> summary(g1)
```

Exhibit 12.24 Estimates for GARCH(1,1) Model of a Simulated GARCH(1,1) Series

Coefficient	Estimate	Std. Error	t-value	Pr(> t)
a0	0.007575	0.007590	0.998	0.3183
a1	0.047184	0.022308	2.115	0.0344
b1	0.935377	0.035839	26.100	< 0.0001

```
> g2=garch(garch11.sim,order=c(1,1))
> summary(g2)
```

Now all coefficient estimates (except a_0) are significant. The AIC of the fitted GARCH(2,2) model is 961.0, while that of the fitted GARCH(1,1) model is 958.0, and thus the GARCH(1,1) model provides a better fit to the data. (Here, AIC is defined as minus two times the log-likelihood of the fitted model plus twice the number of parameters. As in the case of ARIMA models, a smaller AIC is preferable.) A 95% confidence interval for a parameter is given (approximately) by the estimate ± 1.96 times its standard error. So, an approximate 95% confidence interval for ω equals $(-0.0073, 0.022)$, that of α_1 equals $(0.00345, 0.0909)$, and that of β_1 equals $(0.865, 1.01)$. These all contain their true values of 0.02, 0.05, and 0.9, respectively. Note that the standard error of b_1 is 0.0358. Since the standard error is approximately proportional to $1/\sqrt{n}$, the standard error of b_1 is expected to be about 0.0566 (0.0462) if the sample size n is 200 (300). Indeed, fitting the GARCH(1,1) model to the first 200 simulated data, b_1 was found to equal 0.0603 with standard error equal to 50.39! When the sample size was increased to 300, b_1 became 0.935 with standard error equal to 0.0449. This example illustrates that fitting a GARCH model generally requires a large sample size for the theoretical sampling distribution to be valid and useful; see Shephard (1996, p. 10) for a relevant discussion.

For the CREF return data, we earlier identified either a GARCH(1,1) or GARCH(2,2) model. The AIC of the fitted GARCH(1,1) model is 969.6, whereas that of the GARCH(2,2) model is 970.3. Hence the GARCH(1,1) model provides a marginally better fit to the data. Maximum likelihood estimates of the fitted GARCH(1,1) model are reported in Exhibit 12.25.

Exhibit 12.25 Maximum Likelihood Estimates of the GARCH(1,1) Model for the CREF Stock Returns

Parameter	Estimate [†]	Std. Error	t-value	Pr(> t)
a0	0.01633	0.01237	1.320	0.1869
a1	0.04414	0.02097	2.105	0.0353
b1	0.91704	0.04570	20.066	< 0.0001

[†] As remarked earlier, the analysis depends on the scale of measurement. In particular, a GARCH(1,1) model based on the raw CREF stock returns yields estimates $a_0 = 0.00000511$, $a_1 = 0.0941$, and $b_1 = 0.789$.

```
> m1=garch(x=r.cref,order=c(1,1))
> summary(m1)
```

Note that the long-term variance of the GARCH(1,1) model is estimated to be

$$\hat{\omega}/(1 - \hat{\alpha} - \hat{\beta}) = 0.01633/(1 - 0.04414 - 0.91704) = 0.4206 \quad (12.4.8)$$

which is very close to the sample variance of 0.4161.

In practice, the innovations need not be normally distributed. In fact, many financial time series appear to have nonnormal innovations. Nonetheless, we can proceed to esti-

mate the GARCH model by pretending that the innovations are normal. The resulting likelihood function is called the Gaussian likelihood, and estimators maximizing the Gaussian likelihood are called the *quasi-maximum likelihood estimators* (QMLEs). It can be shown that, under some mild regularity conditions, including stationarity, the quasi-maximum likelihood estimators are approximately normal, centered at the true parameter values, and their covariance matrix equals $[(\kappa + 2)/2]\Lambda$, where κ is the (excess) kurtosis of the innovations and Λ is the covariance matrix assuming the innovations are normally distributed—see the discussion above for the normal case. Note that the heavy-tailedness of the innovations will inflate the covariance matrix and hence result in less reliable parameter estimates. In the case where the innovations are deemed nonnormal, this result suggests a simple way to adjust the standard errors of the quasi-maximum likelihood estimates by multiplying the standard errors of the Gaussian likelihood estimates from a routine that assumes normal innovations by $\sqrt{(\kappa + 2)/2}$, where κ can be substituted with the sample kurtosis of the standardized residuals that are defined below. It should be noted that one disadvantage of QMLE is that the AIC is not strictly applicable.

Let the estimated conditional standard deviation be denoted by $\hat{\sigma}_{t|t-1}$. The standardized residuals are then defined as

$$\hat{\varepsilon}_t = r_t / \hat{\sigma}_{t|t-1} \quad (12.4.9)$$

The standardized residuals from the fitted model are proxies for the innovations and can be examined to cast light on the distributional form of the innovations. Once a (parameterized) distribution for the innovations is specified, for example a t -distribution, the corresponding likelihood function can be derived and optimized to obtain maximum likelihood estimators; see Tsay (2005) for details. The price of not correctly specifying the distributional form of the innovation is a loss in efficiency of estimation, although, with large datasets, the computational convenience of the Gaussian likelihood approach may outweigh the loss of estimation efficiency.

12.5 Model Diagnostics

Before we accept a fitted model and interpret its findings, it is essential to check whether the model is correctly specified, that is, whether the model assumptions are supported by the data. If some key model assumptions seem to be violated, then a new model should be specified; fitted, and checked again until a model is found that provides an adequate fit to the data. Recall that the standardized residuals are defined as

$$\hat{\varepsilon}_t = r_t / \hat{\sigma}_{t|t-1} \quad (12.5.1)$$

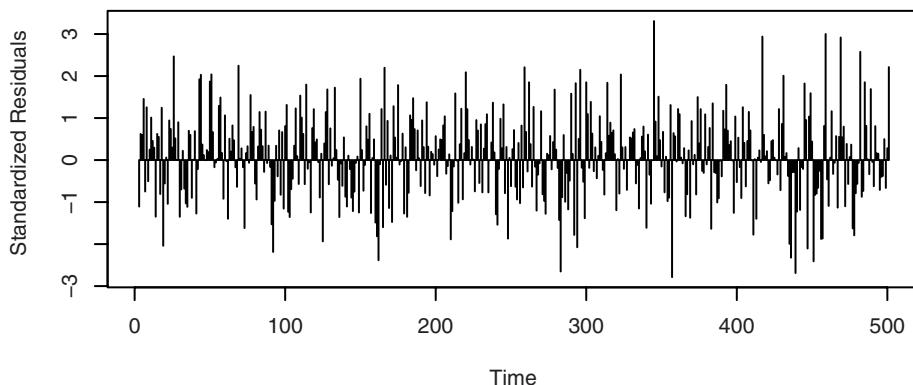
which are approximately independently and identically distributed if the model is correctly specified. As in the case of model diagnostics for ARIMA models, the standardized residuals are very useful for checking the model specification. The normality assumption of the innovations can be explored by plotting the QQ normal scores plot. Deviations from a straight line pattern in the QQ plot furnish evidence against normality and may provide clues on the distributional form of the innovations. The Shapiro-Wilk

test and the Jarque-Bera test are helpful for formally testing the normality of the innovations.[†]

For the GARCH(1,1) model fitted to the simulated GARCH(1,1) process, the sample skewness and kurtosis of the standardized residuals equal -0.0882 and -0.104 , respectively. Moreover, both the Shapiro-Wilk test and the Jarque-Bera test suggest that the standardized residuals are normal.

For the GARCH(1,1) model fitted to the CREF return data, the standardized residuals are plotted in Exhibit 12.26. There is some tendency for the residuals to be larger in magnitude towards the end of the study period, perhaps suggesting that there is some residual pattern in the volatility. The QQ plot of the standardized residuals is shown in Exhibit 12.27. The QQ plot shows a largely straight-line pattern. The skewness and the kurtosis of the standardized residuals are 0.0341 and 0.205 , respectively. The p -value of the Jarque-Bera test equals 0.58 and that of the Shapiro-Wilk test is 0.34 . Hence, the normality assumption cannot be rejected.

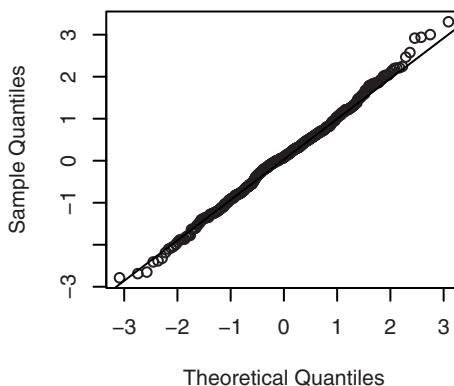
Exhibit 12.26 Standardized Residuals from the Fitted GARCH(1,1) Model of Daily CREF Returns



```
> plot(residuals(m1), type='h', ylab='Standardized Residuals')
```

[†] Chen and Kuan (2006) have shown that the Jarque-Bera test with the residuals from a GARCH model is no longer approximately chi-square distributed under the null hypothesis of normal innovations. Their simulation results suggest that, in such cases, the Jarque-Bera test tends to be liberal; that is, it rejects the normality hypothesis more often than its nominal significance level. The authors have proposed a modification of the Jarque-Bera test that retains the chi-square null distribution approximately. Similarly, it can be expected that the Shapiro-Wilk test may require modification when it is applied to residuals from a GARCH model, although the problem seems open.

Exhibit 12.27 QQ Normal Scores Plot of Standardized Residuals from the Fitted GARCH(1,1) Model of Daily CREF Returns



```
> win.graph(width=2.5, height=2.5, pointsize=8)
> qqnorm(residuals(m1)); qqline(residuals(m1))
```

If the GARCH model is correctly specified, then the standardized residuals $\{\hat{\varepsilon}_t\}$ should be close to independently and identically distributed. The independently and identically distributed assumption of the innovations can be checked by examining their sample acf. Recall that the portmanteau statistic equals

$$n \sum_{k=1}^m \hat{\rho}_k^2$$

where $\hat{\rho}_k$ is the lag k autocorrelation of the standardized residuals and n is the sample size. (Recall that the same statistic is also known as the Box-Pierce statistic and, in a modified version, the Ljung-Box statistic.) Furthermore, it can be shown that the test statistic is approximately χ^2 distributed with m degrees of freedom under the null hypothesis that the model is correctly specified. This result relies on the fact that the sample autocorrelations of nonzero lags from an independently and identically distributed sequence are approximately independent and normally distributed with zero mean and variance $1/n$, and this result holds approximately also for the sample autocorrelations of the standardized residuals if the data are truly generated by a GARCH model of the same orders as those of the fitted model. However, the portmanteau test does not have strong power against uncorrelated and yet serially dependent innovations. In fact, we start out with the assumption that the return data are uncorrelated, so the preceding test is of little interest.

More useful tests may be devised by studying the autocorrelation structure of the absolute standardized residuals or the squared standardized residuals. Let the lag k autocorrelation of the absolute standardized residuals be denoted by $\hat{\rho}_{k,1}$ and that of the squared standardized residuals by $\hat{\rho}_{k,2}$. Unfortunately, the approximate χ^2 distribution with m degrees of freedom for the corresponding portmanteau statistics based on $\hat{\rho}_{k,1}$ ($\hat{\rho}_{k,2}$) is no longer valid, the reason being that the estimation of the unknown param-

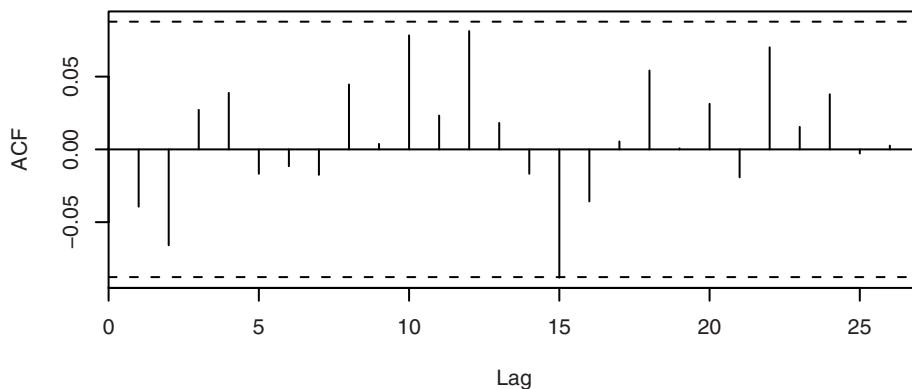
ters induces a nonnegligible effect on the tests. Li and Mak (1994) showed that the χ^2 approximate distribution may be preserved by replacing the sum of squared autocorrelations by a quadratic form in the autocorrelations; see also Li (2003). For the absolute standardized residuals, the test statistic takes the form

$$n \sum_{i=1}^m \sum_{j=1}^m q_{i,j} \hat{\rho}_{i,1} \hat{\rho}_{j,1} \quad (12.5.2)$$

We shall call this modified test statistic the *generalized portmanteau test statistic*. However, the q 's depend on m , the number of lags, and they are specific to the underlying true model and so must be estimated from the data. For the squared residuals, the q 's take different values. See Appendix I on page 318 for the formulas for the q 's.

We illustrate the generalized portmanteau test with the CREF data. Exhibit 12.28, plots the sample ACF of the squared standardized residuals from the fitted GARCH(1,1) model. The (individual) critical limits in the figure are based on the $1/n$ nominal variance under the assumption of independently and identically distributed data. As discussed above, this nominal value could be very different from the actual variance of the autocorrelations of the squared residuals even when the model is correctly specified. Nonetheless, the general impression from the figure is that the squared residuals are serially uncorrelated.

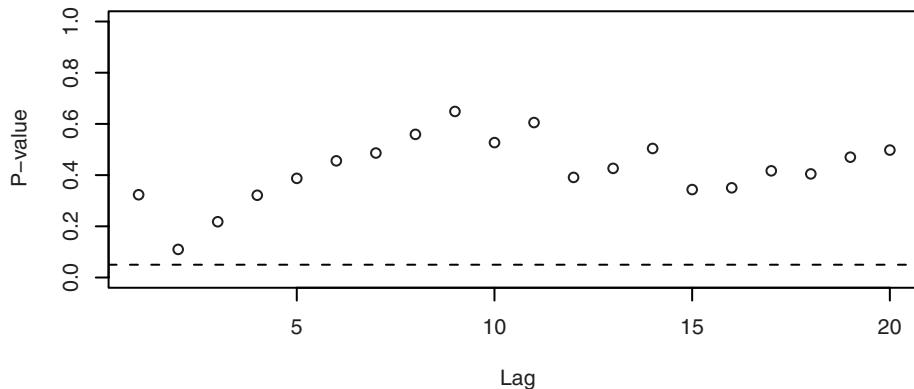
Exhibit 12.28 Sample ACF of Squared Standardized Residuals from the GARCH(1,1) Model of the Daily CREF Returns



```
> acf(residuals(m1)^2, na.action=na.omit)
```

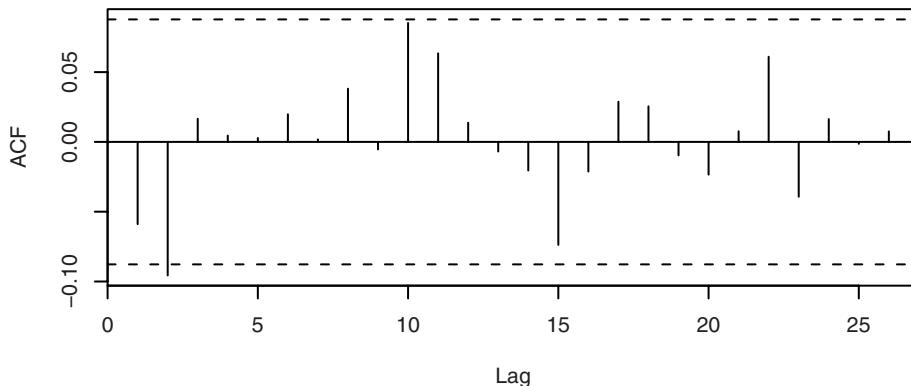
Exhibit 12.29 displays the p -values of the generalized portmanteau tests with the squared standardized residuals from the fitted GARCH(1,1) model of the CREF data for $m = 1$ to 20. All p -values are higher than 5%, suggesting that the squared residuals are uncorrelated over time, and hence the standardized residuals may be independent.

Exhibit 12.29 Generalized Portmanteau Test p -Values for the Squared Standardized Residuals for the GARCH(1,1) Model of the Daily CREF Returns

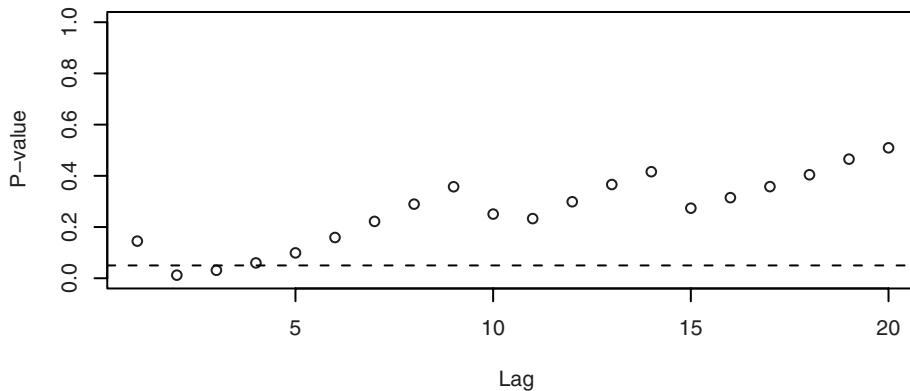


```
> gBox(m1, method='squared')
```

We repeated checking the model using the absolute standardized residuals—see Exhibits 12.30 and 12.31. The lag 2 autocorrelation of the absolute residuals is significant according to the nominal critical limits shown. Furthermore, the generalized portmanteau tests are significant when $m = 2$ and 3 and marginally not significant at $m = 4$. The sample EACF table (not shown) of the absolute standardized residuals suggests an AR(2) model for the absolute residuals and hence points to the possibility that the CREF returns may be identified as a GARCH(1,2) process. However, the fitted GARCH(1,2) model to the CREF data did not improve the fit, as its AIC was 978.2—much higher than 969.6, that of the GARCH(1,1) model. Therefore, we conclude that the fitted GARCH(1,1) model provides a good fit to the CREF data.

Exhibit 12.30 Sample ACF of the Absolute Standardized Residuals from the GARCH(1,1) Model for the Daily CREF Returns


```
> acf(abs(residuals(m1)), na.action=na.omit)
```

Exhibit 12.31 Generalized Portmanteau Test p -Values for the Absolute Standardized Residuals for the GARCH(1,1) Model of the Daily CREF Returns


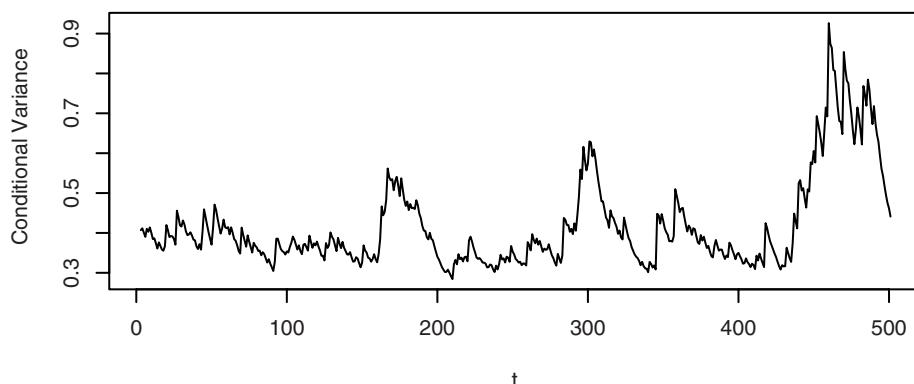
```
> gBox(m1, method='absolute')
```

Given that the GARCH(1,1) model provides a good fit to the CREF data, we may use it to forecast the future conditional variances. Exhibit 12.32 shows the within-sample estimates of the conditional variances, which capture several periods of high volatility, especially the one at the end of the study period. At the final time point, the squared return equals 2.159, and the conditional variance is estimated to be 0.4411. These values combined with Equations (12.3.8) and (12.3.9) can be used to compute the forecasts of future conditional variances. For example, the one-step-ahead forecast of the conditional variance equals $0.01633 + 0.04414 \cdot 2.159 + 0.91704 \cdot 0.4411 = 0.5161$. The two-step forecast of the conditional variance equals $0.01633 + 0.04414 \cdot 0.5161 +$

$0.91704 * 0.5161 = 0.5124$, and so forth, with the longer lead forecasts eventually approaching 0.42066, the long-run variance of the model. The conditional variances may be useful for pricing financial assets through the Black-Scholes formula and calculation of the value at risk (*VaR*); see Tsay (2005) and Andersen et al. (2006).

It is interesting to note that the need for incorporating ARCH in the data is also supported by the McLeod-Li test applied to the residuals of the AR(1) + outlier model; see Exhibit (12.9), page 283.

Exhibit 12.32 Estimated Conditional Variances of the Daily CREF Returns



```
> plot((fitted(m1)[,1])^2, type='l', ylab='Conditional Variance',
      xlab='t')
```

12.6 Conditions for the Nonnegativity of the Conditional Variances

Because the conditional variance $\sigma_{t|t-1}^2$ must be nonnegative, the GARCH parameters are often constrained to be nonnegative. However, the nonnegativity parameter constraints need not be necessary for the nonnegativity of the conditional variances. This issue was first explored by Nelson and Cao (1992) and more recently by Tsai and Chan (2006). To better understand the problem, first consider the case of an ARCH(q) model. Then the conditional variance is given by the formula

$$\sigma_{t|t-1}^2 = \omega + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \cdots + \alpha_q r_{t-q}^2 \quad (12.6.1)$$

Assume that q consecutive returns can take on any arbitrary set of real numbers. If one of the α 's is negative, say $\alpha_1 < 0$, then $\sigma_{t|t-1}^2$ will be negative if r_{t-1}^2 is sufficiently large and the other r 's are sufficiently close to zero. Hence, it is clear that all α 's must be nonnegative for the conditional variances to be nonnegative. Similarly, by letting the returns be close to zero, it can be seen that ω must be nonnegative—otherwise the conditional variance may become negative. Thus, it is clear that for an ARCH model, the

non-negativity of all ARCH coefficients is necessary and sufficient for the conditional variances $\sigma_{t|t-1}^2$ to be always nonnegative.

The corresponding problem for a GARCH(p, q) model can be studied by expressing the GARCH model as an infinite-order ARCH model. The conditional variance process $\{\sigma_{t|t-1}^2\}$ is an ARMA(p, q) model with the squared returns playing the role of the noise process. Recall that an ARMA(p, q) model can be expressed as an MA(∞) model if all the roots of the AR characteristic polynomial lie outside the unit circle. Hence, assuming that all the roots of $1 - \beta_1 x - \beta_2 x^2 - \dots - \beta_p x^p = 0$ have magnitude greater than 1, the conditional variances satisfy the equation

$$\sigma_{t|t-1}^2 = \omega^* + \psi_1 r_{t-1}^2 + \psi_2 r_{t-2}^2 + \dots \quad (12.6.2)$$

where

$$\omega^* = \omega / \left(1 - \sum_{i=1}^p \beta_i \right) \quad (12.6.3)$$

It can be similarly shown that the conditional variances are all nonnegative if and only if ω^* and $\psi_j \geq 0$ for all integers $j \geq 1$. The coefficients in the ARCH(∞) representation relate to the parameters of the GARCH model through the equality

$$\frac{\alpha_1 B + \dots + \alpha_q B^q}{1 - \beta_1 B - \dots - \beta_p B^p} = \psi_1 B + \psi_2 B^2 + \dots \quad (12.6.4)$$

If $p = 1$, then it can be easily checked that $\psi_k = \beta_1 \psi_{k-1}$ for $k > q$. Thus, $\psi_j \geq 0$ for all $j \geq 1$ if and only if $\beta_1 \geq 0$ and $\psi_1 \geq 0, \dots, \psi_q \geq 0$. For higher GARCH order, the situation is more complex. Let λ_j , $1 \leq j \leq p$, be the roots of the characteristic equation

$$1 - \beta_1 x - \dots - \beta_p x^p = 0 \quad (12.6.5)$$

With no loss of generality, we can and shall henceforth assume the convention that

$$|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_p| \quad (12.6.6)$$

Let $i = \sqrt{-1}$ and $\bar{\lambda}$ denote the complex conjugate of λ , $B(x) = 1 - \beta_1 x - \dots - \beta_p x^p$, and $B^{(1)}$ be the first derivative of B . We then have the following result.

Result 1: Consider a GARCH(p, q) model where $p \geq 2$. Assume A1, that all the roots of the equation

$$1 - \beta_1 x - \beta_2 x^2 - \dots - \beta_p x^p = 0 \quad (12.6.7)$$

have magnitude greater than 1, and A2, that none of these roots satisfy the equation

$$\alpha_1 x + \dots + \alpha_q x^q = 0 \quad (12.6.8)$$

Then the following hold:

- (a) $\omega^* \geq 0$ if and only if $\omega \geq 0$.

(b) Assuming the roots $\lambda_1, \dots, \lambda_p$ are distinct, and $|\lambda_1| < |\lambda_2|$, then the conditions given in Equation (12.6.9) are necessary and sufficient for $\psi_k \geq 0$ for all positive integers k :

$$\left. \begin{array}{l} \lambda_1 \text{ is real and } \lambda_1 > 1 \\ \alpha(\lambda_1) > 0 \\ \psi_k \geq 0 \text{ for } k = 1, \dots, k^* \end{array} \right\} \quad (12.6.9)$$

where k^* is the smallest integer greater than or equal to

$$\frac{\log(r_1) - \log[(p-1)r^*]}{\log(|\lambda_1|) - \log(|\lambda_2|)}, \quad r_j = -\frac{\alpha(\lambda_j)}{B^{(1)}(\lambda_j)}, \quad \text{for } 1 \leq j \leq p, \quad \text{and} \quad r^* = \max_{2 \leq j \leq p} (|r_j|) \quad (12.6.10)$$

For $p = 2$, the k^* defined in Result 1 can be shown to be $q + 1$; see Theorem 2 of Nelson and Cao (1992). If the k^* defined in Equations (12.6.10) is a negative number, then it can be seen from the proof given in Tsai and Chan (2006) that $\psi_k \geq 0$ for all positive k .

Tsai and Chan (2006) have also derived some more readily verifiable conditions for the conditional variances to be always nonnegative.

Result 2: Let the assumptions of Result 1 be satisfied. Then the following hold:

(a) For a GARCH($p, 1$) model, if λ_j is real and $\lambda_j > 1$, for $j = 1, \dots, p$, and $\alpha_1 \geq 0$, then $\psi_k \geq 0$ for all positive integers k .

(b) For a GARCH($p, 1$) model, if $\psi_k \geq 0$ for all positive integers k , then $\alpha_1 \geq 0$,

$$\sum_{j=1}^p \lambda_j^{-1} \geq 0, \quad \lambda_1 \text{ is real, and } \lambda_1 > 1.$$

(c) For a GARCH($3, 1$) model, $\psi_k \geq 0$ for all positive integers k if and only if $\alpha_1 \geq 0$ and either of the following cases hold:

Case 1. All the λ_j 's are real numbers, $\lambda_1 > 1$, and $\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} \geq 0$.

Case 2. $\lambda_1 > 1$, and $\lambda_2 = \bar{\lambda}_3 = |\lambda_2|e^{i\theta} = a + bi$, where a and b are real numbers, $b > 0$, and $0 < \theta < \pi$:

Case 2.1. $\theta = 2\pi/r$ for some integer $r \geq 3$, and $1 < \lambda_1 \leq |\lambda_2|$.

Case 2.2. $\theta \notin \{2\pi/r \mid r = 3, 4, \dots\}$, and $|\lambda_2|/\lambda_1 \geq x_0 > 1$, where x_0 is the largest real root of $f_{n,\theta}(x) = 0$, and

$$f_{n,\theta}(x) = x^{n+2} - x \frac{\sin[(n+2)\theta]}{\sin\theta} + \frac{\sin[(n+1)\theta]}{\sin\theta} \quad (12.6.11)$$

where n is the smallest positive integer such that $\sin((n+1)\theta) < 0$ and $\sin((n+2)\theta) > 0$.

(d) For a GARCH(3,1) model, if $\lambda_2 = \bar{\lambda}_3 = |\lambda_2|e^{i\theta} = a + bi$, where a and b are real numbers, $b > 0$, and $a \geq \lambda_1 > 1$, then $\psi_k \geq 0$ for all positive integers k .

(e) For a GARCH(4,1) model, if the λ_j 's are real for $1 \leq j \leq 4$, then a necessary and sufficient condition for $\{\psi_i\}_{i=0}^{\infty}$ to be nonnegative is that $\alpha_1 \geq 0$, $\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \lambda_4^{-1} \geq 0$, and $\lambda_1 > 1$.

Note that x_0 is the only real root of Equation (12.6.11) that is greater than or equal to 1. Also, Tsai and Chan (2006) proved that if the ARCH coefficients (α 's) of a GARCH(p,q) model are all nonnegative, the model has nonnegative conditional variances if the nonnegativity property holds for the associated GARCH($p,1$) models with a nonnegative α_1 coefficient.

12.7 Some Extensions of the GARCH Model

The GARCH model may be generalized in several directions. First, the GARCH model assumes that the conditional mean of the time series is zero. Even for financial time series, this strong assumption need not always hold. In the more general case, the conditional mean structure may be modeled by some ARMA(u,v) model, with the white noise term of the ARMA model modeled by some GARCH(p,q) model. Specifically, let $\{Y_t\}$ be a time series given by (now we switch to using the notation Y_t to denote a general time series)

$$\left. \begin{aligned} Y_t &= \phi_1 Y_{t-1} + \cdots + \phi_u Y_{t-u} \theta_0 + e_t + \theta_1 e_{t-1} + \cdots + \theta_v e_{t-v} \\ e_t &= \sigma_{t|t-1} \varepsilon_t \\ \sigma_{t|t-1}^2 &= \omega + \alpha_1 e_{t-1}^2 + \cdots + \alpha_q e_{t-q}^2 + \beta_1 \sigma_{t-1|t-2}^2 + \cdots + \beta_p \sigma_{t-p|t-p-1}^2 \end{aligned} \right\} \quad (12.7.1)$$

and where we have used the plus convention in the MA parts of the model. The ARMA orders can be identified based on the time series $\{Y_t\}$, whereas the GARCH orders may be identified based on the squared residuals from the fitted ARMA model. Once the orders are identified, full maximum likelihood estimation for the ARMA + GARCH model can be carried out by maximizing the likelihood function as defined in Equation (12.4.4) on page 298 but with r_t there replaced by e_t that are recursively computed according to Equation (12.7.1). The maximum likelihood estimators of the ARMA parameters are approximately independent of their GARCH counterparts if the innovations ε_t have a symmetric distribution (for example, a normal or t -distribution) and their standard errors are approximately given by those in the pure ARMA case. Likewise, the GARCH parameter estimators enjoy distributional results similar to those for the pure GARCH case. However, the ARMA estimators and the GARCH estimators are correlated if the innovations have a skewed distribution. In the next section, we illustrate the ARMA + GARCH model with the daily exchange rates of the U.S. dollar to the Hong Kong dollar.

Another direction of generalization concerns nonlinearity in the volatility process. For financial data, this is motivated by a possible asymmetric market response that may,

for example, react more strongly to a negative return than a positive return of the same magnitude. The idea can be simply illustrated in the setting of an ARCH(1) model, where the asymmetry can be modeled by specifying that

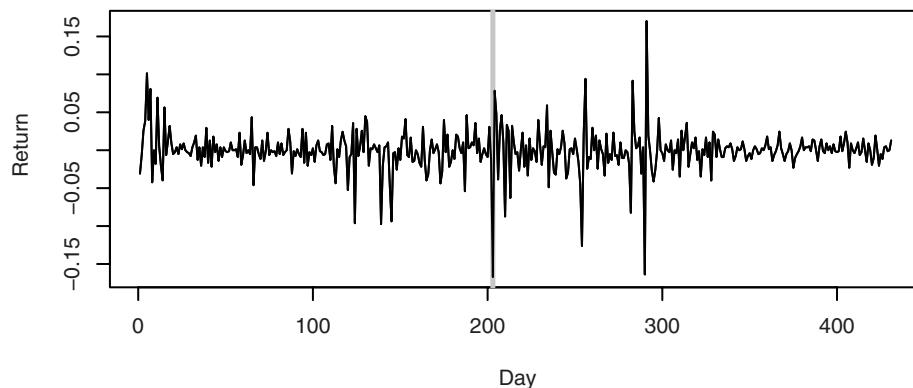
$$\sigma_{t|t-1}^2 = \omega + \alpha e_{t-1}^2 + \gamma \min(e_{t-1}, 0)^2 \quad (12.7.2)$$

Such a model is known as a GJR model—a variant of which allows the threshold to be unknown and other than 0. See Tsay (2005) for other useful extensions of the GARCH models.

12.8 Another Example: The Daily USD/HKD Exchange Rates

As an illustration for the ARIMA + GARCH model, we consider the daily USD/HKD (U.S. dollar to Hong Kong dollar) exchange rate from January 1, 2005 to March 7, 2006, altogether 431 days of data. The returns of the daily exchange rates are shown in Exhibit 12.33 and appear to be stationary, although volatility clustering is evident in the plot.

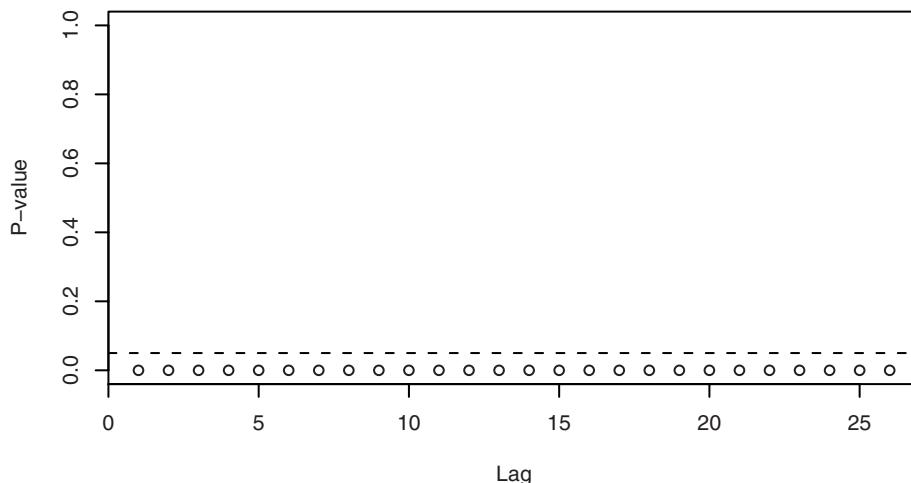
Exhibit 12.33 Daily Returns of USD/HKD Exchange Rate: 1/1/05–3/7/06



```
> data(usd.hkd)
> plot(ts(usd.hkd$hkrate,freq=1),type='l',xlab='Day',
      ylab='Return')
```

It is interesting to note that the need for incorporating ARCH in the data is also supported by the McLeod-Li test applied to the residuals of the AR(1) + outlier model; see below for further discussion of the additive outlier. Exhibit 12.34 shows that the tests are all significant when the number of lags of the autocorrelations of the squared residuals ranges from 1 to 26, displaying strong evidence of conditional heteroscedasticity.

Exhibit 12.34 McLeod-Li Test Statistics for the USD/HKD Exchange Rate



```
> attach(usd.hkd)
> McLeod.Li.test(arima(hkrate,order=c(1,0,0),
+ xreg=data.frame(outlier1)))
```

An AR(1) + GARCH(3,1) model was fitted to the (raw) return data with an additive outlier one day after July 22, 2005, the date when China revalued the yuan by 2.1% and adopted a floating-rate system for it. The outlier is shaded in gray in Exhibit 12.33. The intercept term in the conditional mean function was found to be insignificantly different from zero and hence is omitted from the model. Thus we take the returns to have zero mean unconditionally. The fitted model has an AIC = -2070.9 , being smallest among various competing (weakly) stationary models—see Exhibit 12.35. Interestingly, for lower GARCH orders ($p \leq 2$), the fitted models are nonstationary, but the fitted models are largely stationary when the GARCH order is higher than 2. As the data appear to be stationary, we choose the AR(1) + GARCH(3,1) model as the final model.

The AR + GARCH models partially reported in Exhibit 12.35 were fitted using the Proc Autoreg routine in the SAS software.[†] We used the default option of imposing that the Nelson-Cao inequality constraints for the GARCH conditional variance process be nonnegative. However, the inequality constraints so imposed are only necessary and sufficient for the nonnegativity of the conditional variances of a GARCH(p, q) model for $p \leq 2$. For higher-order GARCH models, Proc Autoreg imposes the constraints that (1) $\psi_k \geq 0$, $1 \leq k \leq \max(q - 1, p) + 1$ and (2) the nonnegativity of the in-sample conditional variances; see the SAS 9.1.3 Help and Documentation manual. Hence, higher-order GARCH models estimated by Proc Autoreg with the Nelson-Cao option need not have nonnegative conditional variances with probability one.

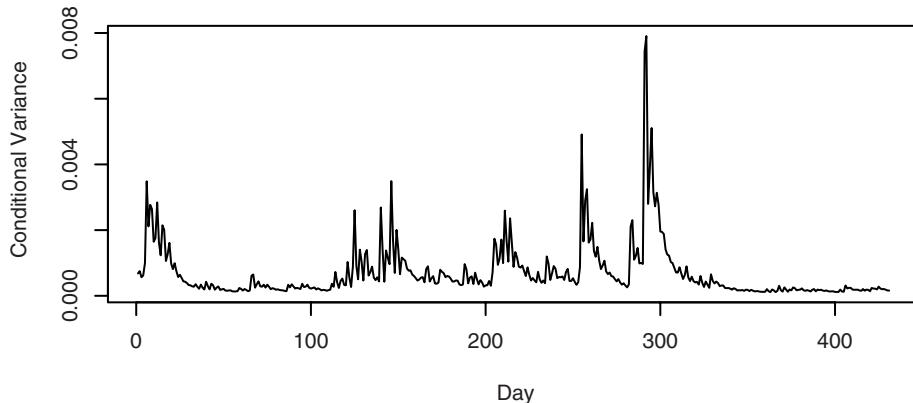
[†] Proc Autoreg of SAS has the option of imposing the Nelson-Cao inequality constraint in the GARCH model, hence it is used here.

Exhibit 12.35 AIC Values for Various Fitted Models for the Daily Returns of the USD/HKD Exchange Rate

AR order	GARCH order (p)	ARCH order (q)	AIC	Stationarity
0	3	1	-1915.3	nonstationary
1	1	1	-2054.3	nonstationary
1	1	2	-2072.5	nonstationary
1	1	3	-2051.0	nonstationary
1	2	1	-2062.2	nonstationary
1	2	2	-2070.5	nonstationary
1	2	3	-2059.2	nonstationary
1	3	1	-2070.9	stationary
1	3	2	-2064.8	stationary
1	3	3	-2062.8	stationary
1	4	1	-2061.7	nonstationary
1	4	2	-2054.8	stationary
1	4	3	-2062.4	stationary
2	3	1	-2066.6	stationary

For the Hong Kong exchange rate data, the fitted model from Proc Autoreg is listed in Exhibit 12.37 with the estimated conditional variances shown in Exhibit 12.36. Note that the GARCH2 (β_2) coefficient estimate is negative.

Exhibit 12.36 Estimated Conditional Variances of the Daily Returns of USD/HKD Exchange Rate from the Fitted AR(1) + GARCH(3,1) Model



```
> plot(ts(usd.hkd$v,freq=1),type='l',xlab='Day',
      ylab='Conditional Variance')
```

Since both the intercept and the ARCH coefficient are positive, we can apply part (c) of Result 2 to check whether or not the conditional variance process defined by the fitted model is always nonnegative. The characteristic equation $1 - \beta_1 x - \beta_2 x^2 - \beta_3 x^3 = 0$ admits three roots equal to 1.153728 and $-0.483294 \pm 1.221474i$. Thus $\lambda_1 = 1.153728$ and $|\lambda_2|/\lambda_1 = 1.138579$. Based on numerical computations, n in Equation (12.6.11) turns out to be 2 and Equation (12.6.11) has one real root equal to 1.1385751 which is strictly less than $1.138579 = |\lambda_2|/\lambda_1$. Hence, we can conclude that the fitted model always results in nonnegative conditional variances.

Exhibit 12.37 Fitted AR(1) + ARCH(3,1) Model for Daily Returns of USD/HKD Exchange Rate

Coefficient	Estimate	Std. error	t-ratio	p-value
AR1	0.1635	0.005892	21.29	0.0022
ARCH0 (ω)	2.374×10^{-5}	6.93×10^{-6}	3.42	0.0006
ARCH1 (α_1)	0.2521	0.0277	9.09	< 0.0001
GARCH1 (β_1)	0.3066	0.0637	4.81	< 0.0001
GARCH2 (β_2)	-0.09400	0.0391	-2.41	0.0161
GARCH3 (β_3)	0.5023	0.0305	16.50	< 0.0001
Outlier	-0.1255	0.00589	-21.29	< 0.0001

```
> SAS code: data hkex; infile 'hkrate.dat'; input hkrate;
  outlier1=0;
  day+1; if day=203 then outlier1=1;
  proc autoreg data=hkex;
    model hkrate=outlier1 /noint nlag=1 garch=(p=3,q=1)
    maxiter=200 archtest;
      /*hetero outlier /link=linear;*/
      output out=a cev=v residual=r;
  run;
```

12.9 Summary

This chapter began with a brief description of some terms and issues associated with financial time series. Autoregressive conditional heteroscedasticity (ARCH) models were then introduced in an attempt to model the changing variance of a time series. The ARCH model of order 1 was thoroughly explored from identification through parameter estimation and prediction. These models were then generalized to the generalized autoregressive conditional heteroscedasticity, GARCH(p, q), model. The GARCH models were also thoroughly explored with respect to identification, maximum likelihood estimation, prediction, and model diagnostics. Examples with both simulated and real time series data were used to illustrate the ideas.

EXERCISES

- 12.1** Display the time sequence plot of the absolute returns for the CREF data. Repeat the plot with the squared returns. Comment on the volatility patterns observed in these plots. (The data are in file named CREF.)
- 12.2** Plot the time sequence plot of the absolute returns for the USD/HKD exchange rate data. Repeat the plot with the squared returns. Comment on the volatility patterns observed in these plots. (The data are in the file named usd.hkd.)
- 12.3** Use the definition $\eta_t = r_t^2 - \sigma_{t|t-1}^2$ [Equation (12.2.4) on page 287] and show that $\{\eta_t\}$ is a serially uncorrelated sequence. Show also that η_t is uncorrelated with past squared returns, that is, show that $\text{Corr}(\eta_t, r_{t-k}^2) = 0$ for $k > 0$.
- 12.4** Substituting $\sigma_{t|t-1}^2 = r_t^2 - \eta_t$ into Equation (12.2.2) on page 285 show the algebra that leads to Equation (12.2.5) on page 287.
- 12.5** Verify Equation (12.2.8) on page 288.
- 12.6** Without doing any theoretical calculations, order the kurtosis values of the following four distributions in ascending order: the t -distribution with 10 DF, the t -distribution with 30 DF, the uniform distribution on $[-1,1]$, and the normal distribution with mean 0 and variance 4. Explain your answer.
- 12.7** Simulate a GARCH(1,1) process with $\alpha = 0.1$ and $\beta = 0.8$ and of length 500. Plot the time series and inspect its sample ACF, PACF, and EACF. Are the data consistent with the assumption of white noise?
- (a) Square the data and identify a GARCH model for the raw data based on the sample ACF, PACF, and EACF of the squared data.
 - (b) Identify a GARCH model for the raw data based on the sample ACF, PACF and EACF of the absolute data. Discuss and reconcile any discrepancy between the tentative model identified with the squared data and that with the absolute data.
 - (c) Perform the McLeod-Li test on your simulated series. What do you conclude?
 - (d) Repeat the exercise but now using only the first 200 simulated data. Discuss your findings.
- 12.8** The file cref.bond contains the daily price of the CREF bond fund from August 26, 2004 to August, 15, 2006. These data are available only on trading days, but proceed to analyze the data as if they were sampled regularly.
- (a) Display the time sequence plot of the daily bond price data and comment on the main features in the data.
 - (b) Compute the daily bond returns by log-transforming the data and then computing the first differences of the transformed data. Plot the daily bond returns, and comment on the result.
 - (c) Perform the McLeod-Li test on the returns series. What do you conclude?
 - (d) Show that the returns of the CREF bond price series appear to be independently and identically distributed and not just serially uncorrelated; that is, there is no discernible volatility clustering.

- 12.9** The daily returns of Google stock from August 14, 2004 to September 13, 2006 are stored in the file named *google*.
- (a) Display the time sequence plot for the return data and show that the data are essentially uncorrelated over time.
 - (b) Compute the mean of the Google daily returns. Does it appear to be significantly different from 0?
 - (c) Perform the McLeod-Li test on the Google daily returns series. What do you conclude?
 - (d) Identify a GARCH model for the Google daily return data. Estimate the identified model and perform model diagnostics with the fitted model.
 - (e) Draw and comment on the time sequence plot of the estimated conditional variances.
 - (f) Plot the QQ normal plot for the standardized residuals from the fitted model. Do the residuals appear to be normal? Discuss the effects of the normality on the model fit, for example, regarding the computation of the confidence interval.
 - (g) Construct a 95% confidence interval for b_1 .
 - (h) What are the stationary mean and variance according to the fitted GARCH model? Compare them with those of the data.
 - (i) Based on the GARCH model, construct the 95% prediction intervals for h -step-ahead forecast, for $h = 1, 2, \dots, 5$.
- 12.10** In Exercise 11.21 on page 276, we investigated the existence of outliers with the logarithms of monthly oil prices within the framework of an IMA(1,1) model. Here, we explore the effects of “outliers” on the GARCH specification. The data are in the file named *oil.price*.
- (a) Based on the sample ACF, PACF, and EACF of the absolute and squared residuals from the fitted IMA(1,1) model (without outlier adjustment), show that a GARCH(1,1) model may be appropriate for the residuals.
 - (b) Fit an IMA(1,1) + GARCH(1,1) model to the logarithms of monthly oil prices.
 - (c) Draw the time sequence plot for the standardized residuals from the fitted IMA(1,1) + GARCH(1,1) model. Are there any outliers?
 - (d) For the log oil prices, fit an IMA(1,1) model with two IOs at $t = 2$ and $t = 56$ and an AO at $t = 8$. Show that the residuals from the IMA plus outlier model appear to be independently and identically distributed and not just serially uncorrelated; that is, there is no discernible volatility clustering.
 - (e) Between the outlier and the GARCH model, which one do you think is more appropriate for the oil price data? Explain your answer.

Appendix I: Formulas for the Generalized Portmanteau Tests

We first present the formula for $Q = (q_{i,j})$ for the case where the portmanteau test is based on the squared standardized residuals. Readers may consult Li and Mak (1994) for proofs of the formulas. Let θ denote the vector of GARCH parameters. For example, for a GARCH(1,1) model,

$$\theta = \begin{bmatrix} \omega \\ \alpha \\ \beta \end{bmatrix} \quad (12.I.1)$$

Write the i th component of θ as θ_i so that $\theta_1 = \omega$, $\theta_2 = \alpha$, and $\theta_3 = \beta$ for the GARCH(1,1) model. In the general case, let $k = p + q + 1$ be the number of GARCH parameters. Let J be an $m \times k$ matrix whose (i,j) th element equals

$$\frac{1}{n} \sum_{t=i+1}^n \frac{1}{\sigma_{t|t-1}^2} \frac{\partial \sigma_{t|t-1}^2}{\partial \theta_j} (\varepsilon_{t-i}^2 - 1) \quad (12.I.2)$$

and Λ be the $k \times k$ covariance matrix of the approximate normal distribution of the maximum likelihood estimator of θ for the model assuming normal innovations; see Section 12.4. Let $Q = (q_{i,j})$ be the matrix of the q 's appearing in the quadratic form of the generalized portmanteau test. It can be shown that the matrix Q equals

$$\left[I - \frac{1}{2(\kappa + 2)} J \Lambda J^T \right]^{-1} \quad (12.I.3)$$

where I is the $m \times m$ identity matrix, κ is the (excess) kurtosis of the innovations, J^T is the transpose of J , and the superscript -1 denotes the matrix inverse.

Next, we present the formulas for the case where the tests are computed based on the absolute standardized residuals. In this case, the (i,j) th element of the J matrix equals

$$\frac{1}{n} \sum_t \frac{1}{\sigma_{t|t-1}^2} \frac{\partial \sigma_{t|t-1}^2}{\partial \theta_j} (|\varepsilon_{t-i}| - \tau) \quad (12.I.4)$$

where $\tau = E(|\varepsilon_t|)$, and Q equals

$$\left[I - \frac{-[(\kappa + 2)\tau^2]/8 + \tau(v - \tau)}{(1 - \tau^2)^2} J \Lambda J^T \right]^{-1} \quad (12.I.5)$$

with $v = E(|\varepsilon_t|^3)$.

CHAPTER 13

INTRODUCTION TO SPECTRAL ANALYSIS

Historically, spectral analysis began with the search for “hidden periodicities” in time series data. Chapter 3 discussed fitting cosine trends at various known frequencies to series with strong cyclical trends. In addition, the random cosine wave example in Chapter 2 on page 18, showed that it is possible for a stationary process to look very much like a deterministic cosine wave. We hinted in Chapter 3 that by using enough different frequencies with enough different amplitudes (and phases) we might be able to model nearly any stationary series.[†] This chapter pursues those ideas further with an introduction to spectral analysis. Previous to this chapter, we concentrated on analyzing the correlation properties of time series. Such analysis is often called *time domain* analysis. When we analyze frequency properties of time series, we say that we are working in the *frequency domain*. Frequency domain analysis or *spectral analysis* has been found to be especially useful in acoustics, communications engineering, geophysical science, and biomedical science, for example.

13.1 Introduction

Recall from Chapter 3 the cosine curve with equation[‡]

$$R \cos(2\pi ft + \Phi) \quad (13.1.1)$$

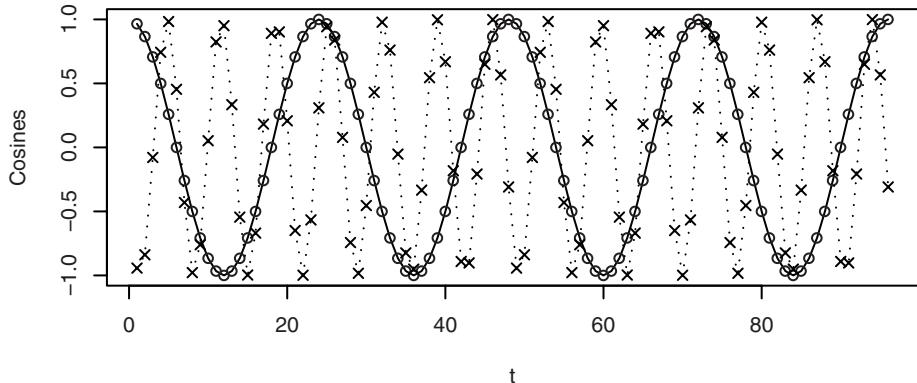
Remember that $R (> 0)$ is the *amplitude*, f the *frequency*, and Φ the *phase* of the curve. Since the curve repeats itself exactly every $1/f$ time units, $1/f$ is called the *period* of the cosine wave.

Exhibit 13.1 displays two discrete-time cosine curves with time running from 1 to 96. We would only see the discrete points, but the connecting line segments are added to help our eyes follow the pattern. The frequencies are $4/96$ and $14/96$, respectively. The lower-frequency curve has a phase of zero, but the higher-frequency curve is shifted by a phase of 0.6π .

Exhibit 13.2 shows the graph of a linear combination of the two cosine curves with a multiplier of 2 on the low-frequency curve and a multiplier of 3 on the higher-frequency curve and a phase of 0.6π ; that is,

[†] See Exercise 2.25 on page 23, in particular.

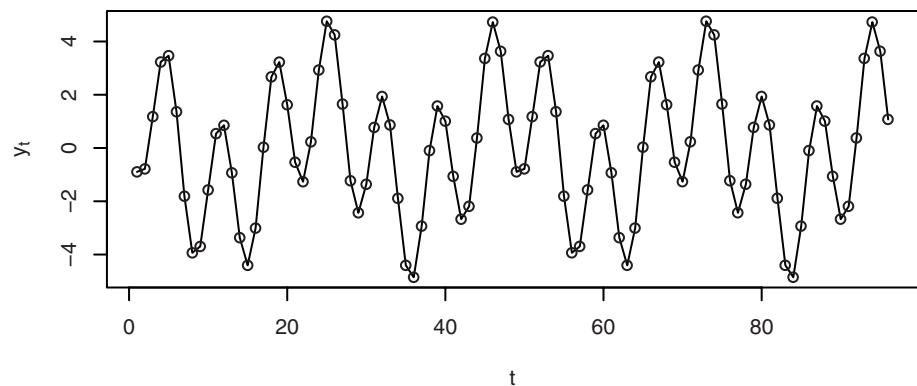
[‡] In this chapter, we use notation slightly different from that in Chapter 3.

Exhibit 13.1 Cosine Curves with $n = 96$ and Two Frequencies and Phases

```
> win.graph(width=4.875, height=2.5, pointsize=8)
> t=1:96; cos1=cos(2*pi*t*4/96); cos2=cos(2*pi*(t*14/96+.3))
> plot(t,cos1, type='o', ylab='Cosines')
> lines(t,cos2,lty='dotted',type='o',pch=4)
```

$$Y_t = 2\cos\left(2\pi t \frac{4}{96}\right) + 3\cos\left[2\pi\left(t \frac{14}{96} + 0.3\right)\right] \quad (13.1.2)$$

Now the periodicity is somewhat hidden. Spectral analysis provides tools for discovering the “hidden” periodicities quite easily. Of course, there is nothing random in this time series.

Exhibit 13.2 Linear Combination of Two Cosine Curves

```
> y=2*cos1+3*cos2; plot(t,y,type='o',ylab=expression(y[t]))
```

As we saw earlier, Equation (13.1.1) is not convenient for estimation because the parameters R and Φ do not enter the expression linearly. Instead, we use a trigonometric identity to reparameterize Equation (13.1.1) as

$$R \cos(2\pi ft + \Phi) = A \cos(2\pi ft) + B \sin(2\pi ft) \quad (13.1.3)$$

where

$$R = \sqrt{A^2 + B^2}, \quad \Phi = \text{atan}(-B/A) \quad (13.1.4)$$

and, conversely,

$$A = R \cos(\Phi), \quad B = -R \sin(\Phi) \quad (13.1.5)$$

Then, for a fixed frequency f , we can use $\cos(2\pi ft)$ and $\sin(2\pi ft)$ as predictor variables and fit the A 's and B 's from the data using ordinary least squares regression.

A general linear combination of m cosine curves with arbitrary amplitudes, frequencies, and phases could be written as[†]

$$Y_t = A_0 + \sum_{j=1}^m [A_j \cos(2\pi f_j t) + B_j \sin(2\pi f_j t)] \quad (13.1.6)$$

Ordinary least squares regression can be used to fit the A 's and B 's, but when the frequencies of interest are of a special form, the regressions are especially easy. Suppose that n is odd and write $n = 2k + 1$. Then the frequencies of the form $1/n, 2/n, \dots, k/n$ ($= 1/2 - 1/(2n)$) are called the *Fourier frequencies*. The cosine and sine predictor variables at these frequencies (and at $f = 0$) are known to be orthogonal,[‡] and the least squares estimates are simply

$$\hat{A}_0 = \bar{Y} \quad (13.1.7)$$

$$\hat{A}_j = \frac{2}{n} \sum_{t=1}^n Y_t \cos(2\pi t j/n) \quad \text{and} \quad \hat{B}_j = \frac{2}{n} \sum_{t=1}^n Y_t \sin(2\pi t j/n) \quad (13.1.8)$$

If the sample size is even, say $n = 2k$, Equations (13.1.7) and (13.1.8) still apply for $j = 1, 2, \dots, k-1$, but

$$\hat{A}_k = \frac{1}{n} \sum_{t=1}^n (-1)^t Y_t \quad \text{and} \quad \hat{B}_k = 0 \quad (13.1.9)$$

Note that here $f_k = k/n = 1/2$.

If we were to apply these formulas to the series shown in Exhibit 13.2, we would obtain perfect results. That is, at frequency $f_4 = 4/96$, we obtain $\hat{A}_4 = 2$ and $\hat{B}_4 = 0$, and at frequency $f_{14} = 14/96$, we obtain $\hat{A}_{14} = -0.927051$ and $\hat{B}_{14} = -2.85317$. We would obtain estimates of zero for the regression coefficients at all other frequencies. These

[†] The A_0 term can be thought of as the coefficient of the cosine curve at zero frequency, which is identically one, and the B_0 can be thought of as the coefficient on the sine curve at frequency zero, which is identically zero and hence does not appear.

[‡] See Appendix J on page 349 for more information on the orthogonality properties of the cosines and sines.

results obtain because there is no randomness in this series and the cosine-sine fits are exact.

Note also that *any* series of any length n , whether deterministic or stochastic and with or without any true periodicities, can be fit perfectly by the model in Equation (13.1.6) by choosing $m = n/2$ if n is even and $m = (n - 1)/2$ if n is odd. There are then n parameters to adjust (estimate) to fit the series of length n .

13.2 The Periodogram

For odd sample sizes with $n = 2k + 1$, the *periodogram* I at frequency $f = j/n$ for $j = 1, 2, \dots, k$, is defined to be

$$I\left(\frac{j}{n}\right) = \frac{n}{2}(\hat{A}_j^2 + \hat{B}_j^2) \quad (13.2.1)$$

If the sample size is even and $n = 2k$, Equations (13.1.7) and (13.1.8) still give the \hat{A} 's and \hat{B} 's and Equation (13.2.1) gives the periodogram for $j = 1, 2, \dots, k - 1$. However, at the extreme frequency $f = k/n = 1/2$, Equations (13.1.9) apply and

$$I\left(\frac{1}{2}\right) = n(\hat{A}_k)^2 \quad (13.2.2)$$

Since the periodogram is proportional to the sum of squares of the regression coefficients associated with frequency $f = j/n$, *the height of the periodogram shows the relative strength of cosine-sine pairs at various frequencies in the overall behavior of the series*. Another interpretation is in terms of an analysis of variance. The periodogram $I(j/n)$ is the sum of squares with two degrees of freedom associated with the coefficient pair (A_j, B_j) at frequency j/n , so we have

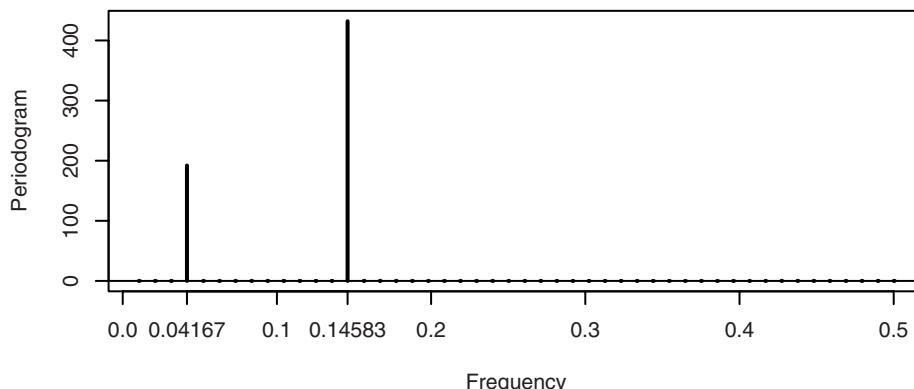
$$\sum_{j=1}^n (Y_j - \bar{Y})^2 = \sum_{j=1}^k I\left(\frac{j}{n}\right) \quad (13.2.3)$$

when $n = 2k + 1$ is odd. A similar result holds when n is even but there is a further term in the sum, $I(1/2)$, with one degree of freedom.

For long series, the computation of a large number of regression coefficients might be intensive. Fortunately, quick, efficient numerical methods based on the fast Fourier transform (FFT) have been developed that make the computations feasible for very long time series.[†]

Exhibit 13.3 displays a graph of the periodogram for the time series in Exhibit 13.2. The heights show the presence and relative strengths of the two cosine-sine components quite clearly. Note also that the frequencies $4/96 \approx 0.04167$ and $14/96 \approx 0.14583$ have been marked on the frequency axis.

[†] Often based on the Cooley-Tukey FFT algorithm; see Gentleman and Sande (1966).

Exhibit 13.3 Periodogram of the Series in Exhibit 13.2

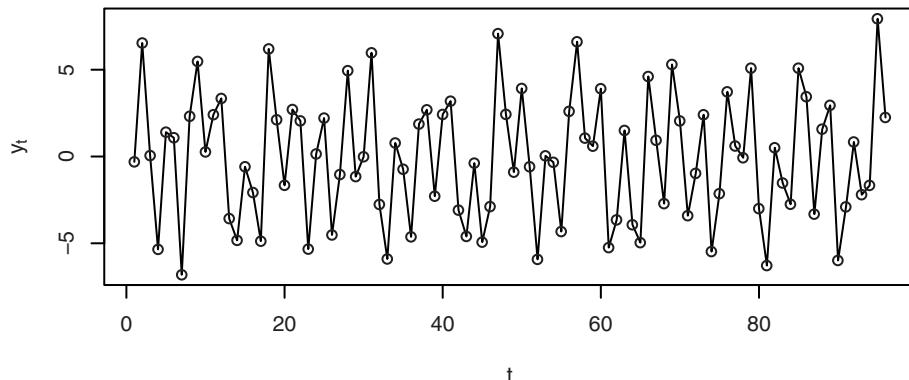
```
> periodogram(y); abline(h=0); axis(1,at=c(0.04167,.14583))
```

Does the periodogram work just as well when we do not know where or even if there are cosines in the series? What if the series contains additional “noise”? To illustrate, we generate a time series using randomness to select the frequencies, amplitudes, and phases and with additional additive white noise. The two frequencies are randomly chosen without replacement from among 1/96, 2/96,..., 47/96. The A 's and B 's are selected independently from normal distributions with means of zero and standard deviations of 2 for the first component and 3 for the second. Finally, a normal white noise series, $\{W_t\}$, with zero mean and standard deviation 1, is chosen independently of the A 's and B 's and added on. The model is[†]

$$Y_t = A_1 \cos(2\pi f_1 t) + B_1 \sin(2\pi f_1 t) + A_2 \cos(2\pi f_2 t) + B_2 \sin(2\pi f_2 t) + W_t \quad (13.2.4)$$

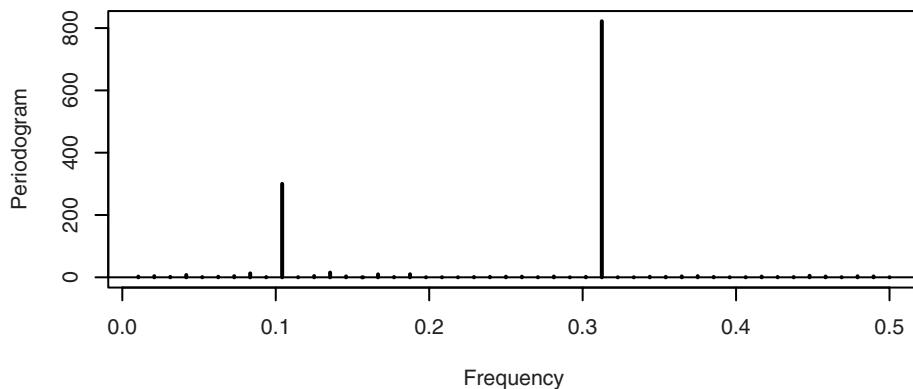
and Exhibit 13.4 displays a time series of length 96 simulated from this model. Once more, the periodicities are not obvious until we view the periodogram shown in Exhibit 13.5.

[†] This model is often described as a *signal plus noise* model. The *signal* could be deterministic (with unknown parameters) or stochastic.

Exhibit 13.4 Time Series with “Hidden” Periodicities

```
> win.graph(width=4.875,height=2.5,pointsize=8)
> set.seed(134); t=1:96; integer=sample(48,2)
> freq1=integer[1]/96; freq2=integer[2]/96
> A1=rnorm(1,0,2); B1=rnorm(1,0,2)
> A2=rnorm(1,0,3); B2=rnorm(1,0,3); w=2*pi*t
> y=A1*cos(w*freq1)+B1*sin(w*freq1)+A2*cos(w*freq2) +
  B2*sin(w*freq2)+rnorm(96,0,1)
> plot(t,y,type='o',ylab=expression(y[t]))
```

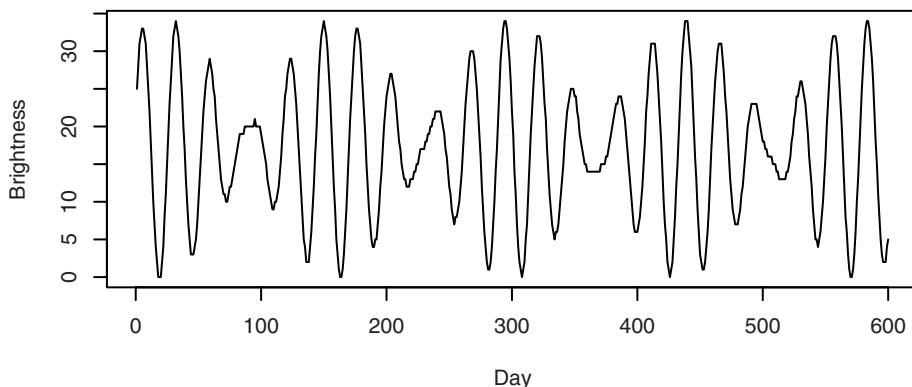
The periodogram clearly shows that the series contains two cosine-sine pairs at frequencies of about 0.11 and 0.32 and that the higher-frequency component is much stronger. There are some other very small spikes in the periodogram, apparently caused by the additive white noise component. (When we checked the simulation in detail, we found that one frequency was chosen as $10/96 \approx 0.1042$ and the other was selected as $30/96 = 0.3125$.)

Exhibit 13.5 Periodogram of the Time Series Shown in Exhibit 13.4

```
> periodogram(y); abline(h=0)
```

Here is an example of the periodogram for a classic time series from Whittaker and Robinson (1924).[†] Exhibit 13.6 displays the time series plot of the brightness (magnitude) of a particular star at midnight on 600 consecutive nights.

Exhibit 13.6 Variable Star Brightness on 600 Consecutive Nights



```
> data(star)
> plot(star,xlab='Day',ylab='Brightness')
```

Exhibit 13.7 shows the periodogram for this time series. There are two very prominent peaks in the periodogram. When we inspect the actual numerical values, we find that the larger peak occurs at frequency $f = 21/600 = 0.035$. This frequency corresponds to a period of $600/21 \approx 28.57$, or nearly 29 days. The secondary peak occurs at $f = 25/600 \approx 0.04167$, which corresponds to a period of 24 days. The much more modest nonzero periodogram values near the major peak are likely caused by *leakage*.

The two sharp peaks suggest a model for this series with just two cosine-sine pairs with the appropriate frequencies or periods, namely

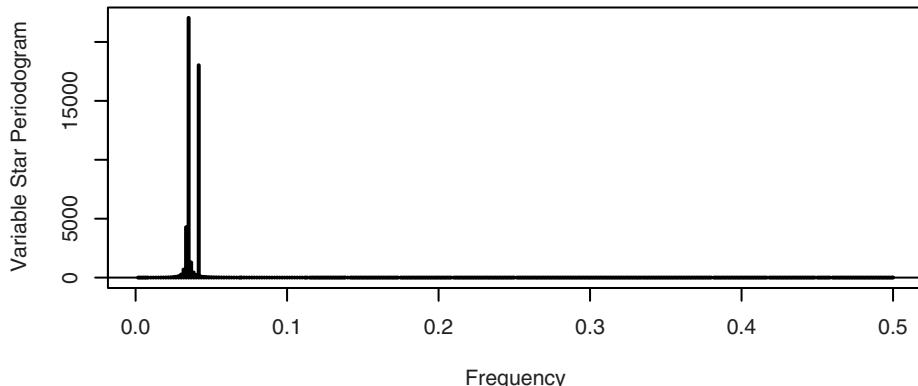
$$Y_t = \beta_0 + \beta_1 \cos(2\pi f_1 t) + \beta_2 \sin(2\pi f_1 t) + \beta_3 \cos(2\pi f_2 t) + \beta_4 \sin(2\pi f_2 t) + e_t \quad (13.2.5)$$

where $f_1 = 1/29$ and $f_2 = 1/24$. If we estimate this regression model as in Chapter 3, we obtain highly statistically significant regression coefficients for all five parameters and a multiple R-square value of 99.9%.

We will return to this time series in Section 14.5 on page 358, where we discuss more about leakage and tapering.

[†] An extensive analysis of this series appears throughout Bloomfield (2000).

Exhibit 13.7 Periodogram of the Variable Star Brightness Time Series



```
> periodogram(star, ylab='Variable Star Periodogram'); abline(h=0)
```

Although the Fourier frequencies are special, we extend the definition of the periodogram to all frequencies in the interval 0 to $\frac{1}{2}$ through the Equations (13.1.8) and (13.2.1). Thus we have for $0 \leq f \leq \frac{1}{2}$

$$I(f) = \frac{n}{2}(\hat{A}_f^2 + \hat{B}_f^2) \quad (13.2.6)$$

where

$$\hat{A}_f = \frac{2}{n} \sum_{t=1}^n Y_t \cos(2\pi t f) \quad \text{and} \quad \hat{B}_f = \frac{2}{n} \sum_{t=1}^n Y_t \sin(2\pi t f) \quad (13.2.7)$$

When viewed in this way, the periodogram is often calculated at a grid of frequencies finer than the Fourier frequencies, and the plotted points are connected by line segments to display a somewhat smooth curve.

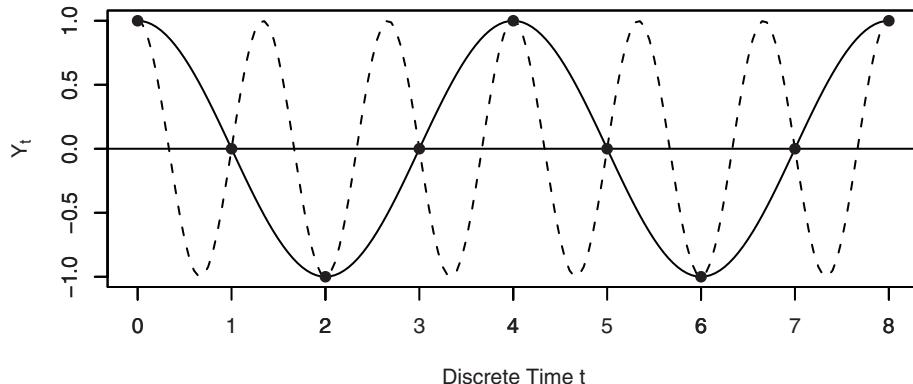
Why do we only consider positive frequencies? Because by the even and odd nature of cosines and sines, any cosine-sine curve with negative frequency, say $-f$, could just as well be expressed as a cosine-sine curve with frequency $+f$. No generality is lost by using positive frequencies.[†]

Secondly, why do we restrict frequencies to the interval from 0 to $\frac{1}{2}$? Consider the graph shown in Exhibit 13.8. Here we have plotted two cosine curves, one with frequency $f = \frac{1}{4}$ and the one shown with dashed lines at frequency $f = \frac{3}{4}$. If we only observe the series at the discrete-time points 0, 1, 2, 3, ..., the two series are identical. With discrete-time observations, we could never distinguish between these two curves. We say that the two frequencies $\frac{1}{4}$ and $\frac{3}{4}$ are **aliased** with one another. In general, each frequency f within the interval 0 to $\frac{1}{2}$ will be aliased with each frequency of the form

[†] The definition of Equation (13.2.6) is often used for $-\frac{1}{2} < f < +\frac{1}{2}$, but the resulting function is symmetric about zero and no new information is gained from the negative frequencies. Later in this chapter, we will use both positive and negative frequencies so that certain nice mathematical relationships hold.

$f + k(\frac{1}{2})$ for any positive integer k , and it suffices to limit attention to frequencies within the interval from 0 to $\frac{1}{2}$.

Exhibit 13.8 Illustration of Aliasing



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> t=seq(0,8,by=.05)
> plot(t,cos(2*pi*t/4),axes=F,type='l',ylab=expression(Y[t]),
+       xlab='Discrete Time t')
> axis(1,at=c(1,2,3,4,5,6,7));axis(1); axis(2); box()
> lines(t,cos(2*pi*t*3/4),lty='dashed',type='l'); abline(h=0)
> points(x=c(0:8),y=cos(2*pi*c(0:8)/4),pch=19)
```

13.3 The Spectral Representation and Spectral Distribution

Consider a time series represented as

$$Y_t = \sum_{j=1}^m [A_j \cos(2\pi f_j t) + B_j \sin(2\pi f_j t)] \quad (13.3.1)$$

where the frequencies $0 < f_1 < f_2 < \dots < f_m < \frac{1}{2}$ are fixed and A_j and B_j are independent normal random variables with zero means and $\text{Var}(A_j) = \text{Var}(B_j) = \sigma_j^2$. Then a straightforward calculation shows that $\{Y_t\}$ is stationary[†] with mean zero and

$$\gamma_k = \sum_{j=1}^m \sigma_j^2 \cos(2\pi k f_j) \quad (13.3.2)$$

In particular, the process variance, γ_0 , is a sum of the variances due to each component at the various fixed frequencies:

[†] Compare this with Exercise 2.29 on page 24.

$$\gamma_0 = \sum_{j=1}^m \sigma_j^2 \quad (13.3.3)$$

If for $0 < f < \frac{1}{2}$ we define two random step functions by

$$a(f) = \sum_{\{j|f_j \leq f\}} A_j \quad \text{and} \quad b(f) = \sum_{\{j|f_j \leq f\}} B_j \quad (13.3.4)$$

then we can write Equation (13.3.1) as

$$Y_t = \int_0^{\frac{1}{2}} \cos(2\pi ft) da(f) + \int_0^{\frac{1}{2}} \sin(2\pi ft) db(f) \quad (13.3.5)$$

It turns out that *any* zero-mean stationary process may be represented as in Equation (13.3.5).[†] It shows how stationary processes may be represented as linear combinations of infinitely many cosine-sine pairs over a continuous frequency band. In general, $a(f)$ and $b(f)$ are zero-mean stochastic processes indexed by frequency on $0 \leq f \leq \frac{1}{2}$, each with uncorrelated[‡] increments, and the increments of $a(f)$ are uncorrelated with the increments of $b(f)$. Furthermore, we have

$$\text{Var}\left(\int_{f_1}^{f_2} da(f)\right) = \text{Var}\left(\int_{f_1}^{f_2} db(f)\right) = F(f_2) - F(f_1), \text{ say.} \quad (13.3.6)$$

Equation (13.3.5) is called the *spectral representation* of the process. The nondecreasing function $F(f)$ defined on $0 \leq f \leq \frac{1}{2}$ is called the *spectral distribution function* of the process.

We say that the special process defined by Equation (13.3.1) has a *purely discrete* (or *line*) *spectrum* and, for $0 \leq f \leq \frac{1}{2}$,

$$F(f) = \sum_{\{j|f_j \leq f\}} \sigma_j^2 \quad (13.3.7)$$

Here the heights of the jumps in the spectral distribution give the variances associated with the various periodic components, and the positions of the jumps indicate the frequencies of the periodic components.

In general, a spectral distribution function has the properties

[†] The proof is beyond the scope of this book. See Cramér and Leadbetter (1967, pp. 128–138), for example. You do not need to understand stochastic Riemann-Stieltjes integrals to appreciate the rest of the discussion of spectral analysis.

[‡] Uncorrelated increments are usually called *orthogonal increments*.

$$\left. \begin{array}{l} 1. \quad F \text{ is nondecreasing} \\ 2. \quad F \text{ is right continuous} \\ 3. \quad F(f) \geq 0 \text{ for all } f \\ 4. \quad \lim_{f \rightarrow 1/2^-} F(f) = \text{Var}(Y_t) = \gamma_0 \end{array} \right\} \quad (13.3.8)$$

If we consider the scaled spectral distribution function $F(f)/\gamma_0$, we have a function with the same mathematical properties as a cumulative distribution function (CDF) for a random variable on the interval 0 to $\frac{1}{2}$ since now $F(\frac{1}{2})/\gamma_0 = 1$.

We interpret the spectral distribution by saying that, for $0 \leq f_1 < f_2 \leq \frac{1}{2}$, the integral

$$\int_{f_1}^{f_2} dF(f) \quad (13.3.9)$$

gives the portion of the (total) process variance $F(\frac{1}{2}) = \gamma_0$ that is attributable to frequencies in the range f_1 to f_2 .

Sample Spectral Density

In spectral analysis, it is customary to first remove the sample mean from the series. For the remainder of this chapter, we assume that in the definition of the periodogram, Y_t represents deviations from its sample mean. Furthermore, for mathematical convenience, we now let various functions of frequency, such as the periodogram, be defined on the interval $(-\frac{1}{2}, \frac{1}{2}]$. In particular, we define the sample spectral density or sample spectrum as $\hat{S}(f) = \frac{1}{2}I(f)$ for all frequencies in $(-\frac{1}{2}, \frac{1}{2})$ and $\hat{S}(\frac{1}{2}) = I(\frac{1}{2})$. Using straightforward but somewhat tedious algebra, we can show that the sample spectral density can also be expressed as

$$\hat{S}(f) = \hat{\gamma}_0 + 2 \sum_{k=1}^{n-1} \hat{\gamma}_k \cos(2\pi fk) \quad (13.3.10)$$

where $\hat{\gamma}_k$ is the sample or estimated covariance function at lag k ($k = 0, 1, 2, \dots, n-1$) given by

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y}) \quad (13.3.11)$$

In Fourier analysis terms, the sample spectral density is the (discrete-time) Fourier transform of the sample covariance function. From Fourier analysis theory, it follows that there is an inverse relationship, namely[†]

$$\hat{\gamma}_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{S}(f) \cos(2\pi fk) df \quad (13.3.12)$$

[†] This may be proved using the orthogonality relationships shown in Appendix J on page 349.

In particular, notice that the total area under the sample spectral density is the sample variance of the time series.

$$\hat{\gamma}_0 = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{S}(f) df = \frac{1}{n} \sum_{t=1}^n (Y_t - \bar{Y})^2 \quad (13.3.13)$$

Since each can be obtained from the other, the sample spectral density and the sample covariance function contain the same information about the observed time series but it is expressed in different ways. For some purposes, one is more convenient or useful, and for other purposes the other is more convenient or useful.

13.4 The Spectral Density

For many processes, such as all stationary ARMA processes, the covariance functions decay rapidly with increasing lag.[†] When that is the case, it seems reasonable to consider the expression formed by replacing sample quantities in the sample spectral density of Equation (13.3.10) with the corresponding theoretical quantities. To be precise, if the covariance function γ_k is absolutely summable, we define the theoretical (or population) *spectral density* for $-\frac{1}{2} < f \leq \frac{1}{2}$ as

$$S(f) = \gamma_0 + 2 \sum_{k=1}^{\infty} \gamma_k \cos(2\pi fk) \quad (13.4.1)$$

Once more, there is an inverse relationship, given by

$$\gamma_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} S(f) \cos(2\pi fk) df \quad (13.4.2)$$

Mathematically, $S(f)$ is the (discrete-time) Fourier transform of the sequence $\dots, \gamma_{-2}, \gamma_{-1}, \gamma_0, \gamma_1, \gamma_2, \dots$, and $\{\gamma_k\}$ is the inverse Fourier transform[‡] of the spectral density $S(f)$ defined on $-\frac{1}{2} < f \leq \frac{1}{2}$.

A spectral density has all of the mathematical properties of a probability density function on the interval $(-\frac{1}{2}, \frac{1}{2}]$, with the exception that the total area is γ_0 rather than 1. Moreover, it can be shown that

[†] Of course, this is *not* the case for the processes defined in Equations (13.2.4) on page 323 and (13.3.1) on page 327. Those processes have discrete components in their spectra.

[‡] Notice that since $\gamma_k = \gamma_{-k}$ and the cosine function is also even, we could write

$$S(f) = \sum_{k=-\infty}^{\infty} \gamma_k e^{-2\pi ikf}$$

where $i = \sqrt{-1}$ is the imaginary unit for complex numbers. This looks more like a standard discrete-time Fourier transform. In a similar way, Equation (13.4.2) may be rewritten as

$$\gamma_k = \int_{-\frac{1}{2}}^{\frac{1}{2}} S(f) e^{2\pi i kf} df.$$

$$F(f) = \int_0^f S(x)dx \quad \text{for } 0 \leq f \leq \frac{1}{2} \quad (13.4.3)$$

Thus, twice the area under the spectral density between frequencies f_1 and f_2 with $0 \leq f_1 < f_2 \leq \frac{1}{2}$ is interpreted as the portion of the variance of the process that is attributable to cosine-sine pairs in that frequency interval that compose the process.

Time-Invariant Linear Filters

A *time-invariant linear filter* is defined by a sequence of absolutely summable constants $\dots, c_{-1}, c_0, c_1, c_2, c_3, \dots$. If $\{X_t\}$ is a time series, we use these constants to filter $\{X_t\}$ and produce a new time series $\{Y_t\}$ using the expression

$$Y_t = \sum_{j=-\infty}^{\infty} c_j X_{t-j} \quad (13.4.4)$$

If $c_k = 0$ for $k < 0$, we say that the filter is *causal*. In this case, the filtering at time t involves only present and past data values and can be carried out in “real time.”

We have already seen many examples of time-invariant linear filters in previous chapters. Differencing (nonseasonal or seasonal) is an example. A combination of one seasonal difference with one nonseasonal difference is another example. Any moving average process can be considered as a linear filtering of a white noise sequence and in fact every general linear process defined by Equation (4.1.1) on page 55 is a linear filtering of white noise.

The expression on the right-hand side of Equation (13.4.4) is frequently called the (discrete-time) *convolution* of the two sequences $\{c_t\}$ and $\{X_t\}$. An extremely useful property of Fourier transforms is that the somewhat complicated operation of convolution in the time domain is transformed into the very simple operation of multiplication in the frequency domain.[†]

In particular, let $S_X(f)$ be the spectral density for the $\{X_t\}$ process and let $S_Y(f)$ be the spectral density for the $\{Y_t\}$ process. In addition, let

$$C(e^{-2\pi if}) = \sum_{j=-\infty}^{\infty} c_j e^{-2\pi ifj} \quad (13.4.5)$$

Then

$$\begin{aligned} Cov(Y_t, Y_{t-k}) &= Cov\left(\sum_{j=-\infty}^{\infty} c_j X_{t-j}, \sum_{s=-\infty}^{\infty} c_s X_{t-k-s}\right) \\ &= \sum_{j=-\infty}^{\infty} \sum_{s=-\infty}^{\infty} c_j c_s Cov(X_{t-j}, X_{t-k-s}) \end{aligned}$$

[†] You may have already seen this with moment-generating functions. The density of the sum of two independent random variables, discrete or continuous, is the convolution of their respective densities, but the moment-generating function for the sum is the product of their respective moment-generating functions.

$$\begin{aligned}
&= \sum_{j=-\infty}^{\infty} \sum_{s=-\infty}^{\infty} c_j c_s \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{2\pi i(s+k-j)f} S_X(f) df \\
&= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left| \sum_{s=-\infty}^{\infty} c_s e^{-2\pi i sf} \right|^2 e^{2\pi i fk} S_X(f) df
\end{aligned}$$

So

$$Cov(Y_t, Y_{t-k}) = \int_{-\frac{1}{2}}^{\frac{1}{2}} |C(e^{-2\pi if})|^2 S_X(f) e^{2\pi i fk} df \quad (13.4.6)$$

But

$$Cov(Y_t, Y_{t-k}) = \int_{-\frac{1}{2}}^{\frac{1}{2}} S_Y(f) e^{2\pi i fk} df \quad (13.4.7)$$

so we must have

$$S_Y(f) = |C(e^{-2\pi if})|^2 S_X(f) \quad (13.4.8)$$

This expression is invaluable for investigating the effect of time-invariant linear filters on spectra. In particular, it helps us find the form of the spectral densities for ARMA processes. The function $|C(e^{-2\pi if})|^2$ is often called the (*power*) *transfer function* of the filter.

13.5 Spectral Densities for ARMA Processes

White Noise

From Equation (13.4.1), it is easy to see that the theoretical spectral density for a white noise process is constant for all frequencies in $-\frac{1}{2} < f \leq \frac{1}{2}$ and, in particular,

$$S(f) = \sigma_e^2 \quad (13.5.1)$$

All frequencies receive equal weight in the spectral representation of white noise. This is directly analogous to the spectrum of white light in physics—all colors (that is, all frequencies) enter equally in white light. Finally, we understand the origin of the name white noise!

MA(1) Spectral Density

An MA(1) process is a simple filtering of white noise with $c_0 = 1$ and $c_1 = -\theta$ and so

$$\begin{aligned}
|C(e^{-2\pi if})|^2 &= (1 - \theta e^{2\pi if})(1 - \theta e^{-2\pi if}) \\
&= 1 + \theta^2 - \theta(e^{2\pi if} + e^{-2\pi if}) \\
&= 1 + \theta^2 - 2\theta \cos(2\pi f)
\end{aligned} \quad (13.5.2)$$

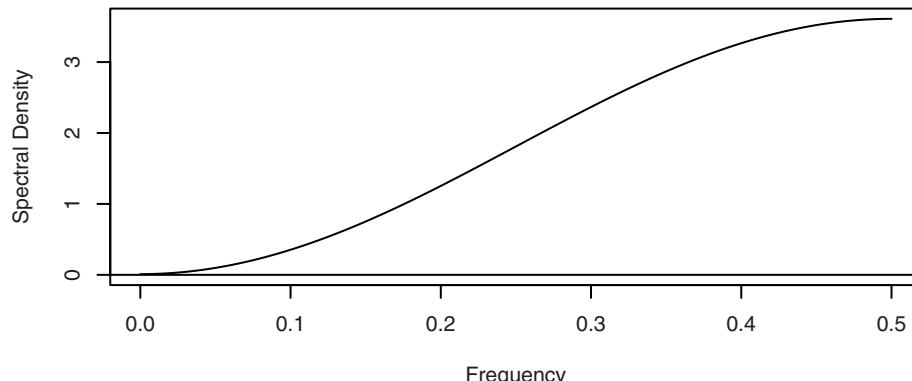
Thus

$$S(f) = [1 + \theta^2 - 2\theta \cos(2\pi f)]\sigma_e^2 \quad (13.5.3)$$

When $\theta > 0$, you can show that this spectral density is an increasing function of nonnegative frequency, while for $\theta < 0$ the function decreases.

Exhibit 13.9 displays the spectral density for an MA(1) process with $\theta = 0.9$.[†] Since spectral densities are symmetric about zero frequency, we will only plot them for positive frequencies. Recall that this MA(1) process has a relatively large negative correlation at lag 1 but all other correlations are zero. This is reflected in the spectrum. We see that the density is much stronger for higher frequencies than for low frequencies. The process has a tendency to oscillate back and forth across its mean level. This rapid oscillation is high-frequency behavior. We might say that the moving average suppresses the lower-frequency components of the white noise process. Researchers sometimes refer to this type of spectrum as a *blue spectrum* since it emphasizes the higher frequencies (that is, those with lower period or wavelength), which correspond to blue light in the spectrum of visible light.

Exhibit 13.9 Spectral Density of MA(1) Process with $\theta = 0.9$

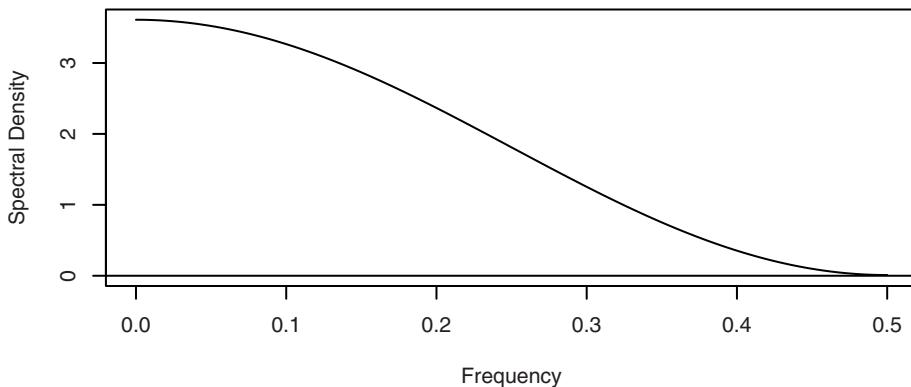


```
> win.graph(width=4.875, height=2.5, pointsize=8)
> theta=.9 # Reset theta for other MA(1) plots
> ARMAspec(model=list(ma=-theta))
```

Exhibit 13.10 displays the spectral density for an MA(1) process with $\theta = -0.9$. This process has positive correlation at lag 1 with all other correlations zero. Such a process will tend to change slowly from one time instance to the next. This is low-frequency behavior and is reflected in the shape of the spectrum. The density is much stronger for lower frequencies than for high frequencies. Researchers sometimes call this a *red spectrum*.

[†] In all of the plots of ARMA spectral densities that follow in this section, we take $\sigma_e^2 = 1$. This only affects the vertical scale of the graphs, not their shape.

Exhibit 13.10 Spectral Density of MA(1) Process with $\theta = -0.9$

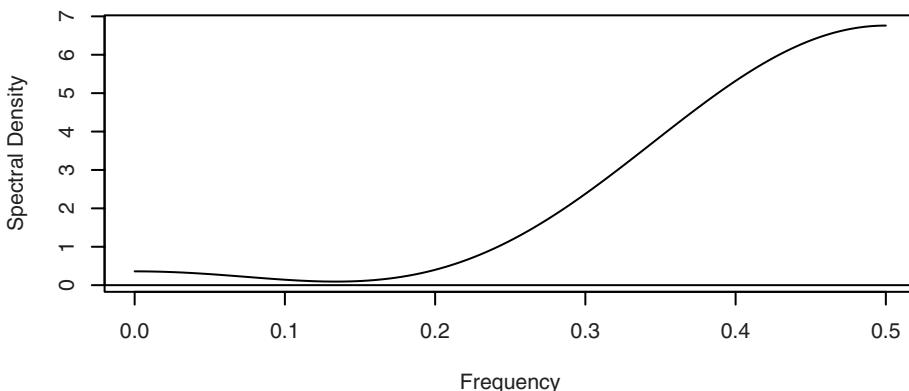
**MA(2) Spectral Density**

The spectral density for an MA(2) model may be obtained similarly. The algebra is a little longer, but the final expression is

$$S(f) = [1 + \theta_1^2 + \theta_2^2 - 2\theta_1(1 - \theta_2)\cos(2\pi f) - 2\theta_2\cos(4\pi f)]\sigma_e^2 \quad (13.5.4)$$

Exhibit 13.11 shows a graph of such a density when $\theta_1 = 1$ and $\theta_2 = -0.6$. The frequencies between about 0.1 and 0.18 have especially small density and there is very little density below the frequency of 0.1. Higher frequencies enter into the picture gradually, with the strongest periodic components at the highest frequencies.

Exhibit 13.11 Spectral Density of MA(2) Process with $\theta_1 = 1$ and $\theta_2 = -0.6$



```
> theta1=1; theta2=-0.6
> ARMAspec(model=list(ma=c(theta1,theta2)))
```

AR(1) Spectral Density

To find the spectral density for AR models, we use Equation (13.4.8) “backwards.” That is, we view the white noise process as being a linear filtering of the AR process. Recalling the spectral density of the MA(1) series, this gives

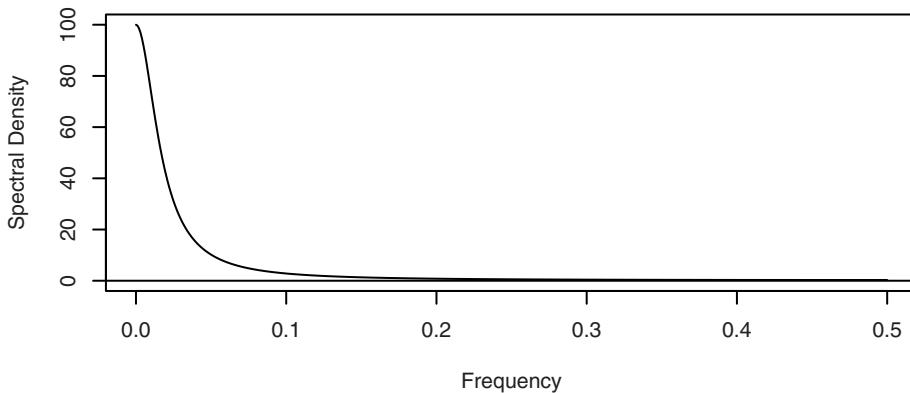
$$[1 + \phi^2 - 2\phi \cos(2\pi f)]S(f) = \sigma_e^2 \quad (13.5.5)$$

which we solve to obtain

$$S(f) = \frac{\sigma_e^2}{1 + \phi^2 - 2\phi \cos(2\pi f)} \quad (13.5.6)$$

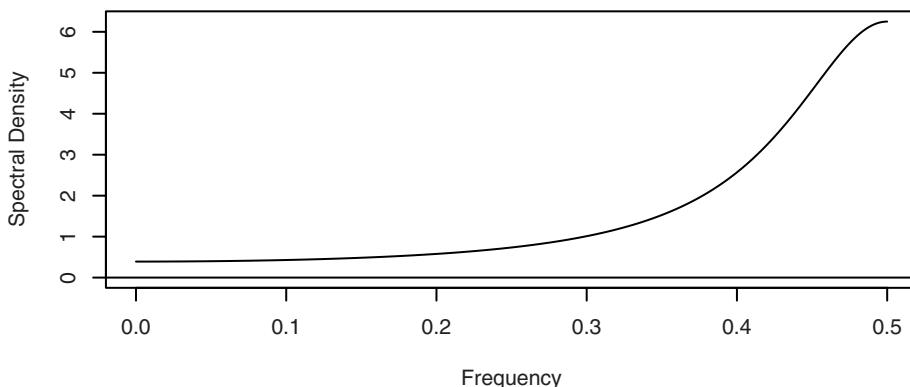
As the next two exhibits illustrate, this spectral density is a decreasing function of frequency when $\phi > 0$, while the spectral density increases for $\phi < 0$.

Exhibit 13.12 Spectral Density of an AR(1) Process with $\phi = 0.9$



```
> phi=0.9 # Reset value of phi for other AR(1) models
> ARMAspec(model=list(ar=phi))
```

Exhibit 13.13 Spectral Density of an AR(1) Process with $\phi = -0.6$



AR(2) Spectral Density

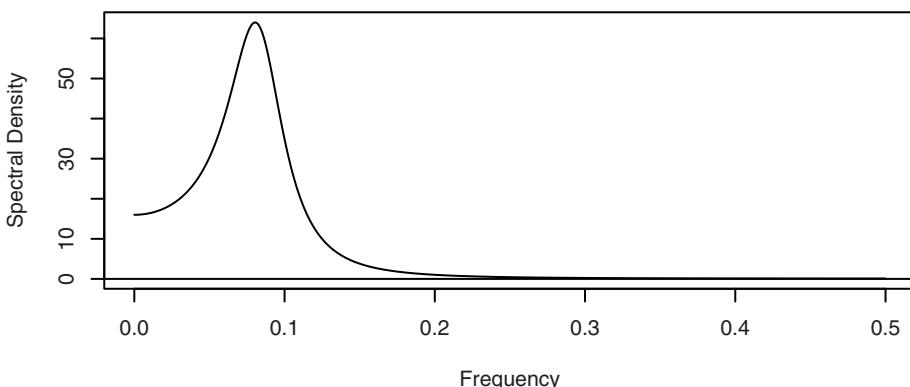
For the AR(2) spectral density, we again use Equation (13.4.8) backwards together with the MA(2) result to obtain

$$S(f) = \frac{\sigma_e^2}{1 + \phi_1^2 + \phi_2^2 - 2\phi_1(1 - \phi_2)\cos(2\pi f) - 2\phi_2\cos(4\pi f)} \quad (13.5.7)$$

Just as with the correlation properties, the spectral density for an AR(2) model can exhibit a variety of behaviors depending on the actual values of the two ϕ parameters.

Exhibits 13.14 and 13.15 display two AR(2) spectral densities that show very different behavior of peak in one case and trough in another.

Exhibit 13.14 Spectral Density of AR(2) Process: $\phi_1 = 1.5$ and $\phi_2 = -0.75$



```
> phi1=1.5; phi2=-.75
> # Reset values of phi1 & phi2 for other AR(2) models
> ARMAspec(model=list(ar=c(phi1,phi2)))
```

Jenkins and Watts (1968, p. 229), have noted that the different spectral shapes for an AR(2) spectrum are determined by the inequality

$$|\phi_1(1 - \phi_2)| < |4\phi_2| \quad (13.5.8)$$

and the results are best summarized in the display in Exhibit 13.16. In this display, the dashed curve is the border between the regions of real roots and complex roots of the AR(2) characteristic equation. The solid curves are determined from the inequality given in Equation (13.5.8).

Exhibit 13.15 Spectral Density of AR(2) Process with $\phi_1 = 0.1$ and $\phi_2 = 0.4$

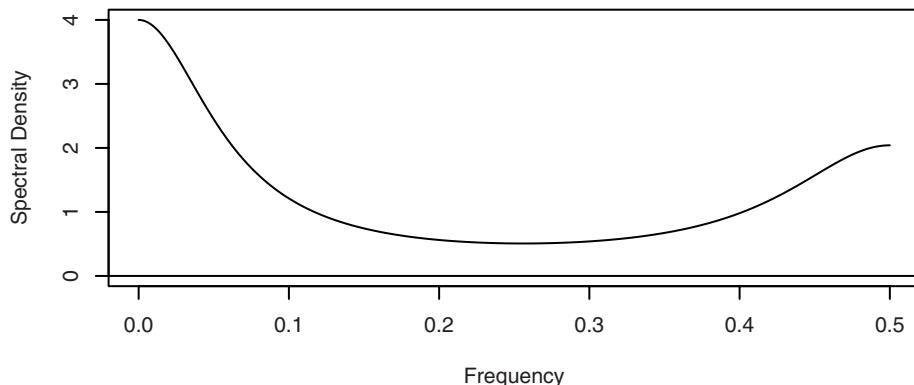
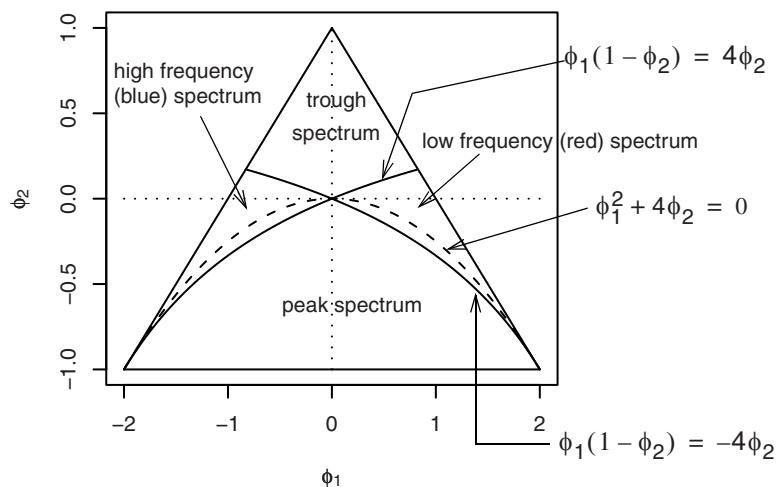


Exhibit 13.16 AR(2) Parameter Values for Various Spectral Density Shapes



Note that Jenkins and Watts also showed that the frequency f_0 at which the peak or trough occurs will satisfy

$$\cos(2\pi f_0) = -\frac{\phi_1(1-\phi_2)}{4\phi_2} \quad (13.5.9)$$

It is commonly thought that complex roots are associated with a peak spectrum. But notice that there is a small region of parameter values where the roots are complex but the spectrum is of either high or low frequency with no intermediate peak.

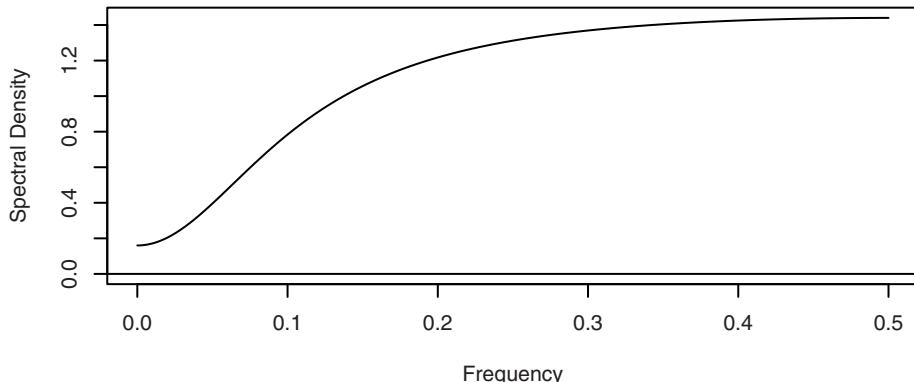
ARMA(1,1) Spectral Density

Combining what we know for MA(1) and AR(1) models, we can easily obtain the spectral density for the ARMA(1,1) mixed model

$$S(f) = \frac{1 + \theta^2 - 2\theta \cos(2\pi f)}{1 + \phi^2 - 2\phi \cos(2\pi f)} \sigma_e^2 \quad (13.5.10)$$

Exhibit 13.17 provides an example of the spectrum for an ARMA(1,1) model with $\phi = 0.5$ and $\theta = 0.8$.

Exhibit 13.17 Spectral Density of ARMA(1,1) with $\phi = 0.5$ and $\theta = 0.8$



```
> phi=0.5; theta=0.8
> ARMAspec(model=list(ar=phi,ma=-theta))
```

ARMA(p, q)

For the general ARMA(p, q) case, the spectral density may be expressed in terms of the AR and MA characteristic polynomials as

$$S(f) = \left| \frac{\theta(e^{-2\pi if})}{\phi(e^{-2\pi if})} \right|^2 \sigma_e^2 \quad (13.5.11)$$

This may be further expressed in terms of the reciprocal roots of these polynomials, but we will not pursue those expressions here. This type of spectral density is often referred to as a rational spectral density.

Seasonal ARMA Processes

Since seasonal ARMA processes are just special ARMA processes, all of our previous work will carry over here. Multiplicative seasonal models can be thought of as applying two linear filters consecutively. We will just give two examples.

Consider the process defined by the seasonal AR model

$$(1 - \phi B)(1 - \Phi B^{12})Y_t = e_t \quad (13.5.12)$$

Manipulating the two factors separately yields

$$S(f) = \frac{\sigma_e^2}{[1 + \phi^2 - 2\phi \cos(2\pi f)][1 + \Phi^2 - 2\Phi \cos(2\pi 12f)]} \quad (13.5.13)$$

An example of this spectrum is shown in Exhibit 13.18, where $\phi = 0.5$, $\Phi = 0.9$, and $s = 12$. The seasonality is reflected in the many spikes of decreasing magnitude at frequencies of $0, 1/12, 2/12, 3/12, 4/12, 5/12$, and $6/12$.

As a second example, consider a seasonal MA process

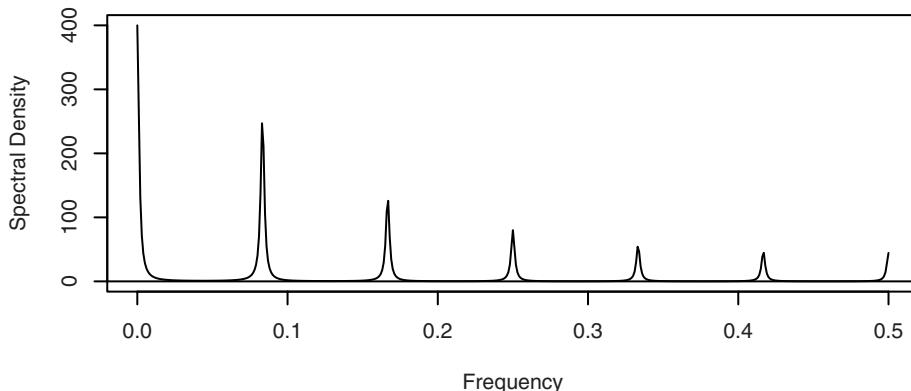
$$Y_t = (1 - \theta B)(1 - \Theta B^{12})e_t \quad (13.5.14)$$

The corresponding spectral density is given by

$$S(f) = [1 + \theta^2 - 2\theta \cos(2\pi f)][1 + \Theta^2 - 2\Theta \cos(2\pi 12f)]\sigma_e^2 \quad (13.5.15)$$

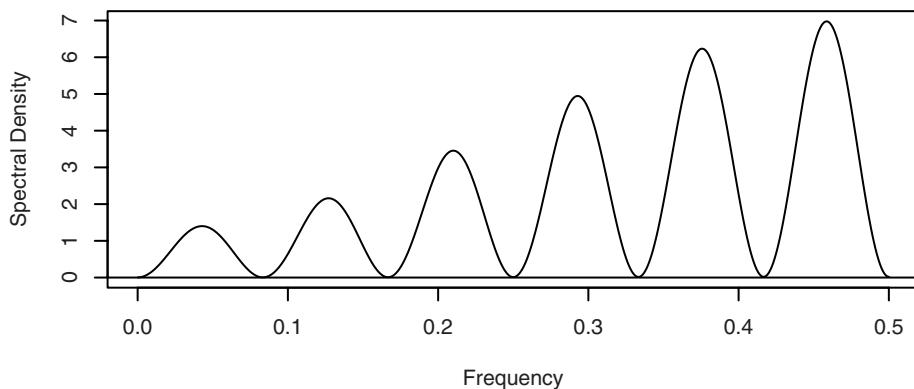
Exhibit 13.19 shows this spectral density for parameter values $\theta = 0.4$ and $\Theta = 0.9$.

Exhibit 13.18 Spectral Density of Seasonal AR with $\phi = 0.5$, $\Phi = 0.9$, $s = 12$



```
> phi=.5; PHI=.9
> ARMAspec(model=list(ar=phi, seasonal=list(sar=PHI, period=12)))
```

Exhibit 13.19 Spectral Density of Seasonal MA with $\theta = 0.4$, $\Theta = 0.9$, $s=12$



```
> theta=.4; Theta=.9
> ARMAAspec(model=list(ma=-theta, seasonal=list(sma=-Theta,
  period=12)))
```

13.6 Sampling Properties of the Sample Spectral Density

To introduce this section, we consider a time series with known properties. Suppose that we simulate an AR(1) model with $\phi = -0.6$ of length $n = 200$. Exhibit 13.13 on page 336, shows the theoretical spectral density for such a series. The sample spectral density for our simulated series is displayed in Exhibit 13.20, with the smooth theoretical spectral density shown as a dotted line. Even with a sample of size 200, the sample spectral density is extremely variable from one frequency point to the next. This is surely not an acceptable estimate of the theoretical spectrum for this process. We must investigate the sampling properties of the sample spectral density to understand the behavior that we see here.

To investigate the sampling properties of the sample spectral density, we begin with the simplest case, where the time series $\{Y_t\}$ is zero-mean normal white noise with variance γ_0 . Recall that

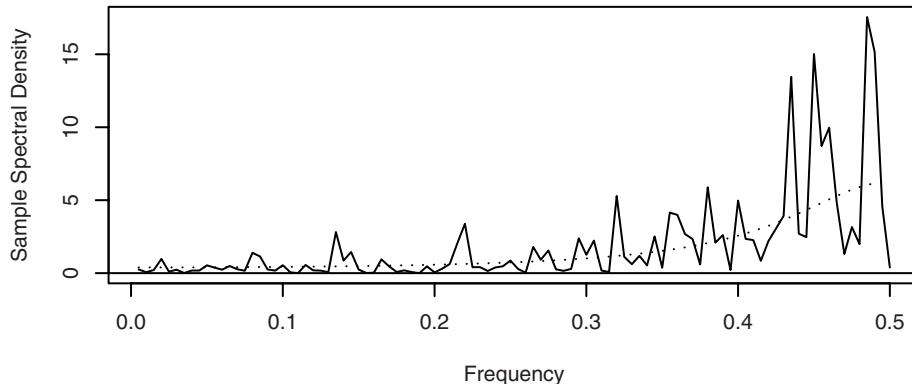
$$\hat{A}_f = \frac{2}{n} \sum_{t=1}^n Y_t \cos(2\pi t f) \quad \text{and} \quad \hat{B}_f = \frac{2}{n} \sum_{t=1}^n Y_t \sin(2\pi t f) \quad (13.6.1)$$

For now, consider only nonzero Fourier frequencies $f = j/n < 1/2$. Since \hat{A}_f and \hat{B}_f are linear functions of the time series $\{Y_t\}$, they each have a normal distribution. We can evaluate the means and variances using the orthogonality properties of the cosines and sines.[†] We find that \hat{A}_f and \hat{B}_f each have mean zero and variance $2\gamma_0/n$. We can also use the orthogonality properties to show that \hat{A}_f and \hat{B}_f are uncorrelated and thus indepen-

[†] See Appendix J on page 349.

dent since they are jointly bivariate normal. Similarly, it can be shown that for any two distinct Fourier frequencies f_1 and f_2 , \hat{A}_{f_1} , \hat{A}_{f_2} , \hat{B}_{f_1} , and \hat{B}_{f_2} are jointly independent.

Exhibit 13.20 Sample Spectral Density for a Simulated AR(1) Process



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> set.seed(271435); n=200; phi=-0.6
> y=arima.sim(model=list(ar=phi), n=n)
> sp=spec(y, log='no', xlab='Frequency',
+           ylab='Sample Spectral Density', sub=' ')
> lines(sp$freq, ARMAspec(model=list(ar=phi), freq=sp$freq,
+                           plot=F)$spec, lty='dotted'); abline(h=0)
```

Furthermore, we know that the square of a standard normal has a chi-square distribution with one degree of freedom and that the sum of independent chi-square variables is chi-square distributed with degrees of freedom added together. Since $S(f) = \gamma_0$, we have

$$\frac{n}{2\gamma_0} [(\hat{A}_f)^2 + (\hat{B}_f)^2] = \frac{2\hat{S}(f)}{S(f)} \quad (13.6.2)$$

has a chi-square distribution with two degrees of freedom.

Recall that a chi-square variable has a mean equal to its degrees of freedom and a variance equal to twice its degrees of freedom. With these facts, we quickly discover that

$$\hat{S}(f_1) \text{ and } \hat{S}(f_2) \text{ are independent for } f_1 \neq f_2 \quad (13.6.3)$$

$$E[\hat{S}(f)] = S(f) \quad (13.6.4)$$

and

$$Var[\hat{S}(f)] = S^2(f) \quad (13.6.5)$$

Equation (13.6.4) expresses the desirable fact that the sample spectral density is an unbiased estimator of the theoretical spectral density.

Unfortunately, Equation (13.6.5) shows that the variance in no way depends on the sample size n . Even in this simple case, *the sample spectral density is not a consistent estimator of the theoretical spectral density*. It does not get better (that is, have smaller variance) as the sample size increases. The reason the sample spectral density is inconsistent is basically this: Even if we only consider Fourier frequencies, $1/n, 2/n, \dots$, we are trying to estimate more and more “parameters”; that is, $S(1/n), S(2/n), \dots$. As the sample size increases, there are not enough data points per parameter to produce consistent estimates.

The results expressed in Equations (13.6.3)–(13.6.5) in fact hold more generally. In the exercises, we ask you to argue that for any white noise—not necessarily normal—the mean result holds exactly and the \hat{A}_f and \hat{B}_f that make up $\hat{S}(f_1)$ and $\hat{S}(f_2)$ are at least uncorrelated for $f_1 \neq f_2$.

To state more general results, suppose $\{Y_t\}$ is any linear process

$$Y_t = e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots \quad (13.6.6)$$

where the e 's are independent and identically distributed with zero mean and common variance. Suppose that the ψ -coefficients are absolutely summable, and let $f_1 \neq f_2$ be any frequencies in 0 to $1/2$. Then it may be shown[†] that as the sample size increases without limit

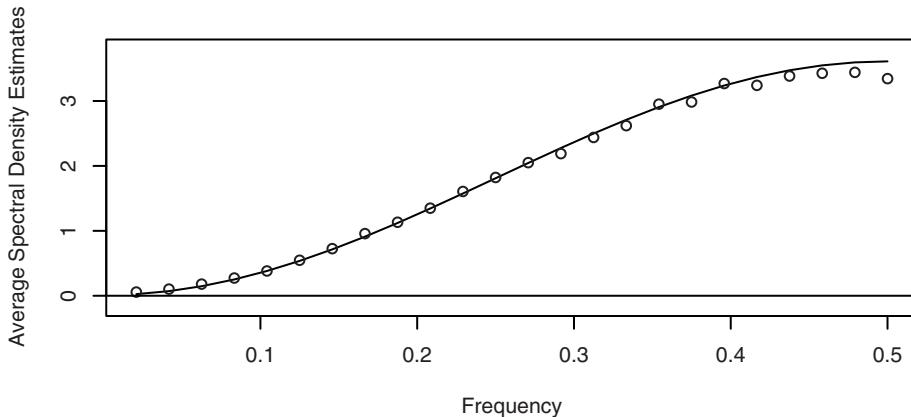
$$\frac{2\hat{S}(f_1)}{S(f_1)} \text{ and } \frac{2\hat{S}(f_2)}{S(f_2)} \quad (13.6.7)$$

converge in distribution to independent chi-square random variables, each with two degrees of freedom.

To investigate the usefulness of approximations based on Equations (13.6.7), (13.6.4), and (13.6.5), we will display results from two simulations. We first simulated 1000 replications of an MA(1) time series with $\theta = 0.9$, each of length $n = 48$. The white noise series used to create each MA(1) series was selected independently from a t -distribution with five degrees of freedom scaled to unit variance. From the 1000 series, we calculated 1000 sample spectral densities.

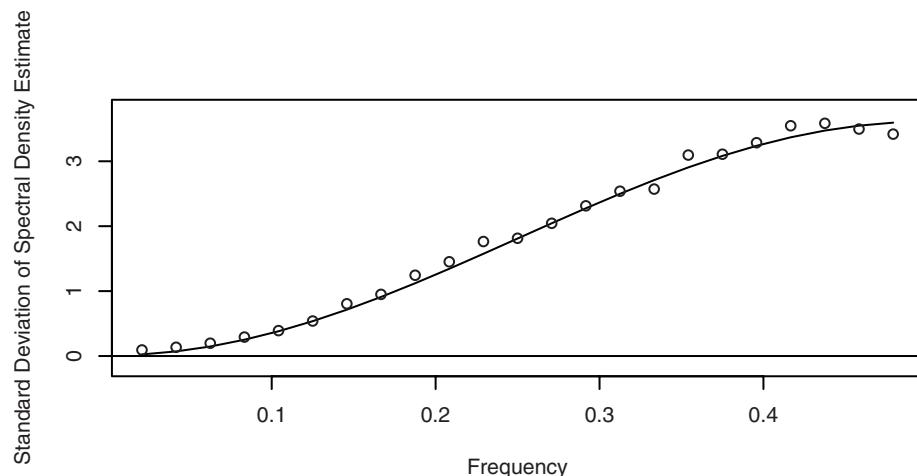
Exhibit 13.21 shows the average of the 1000 sample spectral densities evaluated at the 24 Fourier frequencies associated with $n = 48$. The solid line is the theoretical spectral density. It appears that the sample spectral densities are unbiased to a useful approximation in this case.

[†] See, for example, Fuller (1996, pp. 360–361).

**Exhibit 13.21 Average Sample Spectral Density:
Simulated MA(1), $\theta = 0.9$, $n = 48$**


For the extensive R code to produce Exhibits 13.21 through 13.26, please see the Chapter 13 script file associated with this book.

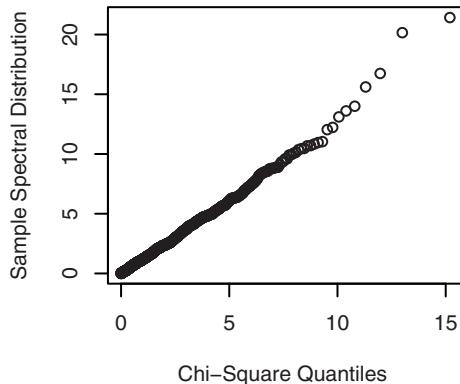
Exhibit 13.22 plots the standard deviations of the sample spectral densities over the 1000 replications. According to Equation (13.6.5), we hope that they match the theoretical spectral density at the Fourier frequencies. Again the approximation seems to be quite acceptable.

**Exhibit 13.22 Standard Deviation of Sample Spectral Density:
Simulated MA(1), $\theta = 0.9$, $n = 48$**


To check on the shape of the sample spectral density distribution, we constructed a QQ plot comparing the observed quantiles with those of a chi-square distribution with

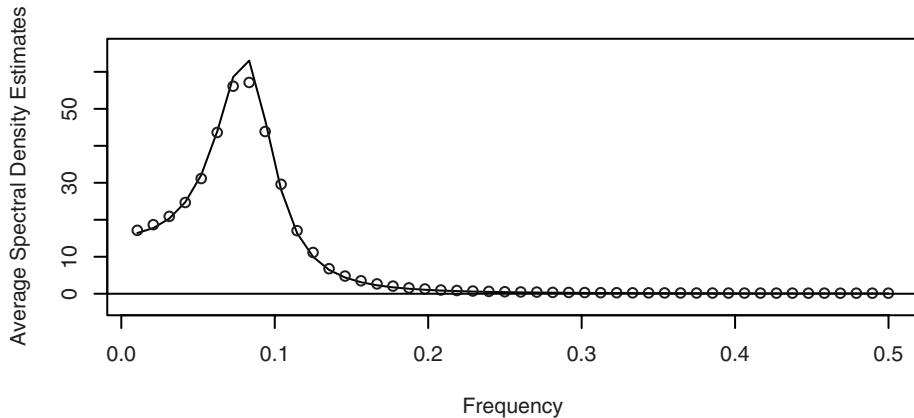
two degrees of freedom. Of course, we could do those for any of the Fourier frequencies. Exhibit 13.23 shows the results at the frequency $15/48$. The agreement with the chi-square distribution appears to be acceptable.

Exhibit 13.23 QQ Plot of Spectral Distribution at $f = 15/48$



We repeated similar displays and calculations when the true model was an AR(2) with $\phi_1 = 1.5$, $\phi_2 = -0.75$, and $n = 96$. Here we used normal white noise. The results are displayed in Exhibits 13.24, 13.25, and 13.26. Once more the simulation results with $n = 96$ and 1000 replications seem to follow those suggested by limit theory quite remarkably.

**Exhibit 13.24 Average Sample Spectral Density:
Simulated AR(2), $\phi_1 = 1.5$, $\phi_2 = -0.75$, $n = 96$**



**Exhibit 13.25 Standard Deviation of Sample Spectral Density:
Simulated AR(2), $\phi_1 = 1.5$, $\phi_2 = -0.75$, $n = 96$**

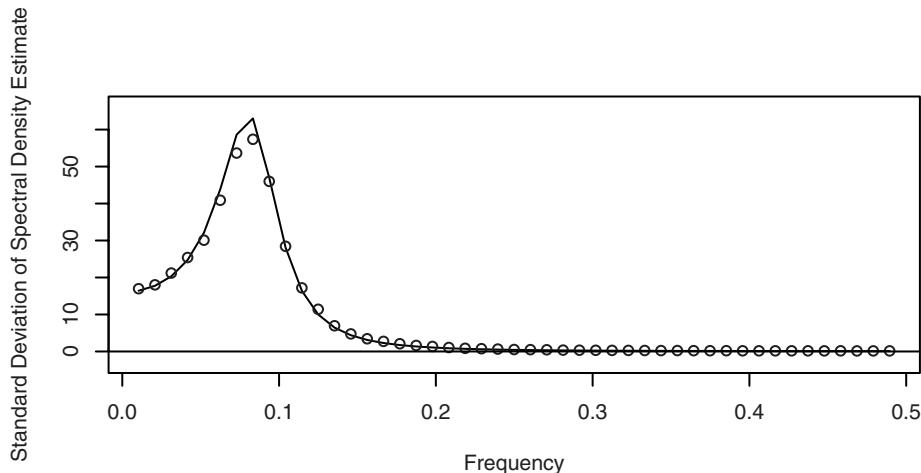
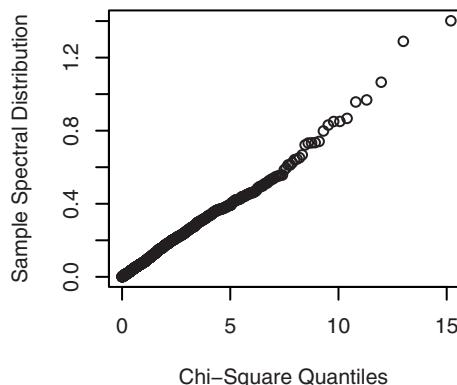


Exhibit 13.26 QQ Plot of Spectral Distribution at $f = 40/96$



Of course, none of these results tell us that the sample spectral density is an acceptable estimator of the underlying theoretical spectral density. The sample spectral density is quite generally approximately unbiased but also inconsistent, with way too much variability to be a useful estimator as it stands. The approximate independence at the Fourier frequencies also helps explain the extreme variability in the behavior of the sample spectral density.

13.7 Summary

The chapter introduces the ideas of modeling time series as linear combinations of sines and cosines—so-called spectral analysis. The periodogram was introduced as a tool for finding the contribution of the various frequencies in the spectral representation of the series. The ideas were then extended to modeling with a continuous range of frequencies. Spectral densities of the ARMA models were explored. Finally, the sampling properties of the sample spectral density were presented. Since the sample spectral density is not a consistent estimator of the theoretical spectral density, we must search further for an acceptable estimator. That is the subject of the next chapter.

EXERCISES

- 13.1** Find A and B so that $3\cos(2\pi ft + 0.4) = A\cos(2\pi ft) + B\sin(2\pi ft)$.
- 13.2** Find R and Φ so that $R\cos(2\pi ft + \Phi) = \cos(2\pi ft) + 3\sin(2\pi ft)$.
- 13.3** Consider the series displayed in Exhibit 13.2 on page 320.
 - (a) Verify that regressing the series on $\cos(2\pi ft)$ and $\sin(2\pi ft)$ for $f = 4/96$ provides perfect estimates of A and B .
 - (b) Use Equations (13.1.5) on page 321 to obtain the relationship between R , Φ , A and B for the cosine component at frequency $f = 14/96$. (For this component, the amplitude is 1 and the phase is 0.6π .)
 - (c) Verify that regressing the series on $\cos(2\pi ft)$ and $\sin(2\pi ft)$ for $f = 14/96$ provides perfect estimates of A and B .
 - (d) Verify that regressing the series on $\cos(2\pi ft)$ and $\sin(2\pi ft)$ for both $f = 4/96$ and $f = 14/96$ together provides perfect estimates of A_4 , B_4 , A_{14} , and B_{14} .
 - (e) Verify that regressing the series on $\cos(2\pi ft)$ and $\sin(2\pi ft)$ for $f = 3/96$ and $f = 13/96$ together provides perfect estimates of A_3 , B_3 , A_{13} , and B_{13} .
 - (f) Repeat part (d) but add a third pair of cosine-sine predictor variables at any other Fourier frequency. Verify that all of the regression coefficients are still estimated perfectly.
- 13.4** Generate or choose *any* series of length $n = 10$. Show that the series may be fit exactly by a linear combination of enough cosine-sine curves at the Fourier frequencies.
- 13.5** Simulate a signal + noise time series from the model in Equation (13.2.4) on page 323. Use the same parameter values used in Exhibit 13.4 on page 324.
 - (a) Plot the time series and look for the periodicities. Can you see them?
 - (b) Plot the periodogram for the simulated series. Are the periodicities clear now?
- 13.6** Show that the covariance function for the series defined by Equation (13.3.1) on page 327 is given by the expression in Equation (13.3.2).
- 13.7** Display the algebra that establishes Equation (13.3.10) on page 329.

- 13.8** Show that if $\{X_t\}$ and $\{Y_t\}$ are independent stationary series, then the spectral density of $\{X_t + Y_t\}$ is the sum of the spectral densities of $\{X_t\}$ and $\{Y_t\}$.
- 13.9** Show that when $\theta > 0$ the spectral density for an MA(1) process is an increasing function of frequency, while for $\theta < 0$ this function decreases.
- 13.10** Graph the theoretical spectral density for an MA(1) process with $\theta = 0.6$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.11** Graph the theoretical spectral density for an MA(1) process with $\theta = -0.8$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.12** Show that when $\phi > 0$ the spectral density for an AR(1) process is a decreasing function of frequency, while for $\phi < 0$ the spectral density increases.
- 13.13** Graph the theoretical spectral density for an AR(1) time series with $\phi = 0.7$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.14** Graph the theoretical spectral density for an AR(1) time series with $\phi = -0.4$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.15** Graph the theoretical spectral density for an MA(2) time series with $\theta_1 = -0.5$ and $\theta_2 = 0.9$. Interpret the implications of the shape of the spectrum on the possible time series plots of the series values.
- 13.16** Graph the theoretical spectral density for an MA(2) time series with $\theta_1 = 0.5$ and $\theta_2 = -0.9$. Interpret the implications of the shape of the spectrum on the possible time series plots of the series values.
- 13.17** Graph the theoretical spectral density for an AR(2) time series with $\phi_1 = -0.1$ and $\phi_2 = -0.9$. Interpret the implications of the shape of the spectrum on the possible time series plots of the series values.
- 13.18** Graph the theoretical spectral density for an AR(2) process with $\phi_1 = 1.8$ and $\phi_2 = -0.9$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.19** Graph the theoretical spectral density for an AR(2) process with $\phi_1 = -1$ and $\phi_2 = -0.8$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.20** Graph the theoretical spectral density for an AR(2) process with $\phi_1 = 0.5$ and $\phi_2 = 0.4$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.21** Graph the theoretical spectral density for an AR(2) process with $\phi_1 = 0$ and $\phi_2 = 0.8$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.22** Graph the theoretical spectral density for an AR(2) process with $\phi_1 = 0.8$ and $\phi_2 = -0.2$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.23** Graph the theoretical spectral density for an ARMA(1,1) time series with $\phi = 0.5$ and $\theta = 0.8$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.

- 13.24** Graph the theoretical spectral density for an ARMA(1,1) process with $\phi = 0.95$ and $\theta = 0.8$. Interpret the implications of the shape of the spectrum on the possible plots of the time series values.
- 13.25** Let $\{X_t\}$ be a stationary time series and $\{Y_t\}$ be defined by $Y_t = (X_t + X_{t-1})/2$.
- Find the power transfer function for this linear filter.
 - Is this a causal filter?
 - Graph the power transfer function and describe the effect of using this filter.
That is, what frequencies will be retained (emphasized) and what frequencies will be deemphasized (attenuated) by this filtering?
- 13.26** Let $\{X_t\}$ be a stationary time series and let $\{Y_t\}$ be defined by $Y_t = X_t - X_{t-1}$.
- Find the power transfer function for this linear filter.
 - Is this a causal filter?
 - Graph the power transfer function and describe the effect of using this filter.
That is, what frequencies will be retained (emphasized) and what frequencies will be deemphasized (attenuated) by this filtering?
- 13.27** Let $\{X_t\}$ be a stationary time series and let $Y_t = (X_{t+1} + X_t + X_{t-1})/3$ define $\{Y_t\}$.
- Find the power transfer function for this linear filter.
 - Is this a causal filter?
 - Graph the power transfer function and describe the effect of using this filter.
That is, what frequencies will be retained (emphasized) and what frequencies will be deemphasized (attenuated) by this filtering?
- 13.28** Let $\{X_t\}$ be a stationary time series and let $Y_t = (X_t + X_{t-1} + X_{t-2})/3$ define $\{Y_t\}$.
- Show that the power transfer function of this filter is the same as the power transfer function of the filter defined in Exercise 13.27.
 - Is this a causal filter?
- 13.29** Let $\{X_t\}$ be a stationary time series and let $Y_t = X_t - X_{t-1}$ define $\{Y_t\}$.
- Find the power transfer function for this linear filter.
 - Graph the power transfer function and describe the effect of using this filter.
That is, what frequencies will be retained (emphasized) and what frequencies will be deemphasized (attenuated) by this filtering?
- 13.30** Let $\{X_t\}$ be a stationary time series and let $\{Y_t\}$ be defined by $Y_t = (X_{t+1} - 2X_t + X_{t-1})/3$.
- Find the power transfer function for this linear filter.
 - Graph the power transfer function and describe the effect of using this filter.
That is, what frequencies will be retained (emphasized) and what frequencies will be deemphasized (attenuated) by this filtering?

- 13.31** Suppose that $\{Y_t\}$ is a white noise process not necessarily normal. Use the orthogonality properties given in Appendix J to establish the following at the Fourier frequencies.
- (a) The sample spectral density is an unbiased estimator of the theoretical spectral density.
 - (b) The variables \hat{A}_{f_1} and \hat{B}_{f_2} are uncorrelated for any Fourier frequencies f_1, f_2 .
 - (c) If the Fourier frequencies $f_1 \neq f_2$, the variables \hat{A}_{f_1} and \hat{A}_{f_2} are uncorrelated.
- 13.32** Carry out a simulation analysis similar to those reported in Exhibits 13.21, 13.22, 13.23, and 13.24. Use an AR(2) model with $\phi_1 = 0.5$, $\phi_2 = 0.8$, and $n = 48$. Replicate the series 1000 times.
- (a) Display the average sample spectral density by frequency and compare it with large sample theory.
 - (b) Display the standard deviation of the sample spectral density by frequency and compare it with large sample theory.
 - (c) Display the QQ plot of the appropriately scaled sample spectral density compared with large sample theory at several frequencies. Discuss your results.
- 13.33** Carry out a simulation analysis similar to those reported in Exhibits 13.21, 13.22, 13.23, and 13.24. Use an AR(2) model with $\phi_1 = -1$, $\phi_2 = -0.75$, and $n = 96$. Replicate the time series 1000 times.
- (a) Display the average sample spectral density by frequency and compare it with the results predicted by large sample theory.
 - (b) Display the standard deviation of the sample spectral density by frequency and compare it with the results predicted by large sample theory.
 - (c) Display the QQ plot of the appropriately scaled sample spectral density and compare with the results predicted by large sample theory at several frequencies. Discuss your results.
- 13.34** Simulate a zero-mean, unit-variance, normal white noise time series of length $n = 1000$. Display the periodogram of the series, and comment on the results.

Appendix J: Orthogonality of Cosine and Sine Sequences

For $j, k = 0, 1, 2, \dots, n/2$, we have

$$\sum_{t=1}^n \cos\left(2\pi \frac{j}{n} t\right) = 0 \quad \text{if } j \neq 0 \quad (13.J.1)$$

$$\sum_{t=1}^n \sin\left(2\pi \frac{j}{n} t\right) = 0 \quad (13.J.2)$$

$$\sum_{t=1}^n \cos\left(2\pi \frac{j}{n} t\right) \sin\left(2\pi \frac{k}{n} t\right) = 0 \quad (13.J.3)$$

$$\sum_{t=1}^n \cos\left(2\pi\frac{j}{n}t\right) \cos\left(2\pi\frac{k}{n}t\right) = \begin{cases} \frac{n}{2} & \text{if } j = k \text{ (} j \neq 0 \text{ or } n/2 \text{)} \\ n & \text{if } j = k = 0 \\ 0 & \text{if } j \neq k \end{cases} \quad (13.J.4)$$

$$\sum_{t=1}^n \sin\left(2\pi\frac{j}{n}t\right) \sin\left(2\pi\frac{k}{n}t\right) = \begin{cases} \frac{n}{2} & \text{if } j = k \text{ (} j \neq 0 \text{ or } n/2 \text{)} \\ 0 & \text{if } j \neq k \end{cases} \quad (13.J.5)$$

These are most easily proved using DeMoivre's theorem

$$e^{-2\pi if} = \cos(2\pi f) - i \sin(2\pi f) \quad (13.J.6)$$

or, equivalently, Euler's formulas,

$$\cos(2\pi f) = \frac{e^{2\pi if} + e^{-2\pi if}}{2} \text{ and } \sin(2\pi f) = \frac{e^{2\pi if} - e^{-2\pi if}}{2i} \quad (13.J.7)$$

together with the result for the sum of a finite geometric series, namely

$$\sum_{j=1}^n r^j = \frac{r(1-r^n)}{1-r} \quad (13.J.8)$$

for real or complex $r \neq 1$.

CHAPTER 14

ESTIMATING THE SPECTRUM

Several alternative methods for constructing reasonable estimators of the spectral density have been proposed and investigated over the years. We will highlight just a few of them that have gained the most acceptance in light of present-day computing power. So-called nonparametric estimation of the spectral density (that is, smoothing of the sample spectral density) assumes very little about the shape of the “true” spectral density. Parametric estimation assumes that an autoregressive model—perhaps of high order—provides an adequate fit to the time series. The estimated spectral density is then based on the theoretical spectral density of the fitted AR model. Some other methods are touched on briefly.

14.1 Smoothing the Spectral Density

The basic idea here is that most spectral densities will change very little over small intervals of frequencies. As such, we should be able to average the values of the sample spectral density over small intervals of frequencies to gain reduced variability. In doing so, we must keep in mind that we may introduce bias into the estimates if, in fact, the theoretical spectral density does change substantially over that interval. *There will always be a trade-off between reducing variability and introducing bias.* We will be required to use judgment to decide how much averaging to perform in a particular case.

Let f be a Fourier frequency. Consider taking a simple average of the neighboring sample spectral density values centered on frequency f and extending m Fourier frequencies on either side of f . We are averaging $2m + 1$ values of the sample spectrum, and the *smoothed sample spectral density* is given by

$$\bar{S}(f) = \frac{1}{2m+1} \sum_{j=-m}^m \hat{S}\left(f + \frac{j}{n}\right) \quad (14.1.1)$$

(When averaging for frequencies near the end points of 0 and $\frac{1}{2}$, we treat the periodogram as symmetric about 0 and $\frac{1}{2}$.)

More generally, we may smooth the sample spectrum with a weight function or *spectral window* $W_m(f)$ with the properties

$$\left. \begin{array}{l} W_m(k) \geq 0 \\ W_m(k) = W_m(-k) \\ \sum_{k=-m}^m W_m(k) = 1 \end{array} \right\} \quad (14.1.2)$$

and obtain a smoothed estimator of the spectral density as

$$\bar{S}(f) = \sum_{k=-m}^m W_m(k) \hat{S}\left(f + \frac{k}{n}\right) \quad (14.1.3)$$

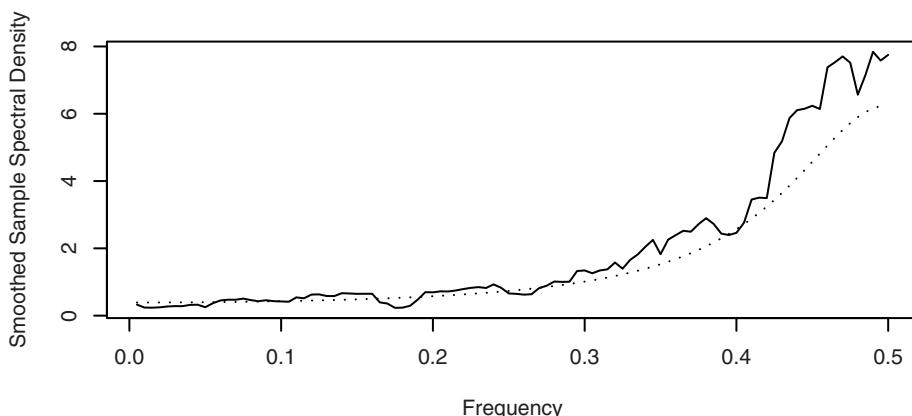
The simple averaging shown in Equation (14.1.1) corresponds to the rectangular spectral window

$$W_m(k) = \frac{1}{2m+1} \text{ for } -m \leq k \leq m \quad (14.1.4)$$

For historical reasons, this spectral window is usually called the *Daniell spectral window* after P. J. Daniell, who first used it in the 1940s.

As an example, consider the simulated AR(1) series whose sample spectral density was shown in Exhibit 13.20 on page 341. Exhibit 14.1 displays the smoothed sample spectrum using the Daniell window with $m = 5$. The true spectrum is again shown as a dotted line. The smoothing did reduce some of the variability that we saw in the sample spectrum.

Exhibit 14.1 Smoothed Spectrum Using the Daniell Window With $m = 5$

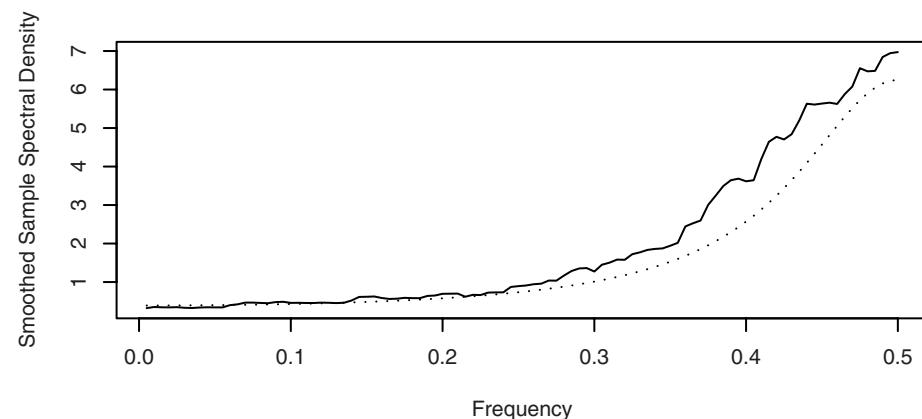


```
> win.graph(width=4.875, height=2.5, pointsize=8)
> set.seed(271435); n=200; phi=-0.6
> y=arima.sim(model=list(ar=phi), n=n)
> k=kernel('daniell', m=5)
```

```
> sp=spec(y,kernel=k,log='no',sub='',xlab='Frequency',
   ylab='Smoothed Sample Spectral Density')
> lines(sp$freq,ARMAspec(model=list(ar=phi),freq=sp$freq,
   plot=F)$spec,lty='dotted')
```

If we make the smoothing window wider (that is, increase m) we will reduce the variability even further. Exhibit 14.2 shows the smoothed spectrum with a choice of $m = 15$. The danger with more and more smoothing is that we may lose important details in the spectrum and introduce bias. The amount of smoothing needed will always be a matter of judgmental trial and error, recognizing the trade-off between reducing variability at the expense of introducing bias.

Exhibit 14.2 Smoothed Spectrum Using the Daniell Window With $m = 15$

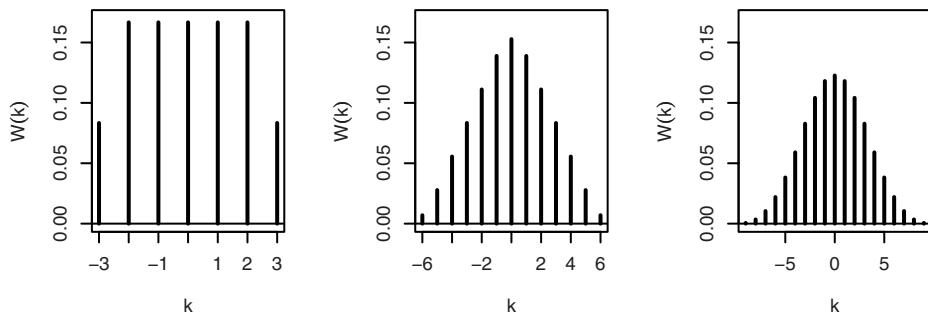


```
> k=kernel('daniell',m=15)
> sp=spec(y,kernel=k,log='no',sub='',xlab='Frequency',
   ylab='Smoothed Sample Spectral Density')
> lines(sp$freq,ARMAspec(model=list(ar=phi),freq=sp$freq,
   plot=F)$spec,lty='dotted')
```

Other Spectral Windows

Many other spectral windows have been suggested over the years. In particular, the abrupt change at the end points of the Daniell window could be softened by making the weights decrease at the extremes. The so-called modified Daniell spectral window simply defines the two extreme weights as half of the other weights still retaining the property that the weights sum to 1. The leftmost graph in Exhibit 14.3 shows the modified Daniell spectral window for $m = 3$.

Exhibit 14.3 The Modified Daniell Spectral Window and Its Convolutions



Another common way to modify spectral windows is to use them to smooth the periodogram more than once. Mathematically, this amounts to using the convolution of the spectral windows. If the modified Daniell spectral window with $m = 3$ is used twice (convolved with itself), we in fact are using the (almost) triangular-shaped window shown in the middle display of Exhibit 14.3. A third smoothing (with $m = 3$) is equivalent to using the spectral window shown in the rightmost panel. This spectral window appears much like a normal curve. We could also use different values of m in the various components of the convolutions.

Most researchers agree that the shape of the spectral window is not nearly as important as the choice of m (or the bandwidth—see below). We will use the modified Daniell spectral window—possibly with one or two convolutions—in our examples.[†]

14.2 Bias and Variance

If the theoretical spectral density does not change much over the range of frequencies that the smoothing window covers, we expect the smoothed estimator to be approximately unbiased. A calculation using this approximation, the spectral window properties in Equations (14.1.2), and a short Taylor expansion produces

$$\begin{aligned} E[\bar{S}(f)] &\approx \sum_{k=-m}^m W_m(k) S\left(f + \frac{k}{n}\right) \\ &\approx \sum_{k=-m}^m W_m(k) \left[S(f) + \frac{k}{n} S'(f) + \frac{1}{2} \left(\frac{k}{n}\right)^2 S''(f) \right] \end{aligned}$$

or

$$E[\bar{S}(f)] \approx S(f) + \frac{1}{n^2} \frac{S''(f)}{2} \sum_{k=-m}^m k^2 W_m(k) \quad (14.2.1)$$

[†]In R, the modified Daniell kernel is the default kernel for smoothing sample spectra, and m may be specified by simply specifying `span = 2m + 1` in the `spec` function where `span` is an abbreviation of the `spans` argument.

So an approximate value for the bias in the smoothed spectral density is given by

$$\text{bias} \approx \frac{1}{n^2} \frac{S''(f)}{2} \sum_{k=-m}^m k^2 W_m(k) \quad (14.2.2)$$

For the Daniell rectangular spectral window, we have

$$\frac{1}{n^2} \sum_{k=-m}^m k^2 W_m(k) = \frac{2}{n^2(2m+1)} \left(\frac{m^3}{3} + \frac{m^2}{2} + \frac{m}{6} \right) \quad (14.2.3)$$

and thus the bias tends to zero as $n \rightarrow \infty$ as long as $m/n \rightarrow 0$.

Using the fact that the sample spectral density values at the Fourier frequencies are approximately uncorrelated and Equation (13.6.5) on page 341, we may also obtain a useful approximation for the variance of the smoothed spectral density as

$$\text{Var}[\bar{S}(f)] \approx \sum_{k=-m}^m W_m^2(k) \text{Var}\left[\hat{S}\left(f + \frac{k}{n}\right)\right] \approx \sum_{k=-m}^m W_m^2(k) S^2(f)$$

so that

$$\text{Var}[\bar{S}(f)] \approx S^2(f) \sum_{k=-m}^m W_m^2(k) \quad (14.2.4)$$

Note that for the Daniell or rectangular spectral window $\sum_{k=-m}^m W_m^2(k) = \frac{1}{2m+1}$, so that as long as $m \rightarrow \infty$ (as $n \rightarrow \infty$) we have consistency.

In general, we require that as $n \rightarrow \infty$ we have $m/n \rightarrow 0$ to reduce bias and $m \rightarrow \infty$ to reduce variance. As a practical matter, the sample size n is usually fixed and we must choose m to balance bias and variance considerations.

Jenkins and Watts (1968) suggest trying three different values of m . A small value will give an idea where the large peaks in $S(f)$ are but may show a large number of peaks, many of which are spurious. A large value of m may produce a curve that is likely to be too smooth. A compromise may then be achieved with the third value of m . Chatfield (2004, p. 135) suggests using $m = \sqrt{n}$. Often trying values for m of $2\sqrt{n}$, \sqrt{n} , and $\frac{1}{2}\sqrt{n}$ will give you some insight into the shape of the true spectrum. Since the width of the window decreases as m decreases, this is sometimes called *window closing*. As Hannan (1973, p. 311) says, “Experience is the real teacher and cannot be got from a book.”

14.3 Bandwidth

In the approximate bias given by Equation (14.2.2), notice that the factor $S''(f)$ depends on the curvature of the true spectral density and will be large in magnitude if there is a sharp peak in $S(f)$ near f but will be small when $S(f)$ is relatively flat near f . This makes intuitive sense, as the motivation for the smoothing of the sample spectral density assumed that the true density changed very little over the range of frequencies used in the spectral window. The square root of the other factor in the approximate bias from

Equation (14.2.2) is sometimes called the *bandwidth*, BW , of the spectral window, namely

$$BW = \frac{1}{n} \sqrt{\sum_{k=-m}^m k^2 W_m(k)} \quad (14.3.1)$$

As we noted in Equation (14.2.3), for the Daniell window this BW will tend to zero as $n \rightarrow \infty$ as long as $m/n \rightarrow 0$. From Equations (14.1.2) on page 352 a spectral window has the mathematical properties of a discrete zero-mean probability density function, so the BW defined here may be viewed as proportional to the standard deviation of the spectral window. As such, it is one way to measure the width of the spectral window. It is interpreted as a measure of width of the band of frequencies used in smoothing the sample spectral density. If the true spectrum contains two peaks that are close relative to the bandwidth of the spectral window, those peaks will be smoothed together when we calculate $\bar{S}(f)$ and they will not be seen as separate peaks. It should be noted that there are many alternative definitions of bandwidth given in the time series literature. Priestley (1981, pp. 513–528) spends considerable time discussing the advantages and disadvantages of the various definitions.

14.4 Confidence Intervals for the Spectrum

The approximate distributional properties of the smoothed spectral density may be easily used to obtain confidence intervals for the spectrum. The smoothed sample spectral density is a linear combination of quantities that have approximate chi-square distributions. A common approximation in such a case is to use some multiple of another chi-square distribution with degrees of freedom obtained by matching means and variances. Assuming $\bar{S}(f)$ to be roughly unbiased with variance given by Equation (14.2.4), matching means and variances leads to approximating the distribution of

$$\frac{\sqrt{v}\bar{S}(f)}{S(f)} \quad (14.4.1)$$

by a chi-square distribution with degrees of freedom given by

$$v = \frac{2}{\sum_{k=-m}^m W_m^2(k)} \quad (14.4.2)$$

Letting $\chi_{v, \alpha/2}^2$ be the $100(\alpha/2)$ th percentile of a chi-square distribution with v degrees of freedom, the inequality

$$\chi_{v, \alpha/2}^2 < \frac{\sqrt{v}\bar{S}(f)}{S(f)} < \chi_{v, 1-\alpha/2}^2$$

can be converted into a $100(1 - \alpha)\%$ confidence statement for $S(f)$ as

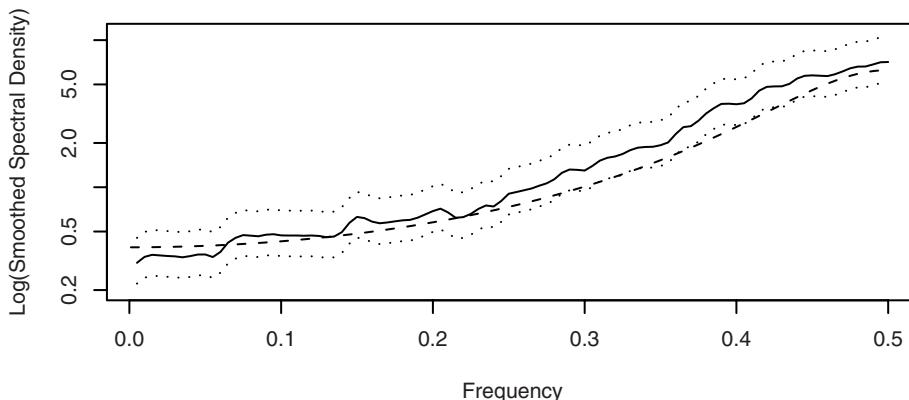
$$\frac{v\bar{S}(f)}{\chi^2_{v, 1-\alpha/2}} < S(f) < \frac{v\bar{S}(f)}{\chi^2_{v, \alpha/2}} \quad (14.4.3)$$

In this formulation, the width of the confidence interval will vary with frequency. A review of Equation (14.2.4) on page 355 shows that the variance of $\bar{S}(f)$ is roughly proportional to the square of its mean. As we saw earlier in Equations (5.4.1) and (5.4.2) on page 98, this suggests that we take the logarithm of the smoothed sample spectral density to stabilize the variance and obtain confidence intervals with width independent of frequency as follows:

$$\log[\bar{S}(f)] + \log\left[\frac{v}{\chi^2_{v, 1-\alpha/2}}\right] \leq \log[S(f)] \leq \log[\bar{S}(f)] + \log\left[\frac{v}{\chi^2_{v, \alpha/2}}\right] \quad (14.4.4)$$

For these reasons it is common practice to plot the logarithms of estimated spectra. If we redo Exhibit 14.2 on page 353 in logarithm terms, we obtain the display shown in Exhibit 14.4, where we have also drawn in the 95% confidence limits (dotted) and the true spectral density (dashed) from the AR(1) model. With a few exceptions, the confidence limits capture the true spectral density.

Exhibit 14.4 Confidence Limits from the Smoothed Spectral Density



```

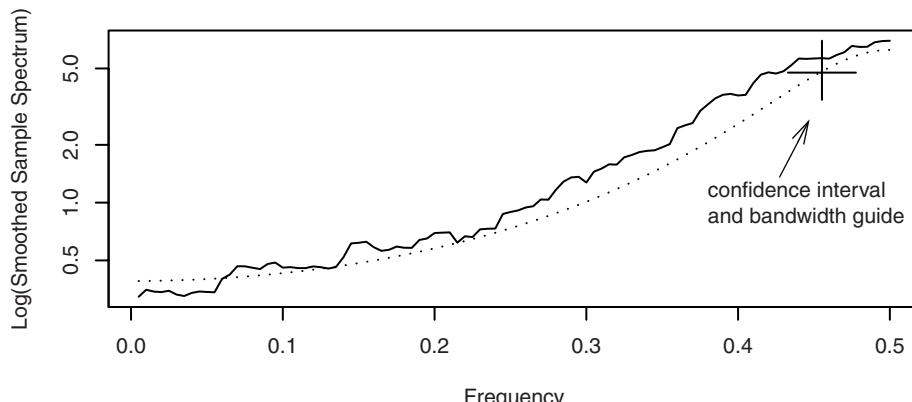
> set.seed(271435); n=200; phi=-0.6
> y=arima.sim(model=list(ar=phi),n=n)
> k=kernel('daniell',m=15)
> sp=spec(y,kernel=k,sub='',xlab='Frequency',
+ ylab='Log (Smoothed Spectral Density)', ci.plot=T,ci.col=NULL)
> lines(sp$freq,ARMAspec(model=list(ar=phi),sp$freq,plot=F)$spec,
+ lty='dashed')

```

Exhibit 14.5 shows a less cluttered display of confidence limits. Here a 95% confidence interval and bandwidth guide is displayed in the upper right-hand corner—the “crosshairs.” The vertical length gives the length (width) of a confidence interval, while

the horizontal line segment indicates the central point[†] of the confidence interval, and its width (length) matches the bandwidth of the spectral window. If you visualize the guide repositioned with the crosshairs centered on the smoothed spectrum above any frequency, you have a visual display of a vertical confidence interval for the “true” spectral density at that frequency and a rough guide of the extent of the smoothing. In this simulated example, we also show the true spectrum as a dotted line.

Exhibit 14.5 Logarithm of Smoothed Spectrum from Exhibit 14.2



```
> sp=spec(y, span=31, sub='', xlab='Frequency',
    ylab='Log (Smoothed Sample Spectrum)')
> lines(sp$freq, ARMAspec(model=list(ar=phi), sp$freq,
    plot=F)$spec, lty='dotted')
```

14.5 Leakage and Tapering

Much of the previous discussion has assumed that the frequencies of interest are the Fourier frequencies. What happens if that is not the case? Exhibit 14.6 displays the periodogram of a series of length $n = 96$ with two pure cosine-sine components at frequencies $f = 0.088$ and $f = 14/96$. The model is simply

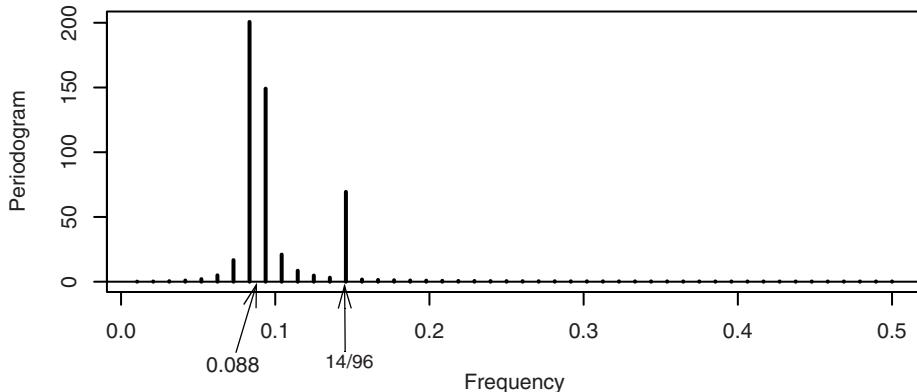
$$Y_t = 3 \cos[2\pi(0.088)t] + \sin\left[2\pi\left(\frac{14}{96}\right)t\right] \quad (14.5.1)$$

Note that with $n = 96$, $f = 0.088$ is *not* a Fourier frequency. The peak with lower power at the Fourier frequency $f = 14/96$ is clearly indicated. However, the peak at $f = 0.088$ is not

[†] The central point is not, in general, halfway between the endpoints, as Equation (14.4.4) determines *asymmetric* confidence intervals. In this example, using the modified Daniell window with $m = 15$, we have $v = 61$ degrees of freedom, so the chi-square distribution used is effectively a normal distribution, and the confidence intervals are nearly symmetric.

there. Rather, the power at this frequency is blurred across several nearby frequencies, giving the appearance of a much wider peak.

Exhibit 14.6 Periodogram of Series with Peaks at $f = 0.088$ and $f = 14/96$



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> t=1:96; f1=0.088; f2=14/96
> y=3*cos(f1*2*pi*t)+sin(f2*2*pi*t)
> periodogram(y); abline(h=0)
```

An algebraic analysis[†] shows that we may view the periodogram as a “smoothed” spectral density formed with the Dirichlet kernel spectral window given by

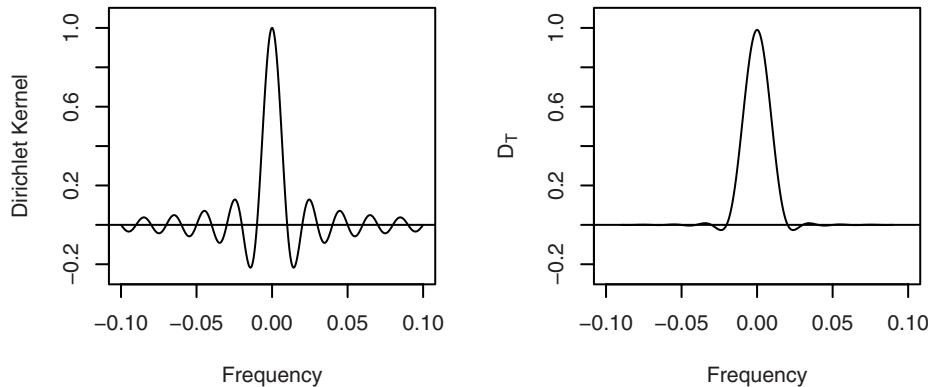
$$D(f) = \frac{1}{n} \frac{\sin(n\pi f)}{\sin(\pi f)} \quad (14.5.2)$$

Note that for all Fourier frequencies $f = j/n$, $D(f) = 0$, so this window has no effect whatsoever at those frequencies. However, the plot of $D(f)$ given on the left-hand side of Exhibit 14.7 shows significant “side lobes” on either side of the main peak. This will cause power at non-Fourier frequencies to leak into the supposed power at the nearby Fourier frequencies, as we see in Exhibit 14.6.

Tapering is one method used to improve the issue with the side lobes. Tapering involves decreasing the data magnitudes at both ends of the series so that the values move gradually toward the data mean of zero. The basic idea is to reduce the end effects of computing a Fourier transform on a series of finite length. If we calculate the periodogram after tapering the series, the effect is to use the modified Dirichlet kernel shown on the right-hand side of Exhibit 14.7 for $n = 100$. Now the side lobes have essentially disappeared.

[†] Appendix K on page 381 gives some of the details.

Exhibit 14.7 Dirichlet Kernel and Dirichlet Kernel after Tapering



The most common form of tapering is based on a cosine bell. We replace the original series Y_t by \tilde{Y}_t , with

$$\tilde{Y}_t = h_t Y_t \quad (14.5.3)$$

where, for example, h_t is the cosine bell given by

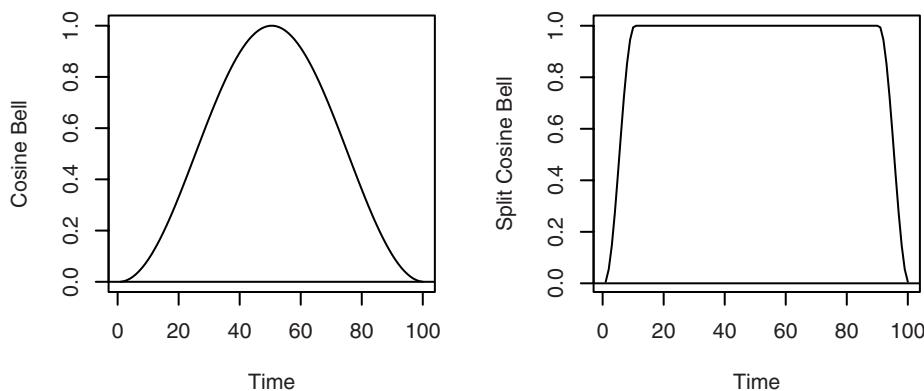
$$h_t = \frac{1}{2} \left\{ 1 - \cos \left[\frac{2\pi(t-0.5)}{n} \right] \right\} \quad (14.5.4)$$

A graph of the cosine bell with $n = 100$ is given on the left-hand side of Exhibit 14.8. A much more common taper is given by a split cosine bell that applies the cosine taper only to the extremes of the time series. The split cosine bell taper is given by

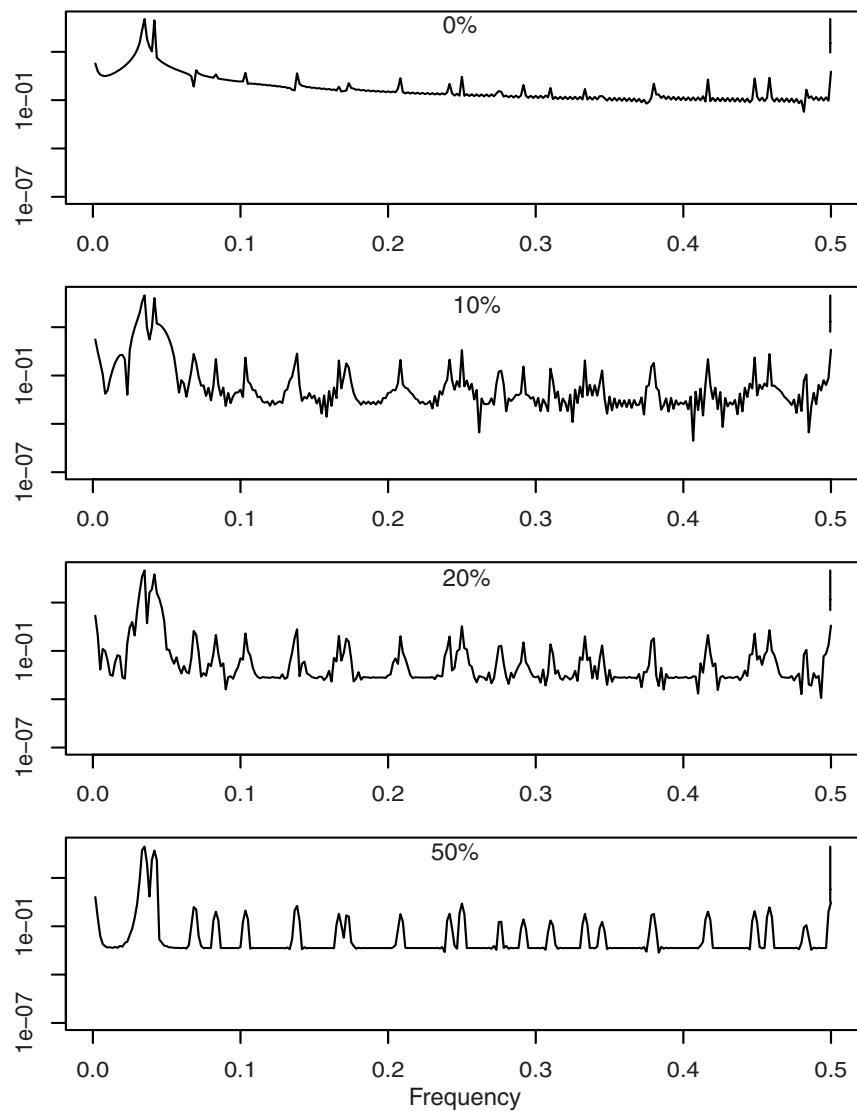
$$h_t = \begin{cases} \frac{1}{2} \left\{ 1 - \cos \left[\frac{\pi(t-1/2)}{m} \right] \right\} & \text{for } 1 \leq t \leq m \\ 1 & \text{for } m+1 \leq t \leq n-m \\ \frac{1}{2} \left\{ 1 - \cos \left[\frac{\pi(n-t+1/2)}{m} \right] \right\} & \text{for } n-m+1 \leq t \leq n \end{cases} \quad (14.5.5)$$

which is called a $100p\%$ cosine bell taper with $p = 2m/n$. A 10% split cosine bell taper is shown on the right-hand side of Exhibit 14.8 again with $n = 100$. Notice that there is a 10% taper on each end, resulting in a total taper of 20%. In practice, split cosine bell tapers of 10% or 20% are in common use.

Exhibit 14.8 Cosine Bell and 10% Taper Split Cosine Bell for $n = 100$



We return to the variable star brightness data first explored on page 325. Exhibit 14.9 displays four periodograms of this series, each with a different amount of tapering. Judging by the length of the 95% confidence intervals displayed in the respective “crosshairs”, we see that the two peaks found earlier in the raw untapered periodogram at frequencies $f_1 = 21/600$ and $f_2 = 25/600$ are clearly real. A more detailed analysis of the minor peaks shown best in the bottom periodogram are all in fact harmonics of the frequencies f_1 and f_2 . There is much more on the topic of leakage reduction and tapering in Bloomfield (2000).

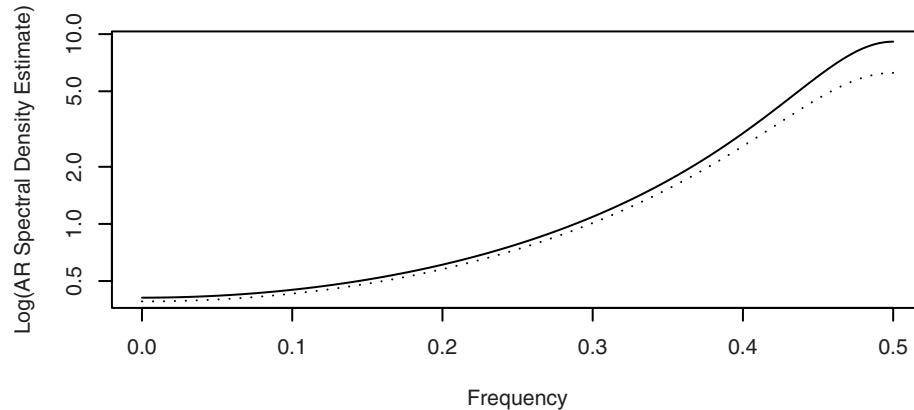
Exhibit 14.9 Variable Star Spectra with Tapers of 0%, 10%, 20%, and 50%

14.6 Autoregressive Spectrum Estimation

In the preceding sections on spectral density estimation, we did not make any assumptions about the parametric form of the true spectral density. However, an alternative method for estimating the spectral density would be to consider fitting an AR, MA, or ARMA model to a time series and then use the spectral density of that model with estimated parameters as our estimated spectral density. (Section 13.5, page 332, discussed the spectral densities of ARMA models.) Often AR models are used with possibly large order chosen to minimize the AIC criterion.

As an example, consider the simulated AR series with $\phi = -0.6$ and $n = 200$ that we used in Exhibits 13.20, 14.1, 14.2, and 14.5. If we fit an AR model, choosing the order to minimize the AIC, and then plot the estimated spectral density for that model, we obtain the results shown in Exhibit 14.10.

Exhibit 14.10 Autoregressive Estimation of the Spectral Density



```
> sp=spec(y,method='ar',sub='',xlab='Frequency',
    ylab='Log(AR Spectral Density Estimate')
> lines(sp$freq,ARMAAspec(model=list(ar=phi),freq=sp$freq,
    plot=F)$spec,lty='dotted')
```

Since these are simulated data, we also show the true spectral density as a dotted line. In this case, the order was chosen as $p = 1$ and the estimated spectral density follows the true density very well. We will show some examples with real time series in Section 14.8.

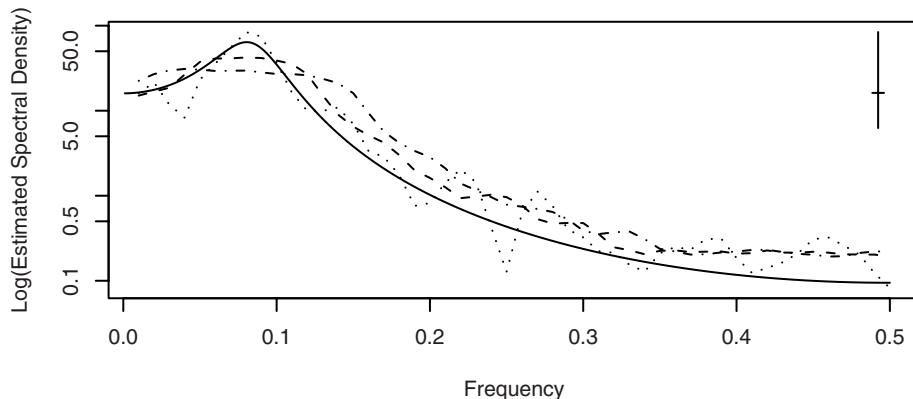
14.7 Examples with Simulated Data

A useful way to get a feel for spectral analysis is with simulated data. Here we know what the answers are and can see what the consequences are when we make choices of spectral window and bandwidth. We begin with an AR(2) model that contains a fairly strong peak in its spectrum.

AR(2) with $\phi_1 = 1.5, \phi_2 = -0.75$: A Peak Spectrum

The spectral density for this model contained a peak at about $f = 0.08$, as displayed in Exhibit 13.14 on page 336. We simulated a time series from this AR(2) model with normal white noise terms with unit variance and sample size $n = 100$. Exhibit 14.11 shows three estimated spectral densities and the true density as a solid line. We used the modified Daniell spectral window with three different values for $span = 2m + 1$ of 3, 9, and 15. A span of 3 gives the least amount of smoothing and is shown as a dotted line. A span of 9 is shown as a dashed line. With $span = 15$, we obtain the most smoothing, and this curve is displayed with a dot-dash pattern. The bandwidths of these three spectral windows are 0.018, 0.052, and 0.087, respectively. The confidence interval and bandwidth guide displayed apply only to the dotted curve estimate. The two others have wider bandwidths and shorter confidence intervals. The estimate based on $span = 9$ is probably the best one, but it does not represent the peak very well.

Exhibit 14.11 Estimated Spectral Densities



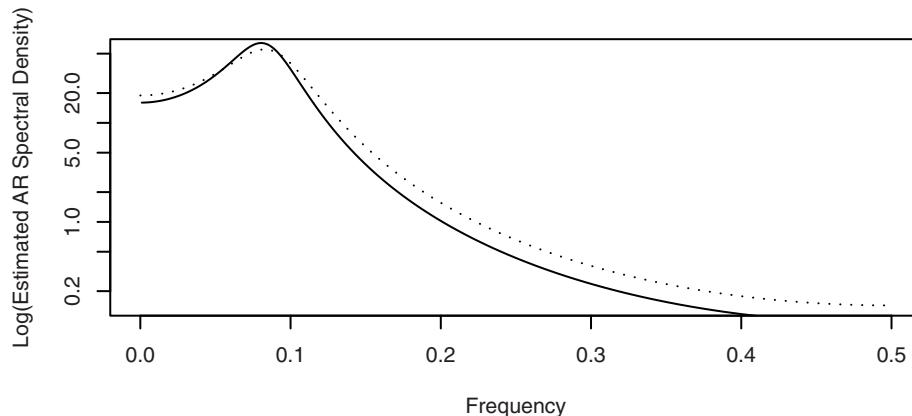
```

> win.graph(width=4.875,height=2.5,pointsize=8)
> set.seed(271435); n=100; phi1=1.5; phi2=-.75
> y=arima.sim(model=list(ar=c(phi1,phi2)),n=n)
> sp1=spec(y,spans=3,sub='',lty='dotted',xlab='Frequency',
+           ylab='Log (Estimated Spectral Density)')
> sp2=spec(y,spans=9,plot=F); sp3=spec(y,spans=15,plot=F)
> lines(sp2$freq,sp2$spec,lty='dashed')
> lines(sp3$freq,sp3$spec,lty='dotdash')
```

```
> f=seq(0.001,.5,by=.001)
> lines(f,ARMAspec(model=list(ar=c(phi1,phi2)),freq=f,
+ plot=F)$spec,lty='solid')
```

We also used the parametric spectral estimation idea and let the software choose the best AR model based on the smallest AIC. The result was an estimated AR(2) model with the spectrum shown in Exhibit 14.12. This is a very good representation of the underlying spectrum, but of course the model was indeed AR(2).

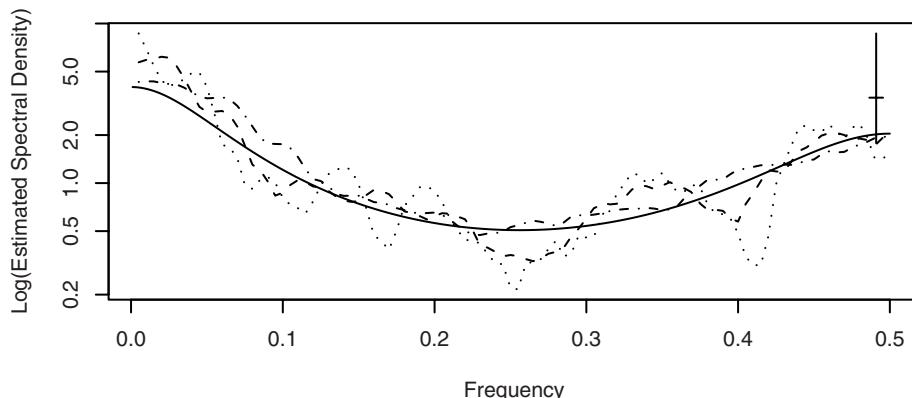
Exhibit 14.12 AR Spectral Estimation: Estimated (dotted), True (solid)



```
> sp4=spec(y,method='ar',lty='dotted',
+ xlab='Frequency',ylab='Log(Estimated AR Spectral Density)')
> f=seq(0.001,0.5, by 0.001)
> lines(f,ARMAspec(model=list(ar=c(phi1,phi2)),freq=f,
+ plot=F)$spec,lty='solid')
> sp4$method # This will tell you order of the AR model selected
```

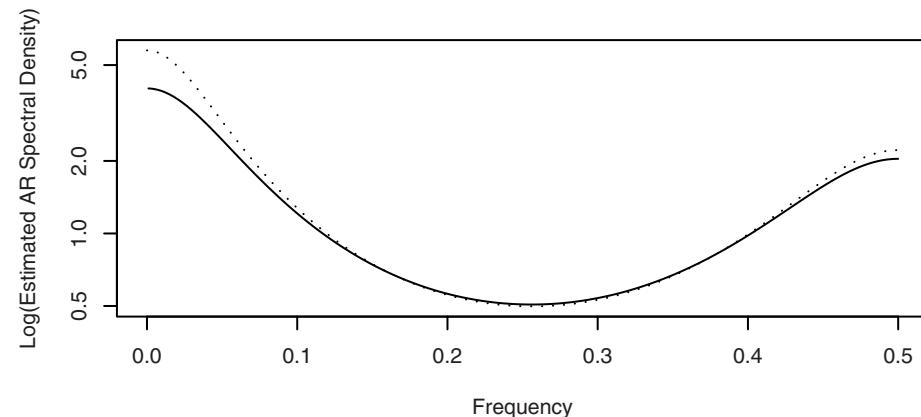
AR(2) with $\phi_1 = 0.1$, $\phi_2 = 0.4$: A Trough Spectrum

Next we look at an AR(2) model with a trough spectrum and a larger sample size. The true spectrum is displayed in Exhibit 13.15 on page 337. We simulated this model with $n = 200$ and unit-variance normal white noise. The three smoothed spectral estimates shown are based on spans of 7, 15, and 31. As before, the confidence limits and bandwidth guide correspond to the smallest span of 7 and hence give the narrowest bandwidth and longest confidence intervals. In our opinion, the middle value of $span = 15$, which is about \sqrt{n} , gives a reasonable estimate of the spectrum.

Exhibit 14.13 Estimated Spectrum for AR(2) Trough Spectrum Model

```
> Use the R code for Exhibit 14.11 with new values for the  
> parameters.
```

Exhibit 14.14 shows the AR spectral density estimate. The minimum AIC was achieved at the true order of the underlying model, AR(2), and the estimated spectral density is quite good.

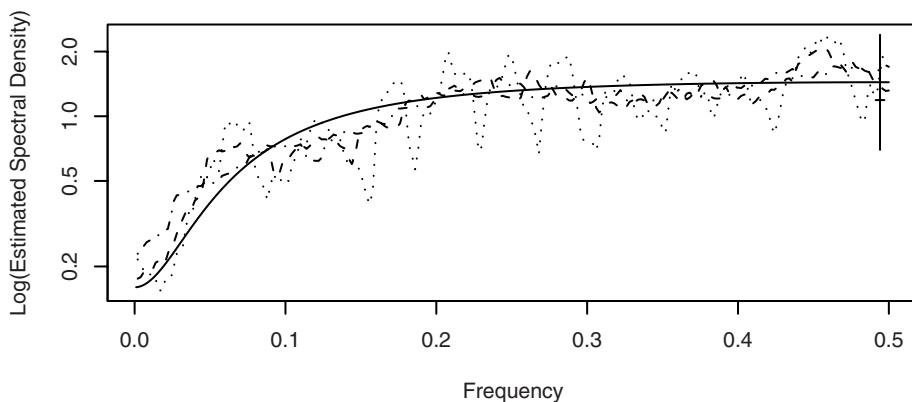
Exhibit 14.14 AR Spectral Estimation: Estimated (dotted), True (solid)

```
> Use the R code for Exhibits 14.11 and 14.12 with new values  
> for the parameters.
```

ARMA(1,1) with $\phi = 0.5$, $\theta = 0.8$

The true spectral density of the mixed model ARMA(1,1) with $\phi = 0.5$ and $\theta = 0.8$ was shown in Exhibit 13.17 on page 338. This model has substantial medium- and high-frequency content but very little power at low frequencies. We simulated this model with a sample size of $n = 500$ and unit-variance normal white noise. Using $\sqrt{n} \approx 22$ as a guide for choosing m , we show three estimates with m of 11, 23, and 45 in Exhibit 14.15. The confidence interval guide indicates that the many peaks produced when $m = 11$ are likely spurious (which, in fact, they are). With such a smooth underlying spectrum, the maximum smoothing shown with $m = 45$ produces a rather good estimate.

Exhibit 14.15 Spectral Estimates for an ARMA(1,1) Process



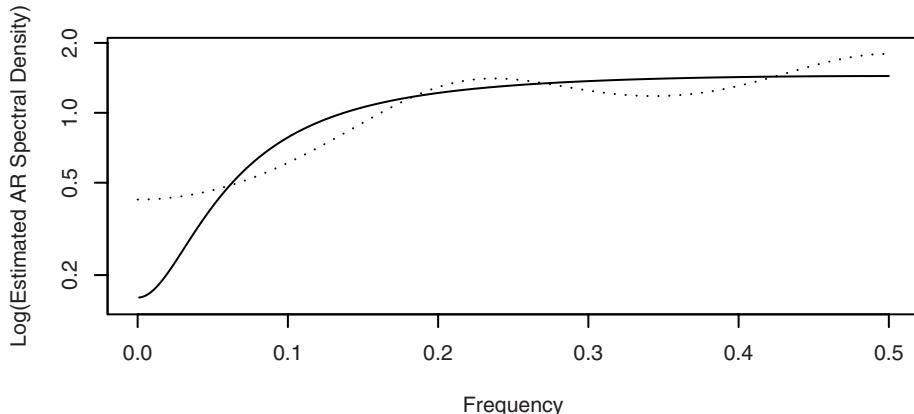
```

> win.graph(width=4.875,height=2.5,pointsize=8)
> set.seed(324135); n=500; phi=.5; theta=.8
> y=arima.sim(model=list(ar=phi,ma=-theta),n=n)
> sp1=spec(y,spans=11,sub='',lty='dotted',
    xlab='Frequency',ylab='Log(Estimated Spectral Density)')
> sp2=spec(y,spans=23,plot=F); sp3=spec(y,spans=45,plot=F)
> lines(sp2$freq,sp2$spec,lty='dashed')
> lines(sp3$freq,sp3$spec,lty='dotdash')
> f=seq(0.001,.5,by=.001)
> lines(f,ARMAspec(model=list(ar=phi,ma=-theta),f,
    plot=F)$spec,lty='solid')

```

In this case, a parametric spectral estimate based on AR models does not work well, as shown in Exhibit 14.16. The software selected an AR(3) model, but the resulting spectral density (dotted) does not reproduce the true density (solid) well at all.

Exhibit 14.16 AR Spectral Estimate for an ARMA(1,1) Process

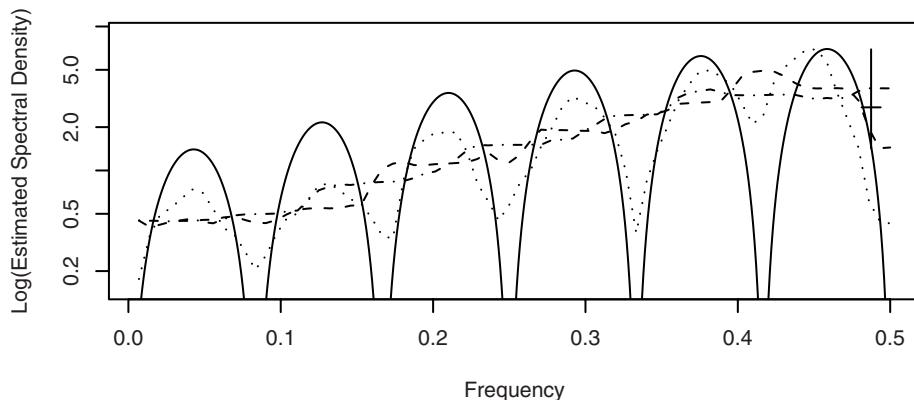


```
> sp4=spec(y,method='ar',lty='dotted',ylim=c(.15,1.9),
  xlab='Frequency',ylab='Log(Estimated AR Spectral Density)')
> f=seq(0.001,.5,by=.001)
> lines(f,ARMAspec(model=list(ar=phi,ma=-theta),f,
  plot=F)$spec,lty='solid')
```

Seasonal MA with $\theta = 0.4$, $\Theta = 0.9$, and $s = 12$

For our final example with simulated data, we choose a seasonal process. The theoretical spectral density is displayed in Exhibit 13.19 on page 340. We simulated $n = 144$ data points with unit-variance normal white noise. We may think of this as 12 years of monthly data. We used modified Daniell spectral windows with $span = 6, 12$, and 24 based on $\sqrt{n} \approx 12$.

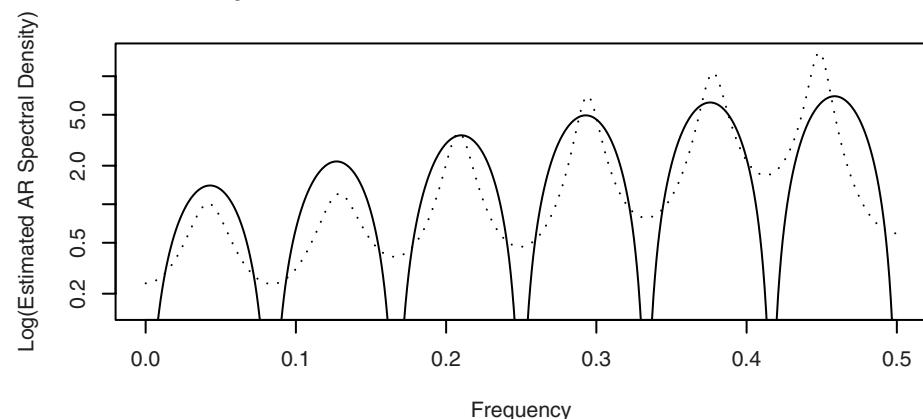
This spectrum contains a lot of detail and is difficult to estimate with only 144 observations. The narrowest spectral window hints at the seasonality, but the two other estimates essentially smooth out the seasonality. The confidence interval widths (corresponding to $m = 6$) do seem to confirm the presence of real seasonal peaks.

Exhibit 14.17 Spectral Estimates for a Seasonal Process

```

> win.graph(width=4.875, height=2.5, pointsize=8)
> set.seed(247135); n=144; theta=.4; THETA=.9
> y=arima.sim(model=list(ma=c(-theta, rep(0, 10), -THETA, theta*THETA
+)), n=n)
> sp1=spec(y, spans=7, sub='', lty='dotted', ylim=c(.15, 9),
+ xlab='Frequency', ylab='Log(Estimated Spectral Density)')
> sp2=spec(y, spans=13, plot=F); sp3=spec(y, spans=25, plot=F)
> lines(sp2$freq, sp2$spec, lty='dashed')
> lines(sp3$freq, sp3$spec, lty='dotdash')
> f=seq(0.001, .5, by=.001)
> lines(f, ARMAspec(model=list(ma=-theta, seasonal=list(sma=-THETA,
+ period=12))), freq=f, plot=F)$spec, lty='solid')

```

Exhibit 14.18 AR Spectral Estimates for a Seasonal Process

```

> sp4=spec(y, method='ar', ylim=c(.15, 15), lty='dotted',
+ xlab='Frequency', ylab='Log(Estimated AR Spectral Density)')

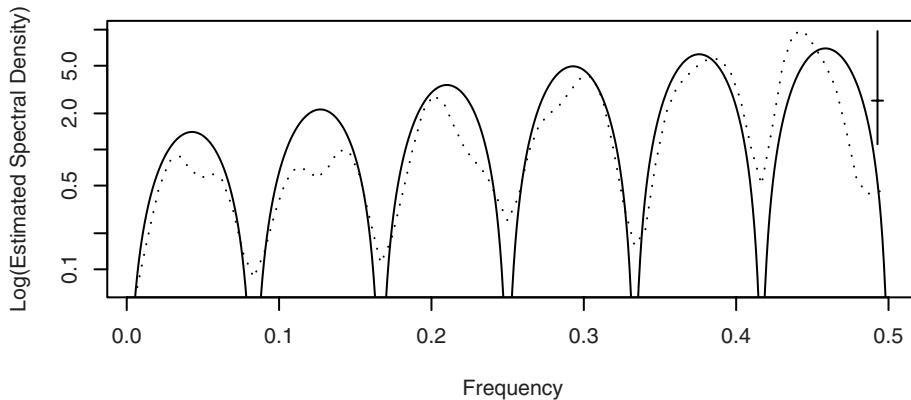
```

```
> f=seq(0.001,.5,by=.001)
> lines(f,ARMAspec(model=list(ma=-theta,seasonal=list(sma=-THETA,
  period=12)),freq=f,plot=F)$spec,lty='solid')
```

Exhibit 14.18 shows the estimated spectrum based on the best AR model. An order of 13 was chosen based on the minimum AIC, and the seasonality does show up quite well. However, the peaks are misplaced at the higher frequencies. Perhaps looking at both Exhibit 14.17 and Exhibit 14.18 we could conclude that the seasonality is real and that a narrow spectral window provides the best estimate of the underlying spectral density given the sample size available.

As a final estimate of the spectrum, we use a convolution of two modified Daniell spectral windows each with *span* = 3, as displayed in the middle of Exhibit 14.3 on page 354. The estimated spectrum is shown in Exhibit 14.19. This is perhaps the best of the estimates that we have shown.

Exhibit 14.19 Estimated Seasonal Spectrum with Convolution Window

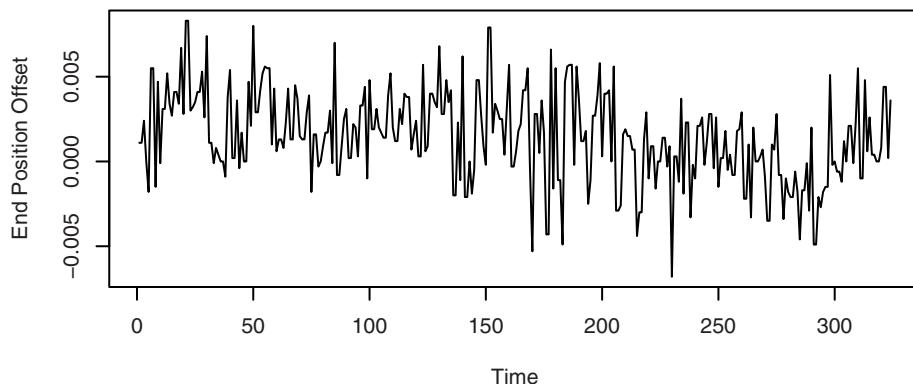


```
> sp5=spec(y,spans=c(3,3),sub='',lty='dotted',
  xlab='Frequency',ylab='Log (Estimated Spectral Density)')
> f=seq(0.001,.5,by=.001)
> lines(f,ARMAspec(model=list(ma=-theta,seasonal=list(sma=-THETA,
  period=12)),freq=f,plot=F)$spec,lty='solid')
```

14.8 Examples with Actual Data

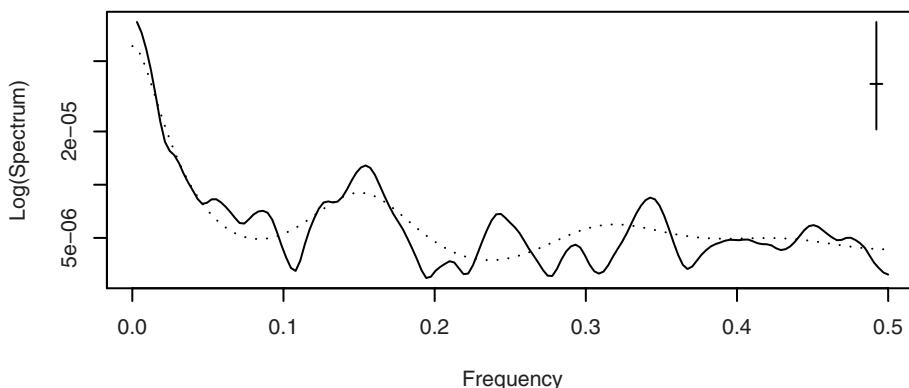
An Industrial Robot

An industrial robot was put through a sequence of maneuvers, and the distance from a desired target end position was recorded in inches. This was repeated 324 times to form the time series shown in Exhibit 14.20.

Exhibit 14.20 Industrial Robot End Position Time Series

```
> data(robot)
> plot(robot, ylab='End Position Offset', xlab='Time')
```

Estimates of the spectrum are displayed in Exhibit 14.21 using the convolution of two modified Daniell spectral windows with $m = 7$ (solid) and with a 10% taper on each end of the series. A plot of this spectral window is shown in the middle of Exhibit 14.3 on page 354. The spectrum was also estimated using a fitted AR(7) model (dotted), the order of which was chosen to minimize the AIC. Given the length of the 95% confidence interval shown, we can conclude that the peak at around a frequency of 0.15 in both estimates is probably real, but those shown at higher frequencies may well be spurious. There is a lot of power shown at very low frequencies, and this agrees with the slowly drifting nature of the series that may be seen in the time series plot in Exhibit 14.20.

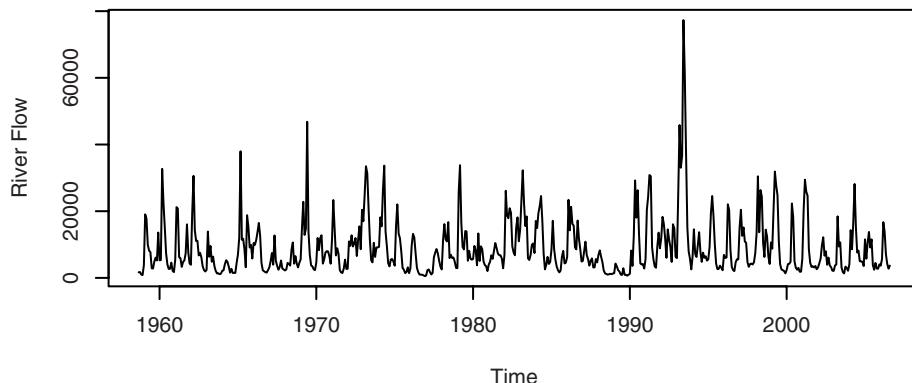
Exhibit 14.21 Estimated Spectrum for the Industrial Robot

```
> spec(robot, spans=c(7,7), taper=.1, sub='', xlab='Frequency',
      ylab='Log (Spectrum)')
> s=spec(robot,method='ar',plot=F)
> lines(s$freq,s$spec,lty='dotted')
```

River Flow

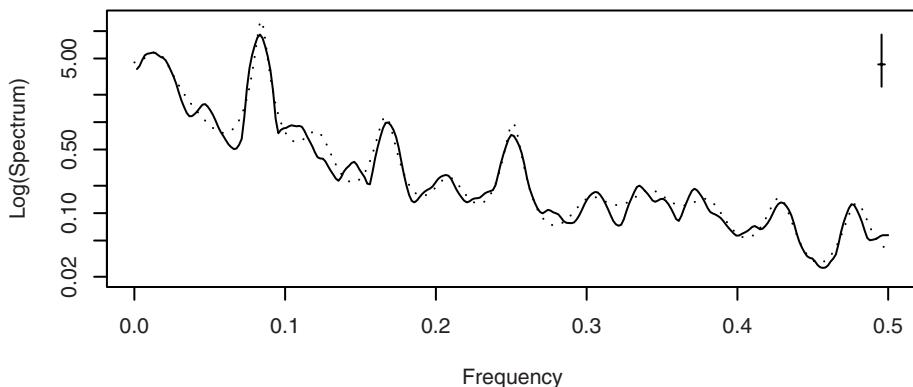
Exhibit 14.22 shows monthly river flow for the Iowa River measured at Wapello, Iowa, for the period September 1958 through August 2006. The data are quite skewed toward the high values, but this was greatly improved by taking logarithms for the analysis.

Exhibit 14.22 River Flow Time Series



```
> data(flow); plot(flow,ylab='River Flow')
```

The sample size for these data is 576 with a square root of 24. The bandwidth of a modified Daniell spectral window is about 0.01. After some experimentation with several spectral window bandwidths, we decided that such a window smoothed too much and we instead used a convolution of two such windows, each with $span = 7$. The bandwidth of this convolved window is about 0.0044. The smoothed spectral density estimate is shown as a solid curve in Exhibit 14.23 together with an estimate based on an AR(7) model (dotted) chosen to minimize the AIC. The prominent peak at frequency $1/12$ represents the strong annual seasonality. There are smaller secondary peaks at about $f \approx 0.17$ and $f \approx 0.25$ that correspond to multiples of the fundamental frequency of $1/12$. They are higher harmonics of the annual frequency.

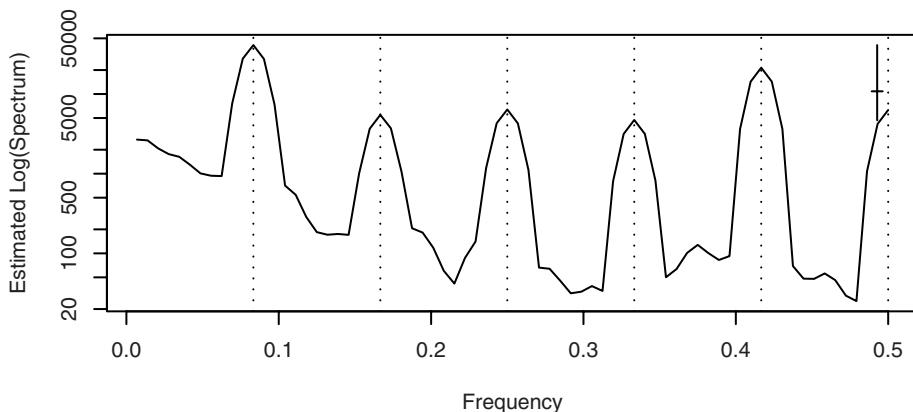
Exhibit 14.23 Log(Spectrum) of Log(Flow)

```
> spec(log(flow), spans=c(7,7), ylim=c(.02,13), sub='',
  ylab='Log (Spectrum)', xlab='Frequency')
> s=spec(log(flow), method='ar', plot=F)
> lines(s$freq,s$spec, lty='dotted')
```

Monthly Milk Production

The top portion of Exhibit 11.14 on page 264, showed U.S. monthly milk production from January 1994 through December of 2005. There is a substantial upward trend together with seasonality. We first remove the upward trend with a simple linear time trend model and consider the residuals from that regression—the seasonals. After trying several spectral bandwidths, we decided to use a convolution of two modified Daniell windows, each with $span = 3$. We believe that otherwise there was too much smoothing. This was confirmed by estimating an AR spectrum that ended up fitting an AR of order 15 with peaks at the same frequencies. Notice that the peaks shown in Exhibit 14.24 are located at frequencies $1/12, 2/12, \dots, 6/12$, with the peak at $1/12$ showing the most power.

Exhibit 14.24 Estimated Spectrum for Milk Production Seasonals



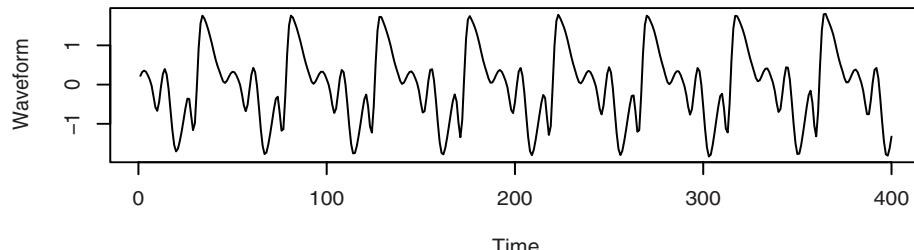
```
> data(milk)
> spec(milk, spans=c(3,3), detrend=T, sub='',
  ylab='Estimated Log(Spectrum)', xlab='Frequency')
> abline(v=seq(1:6)/12, lty='dotted')
```

For a final example in this section, consider the time series shown in Exhibit 14.25. These plots display the first 400 points of two time series of lengths 4423 and 4417, respectively. The complete series were created by recording a trombonist and a euphoniumist each sustaining a B flat (just below middle C) for about 0.4 seconds. The original recording produced data sampled at 44.1 MHz, but this was reduced by subsampling every fourth data point for the analysis shown. Trombones and euphonium are both brass wind instruments that play in the same range, but they have different sized and shaped tubing. The euphonium has larger tubing (a larger bore) that is mostly conical in shape, while the tenor trombone is mostly cylindrical in shape and has a smaller bore. The euphonium sound is considered more mellow than the bright, brassy sound of the trombone. When one listens to these notes being played, they sound rather similar. Our ques-

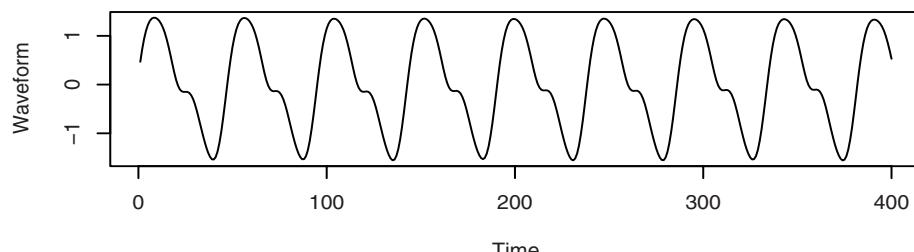
tion is: Does the tubing shape and size affect the harmonics (overtones) enough that the differences may be seen in the spectra of these sounds?

Exhibit 14.25 Trombone and Euphonium Playing Bb

Trombone Bb



Euphonium Bb



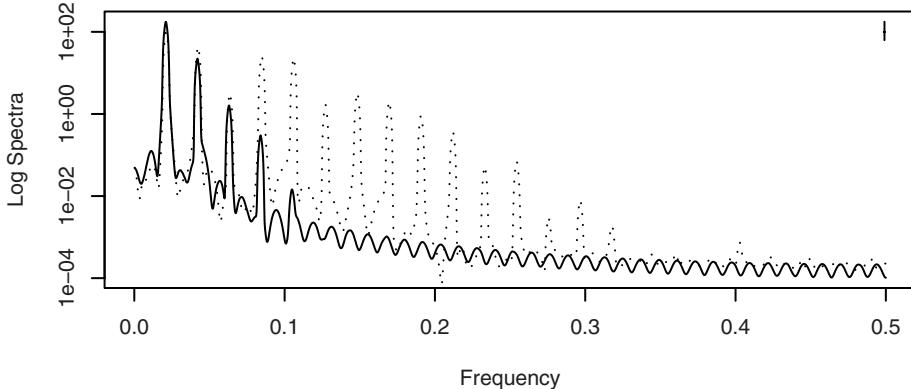
```
> win.graph(width=4.875, height=4, pointsize=8)
> data(tbone); data(euph); oldpar=par; par(mfrow=(c(2,1)))
> trombone=(tbone-mean(tbone))/sd(tbone)
> euphonium=(euph-mean(euph))/sd(euph)
> plot(window(trombone, end=400), main='Trombone Bb',
+      ylab='Waveform', yaxp=c(-1,+1,2))
> plot(window(euphonium, end=400), main='Euphonium Bb',
+      ylab='Waveform', yaxp=c(-1,+1,2)); par=oldpar
```

Exhibit 14.26 displays the estimated spectra for the two waveforms. The solid curve is for the euphonium, and the dotted curve is for the trombone. We used the convolution of two modified Daniell spectral windows, each with $span = 11$, on both series. Since both series are essentially the same length, the bandwidths will both be about 0.0009 and barely perceptible on the bandwidth/confidence interval crosshair shown on the graph.

The first four major peaks occur at the same frequencies, but clearly the trombone has much more spectral power at distinct higher harmonic frequencies. It is suggested

that this may account for the more brassy nature of the trombone sound as opposed to the more mellow sound of the euphonium.

Exhibit 14.26 Spectra for Trombone (dotted) and Euphonium (solid)



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> spec(euph, spans=c(11,11), ylab='Log Spectra',
       xlab='Frequency', sub='')
> s=spec(tbone, spans=c(11,11), plot=F)
> lines(s$freq, s$spec, lty='dotted')
```

14.9 Other Methods of Spectral Estimation

Prior to widespread use of the fast Fourier transform, computing and smoothing the sample spectrum was extremely intensive computationally—especially for long time series. Lag window estimators were used to partially mitigate the computational difficulties.

Lag Window Estimators

Consider the sample spectrum and smoothed sample spectrum. We have

$$\begin{aligned}
 \bar{S}(f) &= \sum_{k=-m}^m W(k) \hat{S}\left(f + \frac{k}{n}\right) \\
 &= \sum_{k=-m}^m W(k) \left[\sum_{j=-n+1}^{n-1} \hat{\gamma}_j e^{-2\pi i \left(f + \frac{k}{n}\right) j} \right] \\
 &= \sum_{j=-n+1}^{n-1} \hat{\gamma}_j \left[\sum_{k=-m}^m W(k) e^{-2\pi i \frac{k}{n} j} \right] e^{-2\pi i f j}
 \end{aligned} \tag{14.9.1}$$

or

$$\bar{S}(f) = \sum_{j=-n+1}^{n-1} \hat{\gamma}_j w\left(\frac{j}{n}\right) e^{-2\pi ifj} \quad (14.9.2)$$

where

$$w\left(\frac{j}{n}\right) = \sum_{k=-m}^m W(k) e^{-2\pi ik\left(\frac{j}{n}\right)} \quad (14.9.3)$$

Equation (14.9.2) suggests defining and investigating a class of spectral estimators defined as

$$\tilde{S}(f) = \sum_{j=-n+1}^{n-1} w\left(\frac{j}{n}\right) \hat{\gamma}_j \cos(2\pi f j) \quad (14.9.4)$$

where the function $w(x)$ has the properties

$$\begin{aligned} w(x) &= w(-x) \\ w(0) &= 1 \\ w(x) &\leq 1 \quad \text{for } |x| \leq 1 \end{aligned} \quad (14.9.5)$$

The function $w(x)$ is called a *lag window* and determines how much weight is given to the sample autocovariance at each lag.

The rectangular lag window is defined by

$$w(x) = 1 \quad \text{for } |x| \leq 1 \quad (14.9.6)$$

and the corresponding lag window spectral estimator is simply the sample spectrum. This estimator clearly gives too much weight to large lags where the sample autocovariances are based on too few data points and are unreliable.

The next simplest lag window is the truncated rectangular lag window, which simply omits large lags from the computation. It is defined as

$$w\left(\frac{j}{n}\right) = 1 \quad \text{for } |j| \leq m \quad (14.9.7)$$

where the computational advantage is achieved by choosing m much smaller than n .

The triangular, or Bartlett, lag window downweights higher lags linearly and is defined as

$$w\left(\frac{j}{n}\right) = 1 - \left| \frac{j}{m} \right| \quad \text{for } |j| \leq m \quad (14.9.8)$$

Other common lag windows are associated with the names of Parzen, Tukey-Hamming, and Tukey-Hanning. We will not pursue these further here, but much more information on the lag window approach to spectral estimation may be found in the books of Bloomfield (2000), Brillinger (2001), Brockwell and Davis (1991), and Priestley (1981).

Other Smoothing Methods

Other methods for smoothing the sample spectrum have been proposed. Kooperberg et al. (1995) proposed using splines to estimate the spectral distribution. Fan and Kreutzberger (1998) investigated local smoothing polynomials and Whittle's likelihood for spectral estimation. This approach uses automatic bandwidth selection to smooth the sample spectrum. See also Yoshihide (2006), Jiang and Hui (2004), and Fay et al. (2002).

14.10 Summary

Given the undesirable characteristics of the sample spectral density, we introduced the smoothed sample spectral density and showed that it could be constructed to improve the properties. The important topics of bias, variance, leakage, bandwidth, and tapering were investigated. A procedure for forming confidence intervals was discussed, and all of the ideas were illustrated with both real and simulated time series data.

EXERCISES

- 14.1 Consider the variance of $\bar{S}(f)$ with the Daniell spectral window. Instead of using Equation (14.2.4) on page 355, use the fact that $2\hat{S}(f)/S(f)$ has approximately a chi-square distribution with two degrees of freedom to show that the smoothed sample spectral density has an approximate variance of $S^2(f)/(2m + 1)$.
- 14.2 Consider various convolutions of the simple Daniell rectangular spectral window.
 - (a) Construct a panel of three plots similar to those shown in Exhibit 14.3 on page 354 but with the Daniell spectral window and with $m = 5$. The middle graph should be the convolution of two Daniell windows and the leftmost graph the convolution of three Daniell windows.
 - (b) Evaluate the bandwidths and degrees of freedom for each of the spectral windows constructed in part (a). Use $n = 100$.
 - (c) Construct another panel of three plots similar to those shown in Exhibit 14.3 but with the modified Daniell spectral window. This time use $m = 5$ for the first graph and convolve two with $m = 5$ and $m = 7$ for the second. Convolve three windows with m 's of 5, 7, and 11 for the third graph.
 - (d) Evaluate the bandwidths and degrees of freedom for each of the spectral windows constructed in part (c). Use $n = 100$.

- 14.3** For the Daniell rectangular spectral window show that

$$(a) \frac{1}{n^2} \sum_{k=-m}^m k^2 W_m(k) = \frac{2}{n^2(2m+1)} \left(\frac{m^3}{3} + \frac{m^2}{2} + \frac{m}{6} \right)$$

- (b) Show that if m is chosen as $m = c\sqrt{n}$ for any constant c , then the right-hand side of the expression in part (a) tends to zero as n goes to infinity.
- (c) Show that if $m = c\sqrt{n}$ for any constant c , then the approximate variance of the smoothed spectral density given by the right-hand side of Equation (14.2.4) on page 355 tends to zero as n tends to infinity.
- 14.4** Suppose that the distribution of $\bar{S}(f)$ is to be approximated by a multiple of a chi-square variable with degrees of freedom v , so that $\bar{S}(f) \approx c\chi_v^2$. Using the approximate variance of $\bar{S}(f)$ given in Equation (14.2.4) on page 355 and the fact that $\bar{S}(f)$ is approximately unbiased, equate means and variances and find the values for c and v (thus establishing Equation (14.4.2) on page 356).
- 14.5** Construct a time series of length $n = 48$ according to the expression

$$Y_t = \sin[2\pi(0.28)t]$$

Display the periodogram of the series and explain its appearance.

- 14.6** Estimate the spectrum of the Los Angeles annual rainfall time series. The data are in the file named larain. Because of the skewness in the series, use the logarithms of the raw rainfall values. The square root of the series length suggests a value for the *span* of about 11. Use the modified Daniell spectral window, and be sure to set the vertical limits of the plot so that you can see the whole confidence interval guide. Comment on the estimated spectrum.
- 14.7** The file named spots1 contains annual sunspot numbers for 306 years from 1700 through 2005.
- (a) Display the time series plot of these data. Does stationarity seem reasonable for this series?
- (b) Estimate the spectrum using a modified Daniell spectral window convoluted with itself and a *span* of 3 for both. Interpret the plot.
- (c) Estimate the spectrum using an AR model with the order chosen to minimize the AIC. Interpret the plot. What order was selected?
- (d) Overlay the estimates obtained in parts (b) and (c) above onto one plot. Do they agree to a reasonable degree?
- 14.8** Consider the time series of average monthly temperatures in Dubuque, Iowa. The data are in the file named tempdub and cover from January 1964 to December 1975 for an n of 144.
- (a) Estimate the spectrum using a variety of *span* values for the modified Daniell spectral window.
- (b) In your opinion, which of the estimates in part (a) best represents the spectrum of the process? Be sure to use bandwidth considerations and confidence limits to back up your argument.

- 14.9** An EEG (electroencephalogram) time series is given in the data file named `eeg`. An electroencephalogram is a noninvasive test used to detect and record the electrical activity generated in the brain. These data were measured at a sampling rate of 256 per second and came from a patient suffering a seizure. The total record length is $n = 13,000$ —or slightly less than one minute.
- (a) Display the time series plot and decide if stationarity seems reasonable.
 - (b) Estimate the spectrum using a modified Daniell spectral window convolved with itself and a *span* of 51 for both components of the convolution. Interpret the plot.
 - (c) Estimate the spectrum using an AR model with the order chosen to minimize the AIC. Interpret the plot. What order was selected?
 - (d) Overlay the estimates obtained in parts (b) and (c) above onto one plot. Do they agree to a reasonable degree?
- 14.10** The file named `electricity` contains monthly U. S. electricity production values from January 1994 to December 2005. A time series plot of the logarithms of these values is shown in Exhibit 11.14 on page 264. Since there is an upward trend and increasing variability at higher levels in these data, use the first difference of the logarithms for the remaining analysis.
- (a) Construct a time series plot of the first difference of the logarithms of the electricity values. Does a stationary model seem warranted at this point?
 - (b) Display the smoothed spectrum of the first difference of the logarithms using a modified Daniell spectral window and *span* values of 25, 13, and 7. Interpret the results.
 - (c) Now use a spectral window that is a convolution of two modified Daniell windows each with *span* = 3. Also use a 10% taper. Interpret the results.
 - (d) Estimate the spectrum using an AR model with the order chosen to minimize the AIC. Interpret the plot. What order was selected?
 - (e) Overlay the estimates obtained in parts (c) and (d) above onto one plot. Do they agree to a reasonable degree?
- 14.11** Consider the monthly milk production time series used in Exhibit 14.24 on page 374. The data are in the file named `milk`.
- (a) Estimate the spectrum using a spectral window that is a convolution of two modified Daniell windows each with *span* = 7. Compare these results with those shown in Exhibit 14.24.
 - (b) Estimate the spectrum using a single modified Daniell spectral window with *span* = 7. Compare these results with those shown in Exhibit 14.24 and those in part (a).
 - (c) Finally, estimate the spectrum using a single modified Daniell spectral window with *span* = 11. Compare these results with those shown in Exhibit 14.24 and those in parts (a) and (b).
 - (d) Among the four different estimates considered here, which do you prefer and why?

14.12 Consider the river flow series displayed in Exhibit 14.22 on page 372. An estimate of the spectrum is shown in Exhibit 14.23 on page 373. The data are in the file named *flow*.

- (a) Here $n = 576$ and $\sqrt{n} = 24$. Estimate the spectrum using $span = 25$ with the modified Daniell spectral window. Compare your results with those shown in Exhibit 14.23.
- (b) Estimate the spectrum using $span = 13$ with the modified Daniell spectral window and compare your results to those obtained in part (a) and in Exhibit 14.23.

14.13 The time series in the file named *tuba* contains about 0.4 seconds of digitized sound from a tuba playing a B flat one octave and one note below middle C.

- (a) Display a time series plot of the first 400 of these data and compare your results with those shown in Exhibit 14.25 on page 375, for the trombone and euphonium.
- (b) Estimate the spectrum of the tuba time series using a convolution of two modified Daniell spectral windows, each with $span = 11$.
- (c) Compare the estimated spectrum obtained in part (b) with those of the trombone and euphonium shown in Exhibit 14.26 on page 376. (You may want to overlay several of these spectra.) Remember that the tuba is playing one octave lower than the two other instruments.
- (d) Do the higher-frequency components of the spectrum for the tuba look more like those of the trombone or those of the euphonium? (Hint: The euphonium is sometimes called a tenor tuba!)

Appendix K: Tapering and the Dirichlet Kernel

Suppose $Y_t = \cos(2\pi f_0 t + \Phi)$ for $t = 1, 2, \dots, n$, where f_0 is not necessarily a Fourier frequency. Since it will not affect the periodogram, we will actually suppose that

$$Y_t = e^{2\pi i f_0 t} \quad (14.K.1)$$

in order to simplify the mathematics. Then the discrete-time Fourier transform of this sequence is given by

$$\frac{1}{n} \sum_{t=1}^n Y_t e^{-2\pi i f t} = \frac{1}{n} \sum_{t=1}^n e^{2\pi i (f_0 - f)t} \quad (14.K.2)$$

By Equations (13.J.7) and (13.J.8) on page 350, for any z ,

$$\begin{aligned} \frac{1}{n} \sum_{t=1}^n e^{2\pi i z t} &= \frac{1}{n} e^{2\pi i z} \frac{(e^{2\pi i n z} - 1)}{(e^{2\pi i z} - 1)} \\ &= \frac{1}{n} e^{\pi i(n+1)z} \frac{(e^{\pi i n z} - e^{-\pi i n z})}{(e^{\pi i z} - e^{-\pi i z})} \end{aligned}$$

so that

$$\frac{1}{n} \sum_{t=1}^n e^{2\pi i z t} = e^{\pi i(n+1)z} \left[\frac{1}{n} \frac{\sin(\pi n z)}{\sin(\pi z)} \right] \quad (14.K.3)$$

The function

$$D(z) = \frac{1}{n} \frac{\sin(\pi n z)}{\sin(\pi z)} \quad (14.K.4)$$

is the Dirichlet kernel shown on the left-hand side of Exhibit 14.7 on page 360 for $n = 100$. These results lead to the following relationship for the periodogram of Y_t :

$$I(f) \propto |D(f - f_0)|^2 \quad (14.K.5)$$

Remember that for all Fourier frequencies $D(f) = 0$, so that this window has no effect at those frequencies. Leakage occurs when there is substantial power at non-Fourier frequencies. Now consider tapering Y_t with a cosine bell. We have

$$\begin{aligned} \tilde{Y}_t &= \frac{1}{2} \left\{ 1 - \cos \left[\frac{2\pi(t-0.5)}{n} \right] \right\} Y_t \\ &= \frac{1}{2} e^{2\pi i f_0 t} - \frac{1}{4} e^{2\pi i f_0 t + 2\pi i(t-1/2)/n} - \frac{1}{4} e^{2\pi i f_0 t - 2\pi i(t-1/2)/n} \end{aligned} \quad (14.K.6)$$

and after some more algebra we obtain

$$\begin{aligned} \frac{1}{n} \sum_{t=1}^n \tilde{Y}_t e^{-2\pi i f t} \\ &= e^{\pi i(n+1)(f_0 - f)t} \left[\frac{1}{4} D\left(f - f_0 - \frac{1}{n}\right) + \frac{1}{2} D(f - f_0) + \frac{1}{4} D\left(f - f_0 + \frac{1}{n}\right) \right] \end{aligned} \quad (14.K.7)$$

The function

$$\tilde{D}(f) = \frac{1}{4} D\left(f - f_0 - \frac{1}{n}\right) + \frac{1}{2} D(f - f_0) + \frac{1}{4} D\left(f - f_0 + \frac{1}{n}\right) \quad (14.K.8)$$

is the tapered or modified Dirichlet kernel that is plotted on the right-hand side of Exhibit 14.7 on page 360 for $n = 100$. The periodogram of the tapered series is proportional to $|\tilde{D}(f)|^2$, and the side lobe problem is substantially mitigated.

CHAPTER 15

THRESHOLD MODELS

It can be shown (Wold, 1948) that any weakly stationary process $\{Y_t\}$ admits the Wold decomposition

$$Y_t = U_t + e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots$$

where e_t equals the deviation of Y_t from the best linear predictor based on all past Y values, and $\{U_t\}$ is a purely deterministic stationary process, with e_t being uncorrelated with U_s , for any t and s . A purely deterministic process is a process that can be predicted to arbitrary accuracy; (that is, with arbitrarily small mean squared error) by some linear predictors of finitely many past lags of the process. A simple example of a purely deterministic process is $U_t \equiv \mu$, a constant. A more subtle example is the random cosine wave model introduced on page 18. In essence, $\{U_t\}$ represents the stochastic, stationary “trend” in the data. The prediction errors $\{e_t\}$ are a white noise sequence, and e_t represents the “new” component making up Y_t and hence is often called the innovation of the process. The Wold decomposition then states that any weakly stationary process is the sum of a (possibly infinite-order) MA process and a deterministic trend. Thus, we can compute the best linear predictor within the framework of $MA(\infty)$ processes that can further be approximated by finite-order ARMA processes. The Wold decomposition thus guarantees the versatility of the ARMA models in prediction with stationary processes.

However, except for convenience, there is no reason for restricting to linear predictors. If we allow nonlinear predictors and seek the best predictor of Y_t based on past values of Y that minimizes the mean squared prediction error, then the best predictor need no longer be the best linear predictor. The solution is simply the conditional mean of Y_t given all past Y values. The Wold decomposition makes it clear that the best one-step-ahead linear predictor is the best one-step-ahead predictor if and only if $\{e_t\}$ in the Wold decomposition satisfies the condition that the conditional mean of e_t given past e 's is identically equal to 0. The $\{e_t\}$ satisfying the latter condition is called a sequence of martingale differences, so the condition will be referred to as the martingale difference condition. The martingale difference condition holds if, for example, $\{e_t\}$ is a sequence of independent, identically distributed random variables with zero mean. But it also holds if $\{e_t\}$ is some GARCH process. Nonetheless, when the martingale difference condition fails, nonlinear prediction will lead to a more accurate prediction. Hannan (1973) defines a linear process to be one where the best one-step-ahead linear predictor is the best one-step-ahead predictor.

The time series models discussed so far are essentially linear models in the sense that, after suitable instantaneous transformation, the one-step-ahead conditional mean is a linear function of the current and past values of the time series variable. If the errors are normally distributed, as is commonly assumed, a linear ARIMA model results in a normally distributed process. Linear time series methods have proved to be very useful in practice. However, linear, normal processes do suffer from some limitations. For example, a stationary normal process is completely characterized by its mean and autocovariance function; hence the process reversed in time has the same distribution as the original process. The latter property is known as time reversibility. Yet, many real processes appear to be time-irreversible. For example, the historical daily closing price of a stock generally rose gradually but, if it crashed, it did so precipitously, signifying a time-irreversible data mechanism. Moreover, the one-step-ahead conditional mean may be nonlinear rather than linear in the current and past values. For example, animal abundance processes may be nonlinear due to finite-resource constraints. Specifically, while moderately high abundance in one period is likely to be followed by higher abundance in the next period, extremely high abundance may lead to a population crash in the ensuing periods. Nonlinear time series models generally display rich dynamical structure. Indeed, May (1976) showed that a very simple nonlinear deterministic difference equation may admit chaotic solutions in the sense that its time series solutions are sensitive to the initial values, which may appear to be indistinguishable from a white noise sequence based on correlation analysis. Nonlinear time series analysis thus may provide more accurate predictions, which can be very substantial in certain parts of the state space, and shed novel insights on the underlying dynamics of the data. Nonlinear time series analysis was earnestly initiated around the late 1970s, prompted by the need for modeling the nonlinear dynamics shown by real data; see Tong (2007). Except for cases with well-developed theory accounting for the underlying mechanism of an observed time series, the nonlinear data mechanism is generally unknown. Thus, a fundamental problem of empirical nonlinear time series analysis concerns the choice of a general nonlinear class of models. Here, our goal is rather modest in that we introduce the threshold model, which is one of the most important classes of nonlinear time series models. For a systematic account of nonlinear time series analysis and chaos, see Tong (1990) and Chan and Tong (2001).

15.1 Graphically Exploring Nonlinearity

In ARIMA modeling, the innovation (error) process is often specified as independent and identically normally distributed. The normal error assumption implies that the stationary time series is also a normal process; that is, any finite set of time series observations are jointly normal. For example, the pair (Y_1, Y_2) has a bivariate normal distribution and so does any pair of Y 's; the triple (Y_1, Y_2, Y_3) has a trivariate normal distribution and so does any triple of Y 's, and so forth. When data are nonnormal, instantaneous transformation of the form $h(Y_t)$, for example, $h(Y_t) = \sqrt{Y_t}$, may be applied to the data in the hope that a normal ARIMA model can serve as a good approximation to the underlying data-generating mechanism. The normality assumption is mainly

adopted for convenience in statistical inference. In practice, an ARIMA model with nonnormal innovations may be entertained. Indeed, such processes have very rich and sometimes exotic dynamics; see Tong (1990). If the normal error assumption is maintained, then a nonlinear time series is generally not normally distributed. Nonlinearity may then be explored by checking whether or not a finite set of time series observations are jointly normal; for example, whether or not the two-dimensional distribution of pairs of Y 's is normal. This can be checked by plotting the scatter diagram of Y_t against Y_{t-1} or Y_{t-2} , and so forth. For a bivariate normal distribution, the scatter diagram should resemble an elliptical data cloud with decreasing density from its center. Departure from such a pattern (for example, existence of a large hole in the data cloud) may signify that the data are nonnormal and the underlying process may be nonlinear.

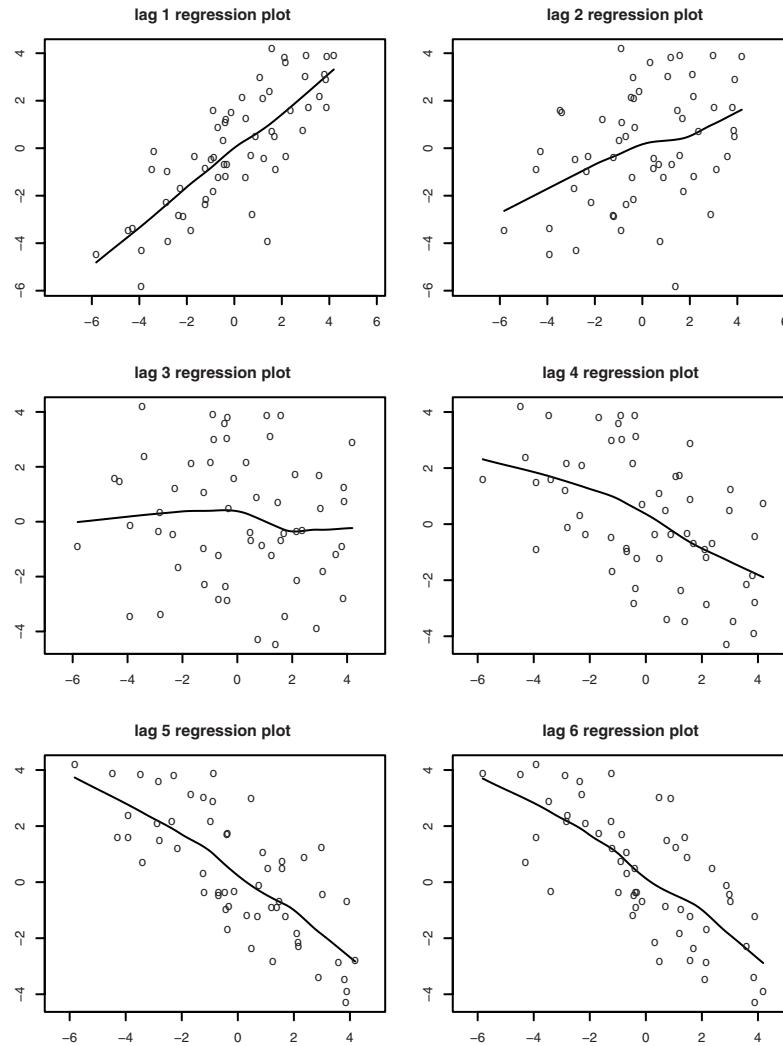
Exhibit 15.1 shows the scatter diagrams of Y_t versus its lag 1 to lag 6, where we simulated data from the ARIMA(2,1) model

$$Y_t = 1.6Y_{t-1} - 0.94Y_{t-2} + e_t - 0.64e_{t-1} \quad (15.1.1)$$

with the innovations being standard normal. Note that the data clouds in the scatter diagrams are roughly elliptically shaped.

To help us visualize the relationship between the response and its lags, we draw fitted nonparametric regression lines on each scatter diagram. For example, on the scatter diagram of Y_t against Y_{t-1} , a nonparametric estimate of the conditional mean function of Y_t given Y_{t-1} , also referred to as the lag 1 regression function, is superimposed. (Specifically, the lag 1 regression function equals $m_1(y) = E(Y_t|Y_{t-1}=y)$ as a function of y .) If the underlying process is linear and normal, the true lag 1 regression function must be linear and so we expect the nonparametric estimate of it to be close to a straight line. On the other hand, a curved lag 1 regression estimate may suggest that the underlying process is nonlinear. Similarly, one can explore the lag 2 regression function (that is, the conditional mean of Y_t given $Y_{t-2} = y$) as a function of y and higher-lag analogues. In the case of strong departure from linearity, the shape of these regression functions may provide some clue as to what nonlinear model may be appropriate for the data. Note that all lagged regression curves in Exhibit 15.1 are fairly straight, suggesting that the underlying process is linear, which indeed we know is the case.

Exhibit 15.1 Lagged Regression Plots for a Simulated ARMA(2,1) Process. Solid lines are fitted regression curves.



```

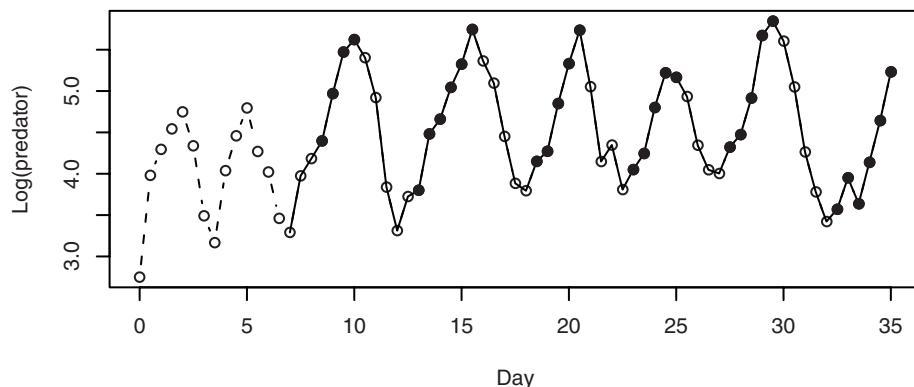
> win.graph(width=4.875, height=6.5, pointsize=8)
> set.seed(2534567); par(mfrow=c(3,2))
> y=arima.sim(n=61,model=list(ar=c(1.6, -0.94), ma=-0.64))
> lagplot(y)

```

We now illustrate the technique of a lagged regression plot with a real example. Exhibit 15.2 plots an experimental time series response as the number of individuals (*Didinium natsutum*, a protozoan) per ml measured every twelve hours over a period of 35 days; see Veilleux (1976) and Jost and Ellner (2000). The experiment studied the

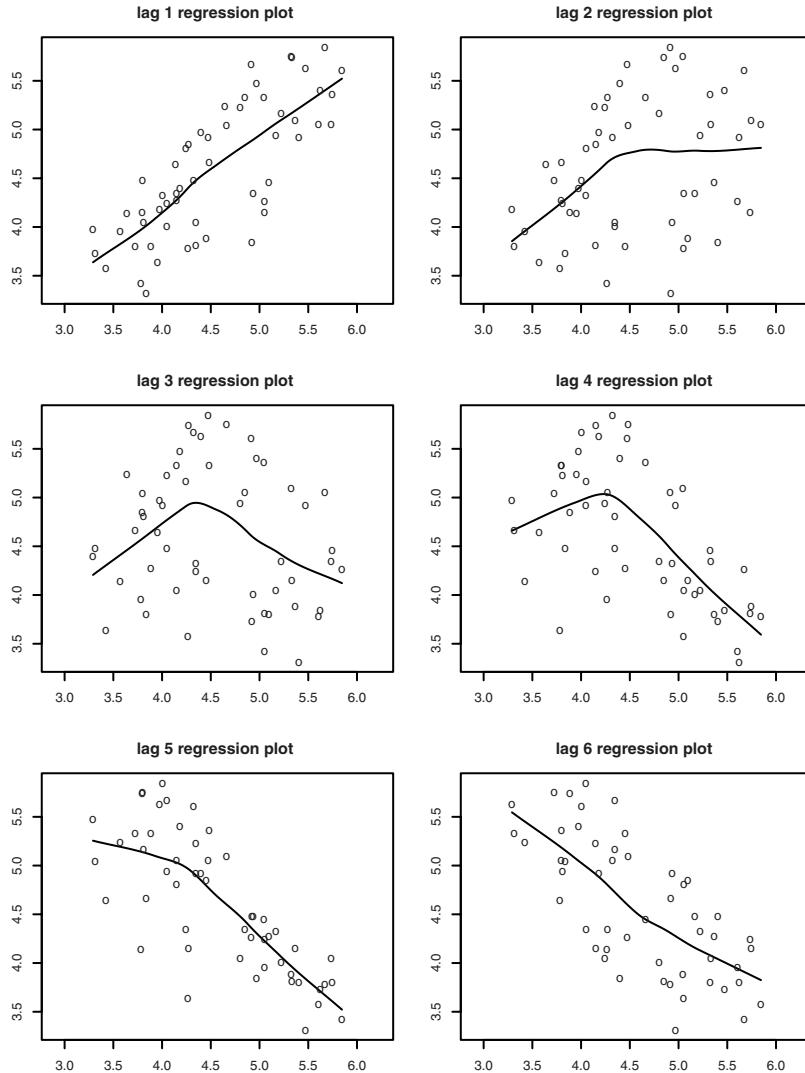
population fluctuation of a prey-predator system; the prey is *Paramecium aurelia*, a unicellular ciliate protozon, whereas the predator species is *Didinium natsutum*. The initial part of the data appears to be nonstationary owing to transient effects. It can be seen that the increasing phase of the series is generally longer than that of the decreasing phase, suggesting that the time series is time-irreversible. Below, we shall omit the first 14 data points from the analysis; that is, only the (log-transformed) data corresponding to the solid curve in Exhibit 15.2 are used in subsequent analysis.

Exhibit 15.2 Logarithmically Transformed Number of Predators. The stationary part of the time series is displayed as a solid line. Solid circles indicate data in the lower regime of a fitted threshold autoregressive model.



```
> data(veilleux); predator=veilleux[,1]
> win.graph(width=4.875,height=2.5,pointsize=8)
> plot(log(predator),lty=2,type='o',xlab='Day',
+       ylab='Log (predator)')
> predator.eq>window(predator,start=c(7,1))
> lines(log(predator.eq))
> index1=zlag(log(predator.eq),3)<=4.661
> points(y=log(predator.eq)[index1],(time(predator.eq))[index1],
+          pch=19)
```

Exhibit 15.3 shows the lagged regression plots of the predator series. Notice that several scatter diagrams have a large hole in the center, hinting that the data need to be nonnormal. Also, the regression function estimates appear to be strongly nonlinear for lags 2 to 4, suggesting a nonlinear data mechanism; in fact, the histogram (not shown) suggests that the series is bimodal.

Exhibit 15.3 Lagged Regression Plots for the Predator Series


```
> win.graph(width=4.875, height=6.5, pointsize=8)
> data(predator.eq)
> lagplot(log(predator.eq)) # libraries mgcv and locfit required
```

We now elaborate on how the regression curves are estimated nonparametrically. Readers not interested in the technical details may skip to the next section. For concreteness, suppose we want to estimate the lag 1 regression function. (The extension to other lags is straightforward.) Nonparametric estimation of the lag 1 regression function gen-

erally makes use of the idea of estimating the conditional mean $m_1(y) = E(Y_t|Y_{t-1} = y)$ by averaging those Y 's whose lag 1 values are close to y . Clearly, the averaging may be rendered more accurate by giving more weight to those Y 's whose lag 1 value is closer to y . The weights are usually assigned systematically via some probability density function $k(y)$ and a bandwidth parameter $h > 0$. The data pair (Y_t, Y_{t-1}) is assigned the weight

$$w_t = \frac{1}{h} k\left(\frac{Y_{t-1} - y}{h}\right) \quad (15.1.2)$$

Hereafter we assume that $k(\cdot)$ is the standard normal probability density function. Note that then the right-hand side of Equation (15.1.2) is the normal probability density function with mean y and variance h^2 . Finally, we define the Nadaraya-Watson estimator[†]

$$\hat{m}_1^{(0)}(y) = \frac{\sum_{t=2}^n w_t Y_t}{\sum_{t=2}^n w_t} \quad (15.1.3)$$

(The meaning of the superscript 0 will become clear later on.) Since the normal probability density function is negligible for values that differ from the mean by more than three standard deviations, the Nadaraya-Watson estimator essentially averages the Y_t whose Y_{t-1} is within $3h$ units from y , and the averaging is weighted with more weight to those observations whose lag 1 values are closer to y . The use of the Nadaraya-Watson estimator of the lag 1 regression function requires us to specify the bandwidth. There are several methods, including cross-validation for determining h . However, for an exploratory analysis, we can always use some default bandwidth value and vary it a bit to get some feel of the shape of the lag 1 regression function.

A more efficient nonparametric estimator may be obtained by assuming that the underlying regression function can be well-approximated locally by a linear function; see Fan and Gijbels (1996). The local linear estimator of the lag 1 regression function at y equals $\hat{m}_1^{(1)}(y) = b_0$, which is obtained by minimizing the local weighted residual sum of squares:

$$\sum_{t=2}^n w_t (Y_t - b_0 - b_1 Y_{t-1})^2 \quad (15.1.4)$$

The reader may now guess that the superscript k in the notation $\hat{m}_1^{(k)}(y)$ refers to the degree of the local polynomial. Often, data are unevenly spaced, in which case a single bandwidth may not work well. Instead, a variable bandwidth tied to the density of the data may be more efficient. A simple scheme is the nearest-neighbor scheme that varies the window width so that it covers a fixed fraction of data nearest to the center of the window. We set the fraction to be 70% for all our reported lagged regression plots.

[†] See Nadaraya (1964) and Watson (1964).

It is important to remember that the local polynomial approach assumes that the true lag 1 regression function is a smooth function. If the true lag 1 regression function is discontinuous, then the local polynomial approach may yield misleading estimates. However, a sharp turn in the estimated regression function may serve as a warning that the smoothness condition may not hold for the true lag 1 regression function.

15.2 Tests for Nonlinearity

Several tests have been proposed for assessing the need for nonlinear modeling in time series analysis. Some of these tests, such as those studied by Keenan (1985), Tsay (1986), and Luukkonen et al. (1988), can be interpreted as Lagrange multiplier tests for specific nonlinear alternatives.

Keenan (1985) derived a test for nonlinearity analogous to Tukey's one degree of freedom for nonadditivity test (see Tukey, 1949). Keenan's test is motivated by approximating a nonlinear stationary time series by a second-order Volterra expansion (Wiener, 1958)

$$Y_t = \mu + \sum_{\mu=-\infty}^{\infty} \theta_{\mu} \varepsilon_{t-\mu} + \sum_{v=-\infty}^{\infty} \sum_{\mu=-\infty}^{\infty} \theta_{\mu v} \varepsilon_{t-\mu} \varepsilon_{t-v} \quad (15.2.1)$$

where $\{\varepsilon_t, -\infty < t < \infty\}$ is a sequence of independent and identically distributed zero-mean random variables. The process $\{Y_t\}$ is linear if the double sum on the right-hand side of (15.2.1) vanishes. Thus, we can test the linearity of the time series by testing whether or not the double sum vanishes. In practice, the infinite series expansion has to be truncated to a finite sum. Let Y_1, \dots, Y_n denote the observations. Keenan's test can be implemented as follows:

- (i) Regress Y_t on Y_{t-1}, \dots, Y_{t-m} , including an intercept term, where m is some pre-specified positive integer; calculate the fitted values $\{\hat{Y}_t\}$ and the residuals $\{\hat{e}_t\}$, for $t = m+1, \dots, n$; and set $RSS = \sum \hat{e}_t^2$, the residual sum of squares.
- (ii) Regress \hat{Y}_t^2 on Y_{t-1}, \dots, Y_{t-m} , including an intercept term, and calculate the residuals $\{\hat{\xi}_t\}$ for $t = m+1, \dots, n$.
- (iii) Regress \hat{e}_t on the residuals $\hat{\xi}_t$ without an intercept for $t = m+1, \dots, n$, and Keenan's test statistic, denoted by \hat{F} , is obtained by multiplying $(n-2m-2)/(n-m-1)$ to the F -statistic for testing that the last regression function is identically zero. Specifically, let

$$\eta = \eta_0 \sqrt{\sum_{t=m+1}^n \hat{\xi}_t^2} \quad (15.2.2)$$

where η_0 is the regression coefficient. Form the test statistic

$$\hat{F} = \frac{\eta^2(n-2m-2)}{RSS - \eta^2} \quad (15.2.3)$$

Under the null hypothesis of linearity, the test statistic \hat{F} is approximately distributed as an F -distribution with degrees of freedom 1 and $n - 2m - 2$.

Keenan's test can be derived heuristically as follows. Consider the following model.

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \dots + \phi_m Y_{t-m} + \exp\left\{\eta\left(\sum_{j=1}^m \phi_j Y_{t-j}\right)^2\right\} + \varepsilon_t \quad (15.2.4)$$

where $\{\varepsilon_t\}$ are independent and normally distributed with zero mean and finite variance. If $\eta = 0$, the exponential term becomes 1 and can be absorbed into the intercept term so that the preceding model becomes an AR(m) model. On the other hand, for non-zero η , the preceding model is nonlinear. Using the expansion $\exp(x) \approx 1 + x$, which holds for x of small magnitude, it can be seen that, for small η , Y_t follows approximately a quadratic AR model:

$$Y_t = \theta_0 + 1 + \phi_1 Y_{t-1} + \dots + \phi_m Y_{t-m} + \eta\left(\sum_{j=1}^m \phi_j Y_{t-j}\right)^2 + \varepsilon_t \quad (15.2.5)$$

This is a restricted linear model in that the last covariate is the square of the linear term $\phi_1 Y_{t-1} + \dots + \phi_m Y_{t-m}$, which is replaced by the fitted values \hat{Y}_t under the null hypothesis. Keenan's test is equivalent to testing $\eta = 0$ in the multiple regression model (with the constant 1 being absorbed into θ_0):

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \dots + \phi_m Y_{t-m} + \eta \hat{Y}_t^2 + \varepsilon_t \quad (15.2.6)$$

which can be carried out in the manner described in the beginning of this section. Note that the fitted values are only available for $n \geq t \geq m + 1$. Keenan's test is the same as the F -test for testing whether or not $\eta = 0$. A more formal approach is facilitated by the Lagrange multiplier test; see Tong (1990).

Keenan's test is both conceptually and computationally simple and only has one degree of freedom, which makes the test very useful for small samples. However, Keenan's test is powerful only for detecting nonlinearity in the form of the square of the approximating linear conditional mean function. Tsay (1986) extended Keenan's approach by considering more general nonlinear alternatives. A more general alternative to nonlinearity may be formulated by replacing the term

$$\exp\left\{\eta\left(\sum_{j=1}^m \phi_j Y_{t-j}\right)^2\right\} \quad (15.2.7)$$

by

$$\left. \begin{aligned} & \exp(\delta_{1,1} Y_{t-1}^2 + \delta_{1,2} Y_{t-1} Y_{t-2} + \dots + \delta_{1,m} Y_{t-1} Y_{t-m} \\ & + \delta_{2,2} Y_{t-2}^2 + \delta_{2,3} Y_{t-2} Y_{t-3} + \dots + \delta_{2,m} Y_{t-2} Y_{t-m} + \dots \\ & + \delta_{m-1,m-1} Y_{t-m+1}^2 + \delta_{m-1,m} Y_{t-m+1} Y_{t-m} + \delta_{m,m} Y_{t-m}^2) + \varepsilon_t \end{aligned} \right\} \quad (15.2.8)$$

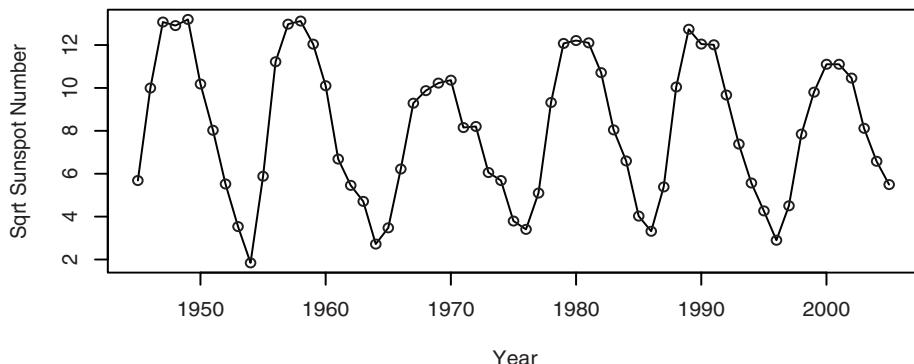
Using the approximation $\exp(x) \approx 1 + x$, we see that the nonlinear model is approximately a quadratic AR model. But the coefficients of the quadratic terms are now unconstrained. Tsay's test is equivalent to considering the following quadratic regression model:

$$\left. \begin{aligned} Y_t &= \theta_0 + \phi_1 Y_{t-1} + \cdots + \phi_m Y_{t-m} \\ &\quad + \delta_{1,1} Y_{t-1}^2 + \delta_{1,2} Y_{t-1} Y_{t-2} + \cdots + \delta_{1,m} Y_{t-1} Y_{t-m} \\ &\quad + \delta_{2,2} Y_{t-2}^2 + \delta_{2,3} Y_{t-2} Y_{t-3} + \cdots + \delta_{2,m} Y_{t-2} Y_{t-m} + \cdots \\ &\quad + \delta_{m-1,m-1} Y_{t-m+1}^2 + \delta_{m-1,m} Y_{t-m+1} Y_{t-m} + \delta_{m,m} Y_{t-m}^2 + \varepsilon_t \end{aligned} \right\} \quad (15.2.9)$$

and testing whether or not all the $m(m + 1)/2$ coefficients $\delta_{i,j}$ are zero. Again, this can be carried out by an F -test that all $\delta_{i,j}$'s are zero in the preceding equation. For a rigorous derivation of Tsay's test as a Lagrange multiplier test, see Tong (1990).

We now illustrate these tests with two real datasets. In the first application, we use the annual American (relative) sunspot numbers collected from 1945 to 2007. The annual (relative) sunspot number is a weighted average of solar activities measured from a network of observatories. Historically, the daily sunspot number was computed as some weighted sum of the count of visible, distinct spots and that of clusters of spots on the solar surface. The sunspot number reflects the intensity of solar activity. Below, the sunspot data are square root transformed to make them more normally distributed; see Exhibit 15.4. The time series plot shows that the sunspot series tends to rise up more quickly than when it declines, suggesting that it is time-irreversible.

Exhibit 15.4 Annual American Relative Sunspot Numbers



```
> win.graph(width=4.875, height=2.5, pointsize=8)
> data(spots)
> plot(sqrt(spots), type='o', xlab='Year',
      ylab='Sqrt Sunspot Number')
```

To carry out the tests for nonlinearity, we have to specify m , the working autoregressive order. Under the null hypothesis that the process is linear, the order can be specified by using some information criterion, for example, the AIC. For the sunspot data, $m = 5$ based on the AIC. Both the Keenan test and the Tsay test reject linearity, with p -values being 0.0002 and 0.0009, respectively.

For the second example, we consider the predator series discussed in the preceding section. The working AR order is found to be 4. Both the Keenan test and the Tsay test reject linearity, with p -values being 0.00001 and 0.03, respectively, which is consistent with the inference drawn from the lagged regression plots reported earlier.

There are some other tests, such as the BDS test developed by Brock, Deckert and Seheinkman (1996), based on concepts that arise in the theory of chaos, and the neural-network test, proposed by White (1989) for testing “neglected nonlinearity.” For a recent review of tests for nonlinearity, see Tong (1990) and Granger and Teräsvirta (1993). We shall introduce one more test later.

15.3 Polynomial Models Are Generally Explosive

In nonlinear regression analysis, polynomial regression models of higher degrees are sometimes employed, even though they are deemed not useful for extrapolation because of their quick blowup to infinity. For this reason, polynomial regression models are of limited practical use. Based on the same reasoning, polynomial time series models may be expected to do poorly in prediction. Indeed, polynomial time series models of degree higher than 1 and with Gaussian errors are invariably explosive. To see this, consider the following simple quadratic AR(1) model.

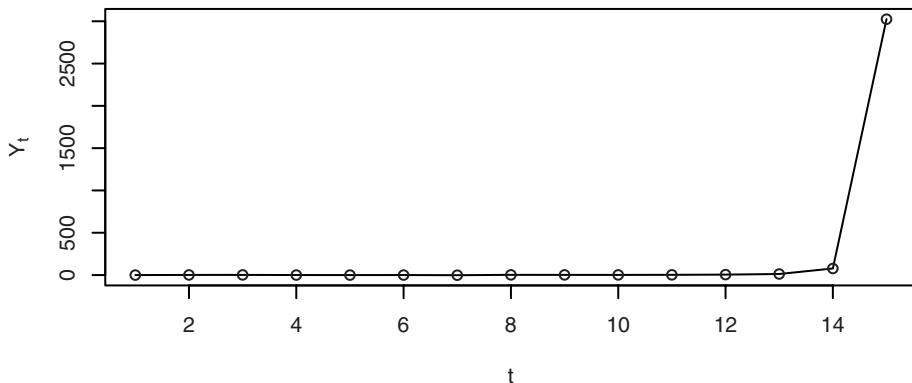
$$Y_t = \phi Y_{t-1}^2 + e_t \quad (15.3.1)$$

where $\{e_t\}$ are independent and identically distributed standard normal random variables. Let $\phi > 0$ and let c be a large number that is greater than $3/\phi$. If $Y_1 > c$ (which may happen with positive probability due to the normality of the errors), then $Y_2 > 3Y_1 + e_2$ and hence $Y_2 > 2c$ with some nonzero probability. With careful probability analysis, it can be shown that, with positive probability, the quadratic AR(1) process satisfies the inequality $Y_t > 2^t c$ for $t = 1, 2, 3, \dots$ and hence blows up to $+\infty$. Indeed, the quadratic AR(1) process, with normal errors, goes to infinity with probability 1.

As an example, Exhibit 15.5 displays a realization from a quadratic AR(1) model with $\phi = 0.5$ and standard normal errors that takes off to infinity at $t = 15$.

Note that the quadratic AR(1) process becomes explosive only when the process takes some value of sufficiently large magnitude. If the coefficient ϕ is small, it may take much longer for the quadratic AR(1) process to take off to infinity. Normal errors can take arbitrarily large values, although rather rarely, but when this happens, the process becomes explosive. Thus, any noise distribution that is unbounded will guarantee the explosiveness of the quadratic AR(1) model. Chan and Tong (1994) further showed that this explosive behavior is true for any polynomial autoregressive process of degree higher than 1 and of any finite order when the noise distribution is unbounded.

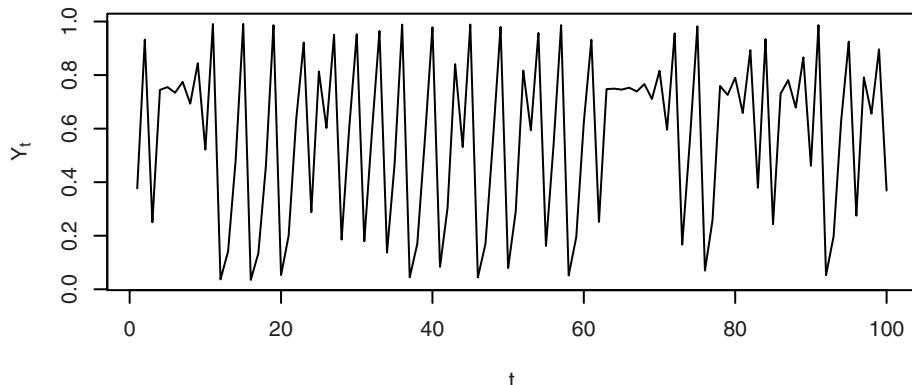
Exhibit 15.5 A Simulated Quadratic AR(1) Process with $\phi = 0.5$



```
> set.seed(1234567)
> plot(y=qar.sim(n=15,phi1=.5,sigma=1),x=1:15,type='o',
+       ylab=expression(Y[t]),xlab='t')
```

It is interesting to note that, for bounded errors, a polynomial autoregressive model may admit a stationary distribution that could be useful for modeling nonlinear time series data; see Chan and Tong (1994). For example, Exhibit 15.6 displays the time series solution of a deterministic logistic map, namely $Y_t = 3.97Y_{t-1}(1 - Y_{t-1})$, $t = 2, 3, \dots$ with the initial value $Y_1 = 0.377$. Its corresponding sample ACF is shown in Exhibit 15.7, which, except for the mildly significant lag 4, resembles that of white noise. Note that, for a sufficiently large initial value, the solution of the logistic map will explode to infinity.

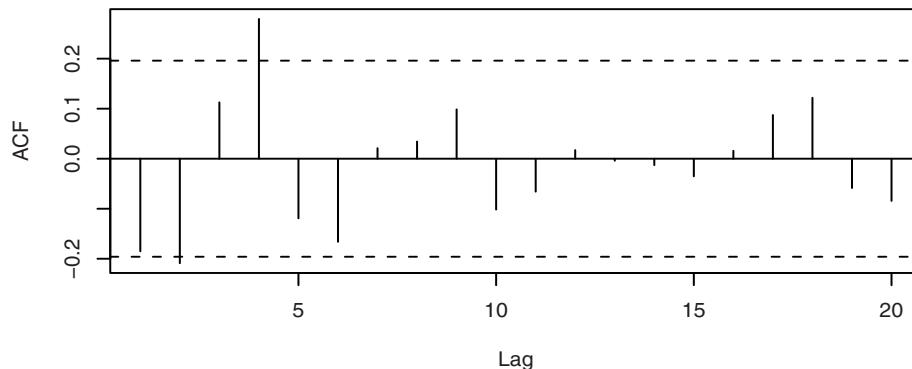
Exhibit 15.6 The Trajectory of the Logistic Map with Parameter 3.97 and Initial Value $Y_1 = 0.377$



```
> y=qar.sim(n=100,const=0.0,phi0=3.97,phi1=-3.97,sigma=0,
+            init=.377)
```

```
> plot(x=1:100,y=y,type='l',ylab=expression(Y[t]),xlab='t')
```

Exhibit 15.7 Sample ACF of the Logistic Time Series



```
> acf(y)
```

However, the bound on the noise distribution necessary for the existence of a stationary polynomial autoregressive model varies with the model parameters and the initial value, which greatly complicates the modeling task. Henceforth, we shall not pursue the use of polynomial models in time series analysis.

15.4 First-Order Threshold Autoregressive Models

The discussion in the preceding section provides an important insight that for a nonlinear time series model to be stationary, it must be either linear or approaching linearity in the “tail.” From this perspective, piecewise linear models, more widely known as threshold models, constitute the simplest class of nonlinear model. Indeed, the usefulness of threshold models in nonlinear time series analysis was well-documented by the seminal work of Tong (1978, 1983, 1990) and Tong and Lim (1980), resulting in an extensive literature of ongoing theoretical innovations and applications in various fields.

The specification of a threshold model requires specifying the number of linear submodels and the mechanism dictating which of them is operational. Consequently, there exist many variants of the threshold model. Here, we focus on the two-regime self-exciting threshold autoregressive (SETAR) model introduced by Tong, for which the switching between the two linear submodels depends solely on the position of the threshold variable. For the SETAR model (simply referred to as the TAR model below), the threshold variable is a certain lagged value of the process itself; hence the adjective self-exciting. (More generally, the threshold variable may be some vector covariate process or even some latent process, but this extension will not be pursued here.) To fix ideas, consider the following first-order TAR model:

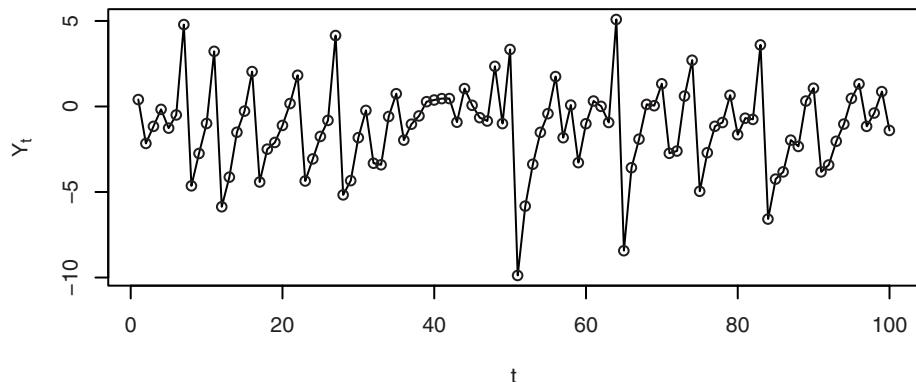
$$Y_t = \begin{cases} \phi_{1,0} + \phi_{1,1} Y_{t-1} + \sigma_1 e_t & \text{if } Y_{t-1} \leq r \\ \phi_{2,0} + \phi_{2,1} Y_{t-1} + \sigma_2 e_t & \text{if } Y_{t-1} > r \end{cases} \quad (15.4.1)$$

where the ϕ 's are autoregressive parameters, σ 's are noise standard deviations, r is the threshold parameter, and $\{e_t\}$ is a sequence of independent and identically distributed random variables with zero mean and unit variance. Thus, if the lag 1 value of Y_t is not greater than the threshold, the conditional distribution of Y_t is the same as that of an AR(1) process with intercept $\phi_{1,0}$, autoregressive coefficient $\phi_{1,1}$, and error variance σ_1^2 , in which case we may say that the first AR(1) submodel is operational. On the other hand, when the lag 1 value of Y_t exceeds the threshold r , the second AR(1) process with parameters $(\phi_{2,0}, \phi_{2,1}, \sigma_2^2)$ is operational. Thus, the process switches between two linear mechanisms dependent on the position of the lag 1 value of the process. When the lag 1 value does not exceed the threshold, we say that the process is in the lower (first) regime, and otherwise it is in the upper regime. Note that the error variance need not be identical for the two regimes, so that the TAR model can account for some conditional heteroscedasticity in the data.

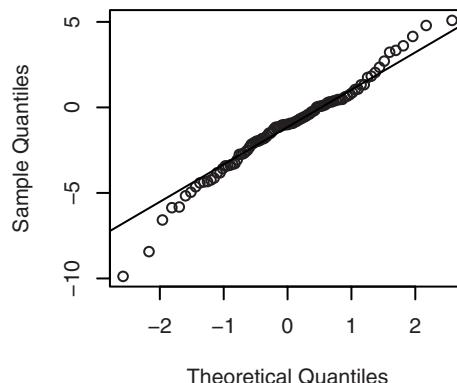
As a concrete example, we simulate some data from the following first-order TAR model:

$$Y_t = \begin{cases} 0.5 Y_{t-1} + e_t & \text{if } Y_{t-1} \leq -1 \\ -1.8 Y_{t-1} + 2e_t & \text{if } Y_{t-1} > -1 \end{cases} \quad (15.4.2)$$

Exhibit 15.8 shows the time series plot of the simulated data of size $n = 100$. A notable feature of the plot is that the time series is somewhat cyclical, with asymmetrical cycles where the series tends to drop rather sharply but rises relatively slowly. This asymmetry means that the probabilistic structure of the process will be different if we reverse the direction of time. One way to see this is to make a transparency of the time series plot and flip the transparency over to see the time series plot with time reversed. In this case, the simulated data will rise sharply and drop slowly with time reversed. Recall that this phenomenon is known as time irreversibility. For a stationary Gaussian ARMA process, the probabilistic structure is determined by its first and second moments, which are invariant with respect to time reversal, hence the process must be time-reversible. Many real time series, for example the predator series and the relative sunspot series, appear to be time-irreversible, suggesting that the underlying process is nonlinear. Exhibit 15.9 shows the QQ normal score plot for the simulated data. It shows that the distribution of simulated data has a thicker tail than a normal distribution, despite the fact that the errors are normally distributed.

Exhibit 15.8 A Simulated First-Order TAR Process

```
> set.seed(1234579)
> y=tar.sim(n=100,Phi1=c(0,0.5),Phi2=c(0,-1.8),p=1,d=1,sigma1=1,
  thd=-1,sigma2=2)$y
> plot(y=y,x=1:100,type='o',xlab='t',ylab=expression(Y[t]))
```

Exhibit 15.9 QQ Normal Plot for the Simulated TAR Process

```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(y); qqline(y)
```

The autoregressive coefficient of the submodel in the upper regime equals -1.8 , yet the simulated data appear to be stationary, which may be unexpected from a linear perspective, as an AR(1) model cannot be stationary if the autoregressive coefficient exceeds 1 in magnitude. This puzzle may be better understood by considering the case of no noise terms in either regime; that is, $\sigma_1 = \sigma_2 = 0$. The deterministic process thus defined is referred to as the *skeleton* of the TAR model. We show below that, for any ini-

tial value, the skeleton is eventually a bounded process; the stability of the skeleton underlies the stationarity of the TAR model. Readers not interested in the detailed analysis verifying the ultimate boundedness of the skeleton may skip to the next paragraph. Let the initial value y_1 be some large number, say 10, a value falling in the upper regime. So, the next value is $y_2 = (-1.8) \times 10 = -18$, which is in the lower regime. Therefore, the third value equals $y_3 = 0.5 \times (-18) = -9$. As the third value is in the lower regime, the fourth value equals $y_4 = 0.5 \times (-9) = -4.5$, which remains in the lower regime, so that the fifth value equals $y_5 = 0.5 \times (-4.5) = -2.25$. It is clear that once the data remain in the lower regime, they will be halved in the next iterate and this process continues until some future iterate crosses the threshold -1 , which occurs for $y_7 = -0.5625$. Now the second linear submodel is operational, so that $y_8 = (-1.8) \times (-0.5625) = 1.0125$ and $y_9 = (-1.8) \times 1.0125 = -1.8225$, which is again in the lower regime. In conclusion, if some iterate is in the lower regime, the next iterate is obtained by halving the previous iterate until some future iterate exceeds -1 . On the other hand, if some iterate exceeds 1, the next iterate must be less than -1 and hence in the lower regime. By routine analysis, it can be checked that the process is eventually trapped between -1 and 1.8 and hence is a bounded process.

A bounded skeleton is stable in some sense. Chan and Tong (1985), showed that under some mild conditions, a TAR model is asymptotically stationary if its skeleton is stable. In fact, stability of the skeleton together with some regularity conditions imply the stronger property of *ergodicity*; namely, the process admits a stationary distribution and for any function $h(Y_t)$ having a finite stationary first moment (which holds if h is a bounded function),

$$\frac{1}{n} \sum_{t=1}^n h(Y_t) \quad (15.4.3)$$

converges to the stationary mean of $h(Y_t)$, computed according to the stationary distribution. See Cline and Pu (2001) for a recent survey on the linkage between stability and ergodicity and counterexamples when this linkage may fail to hold.

The stability analysis of the skeleton can be much simplified by the fact that the ergodicity of a TAR model can be inferred from the stability of an associated skeleton defined by a difference equation obtained by modifying the equation defining the TAR model by suppressing the noise terms and the intercepts (that is, zero errors and zero intercepts) and setting the threshold to 0. For the simulated example, the associated skeleton is then defined by the following difference equation:

$$Y_t = \begin{cases} 0.5Y_{t-1}, & \text{if } Y_{t-1} \leq 0 \\ -1.8Y_{t-1}, & \text{if } Y_{t-1} > 0 \end{cases} \quad (15.4.4)$$

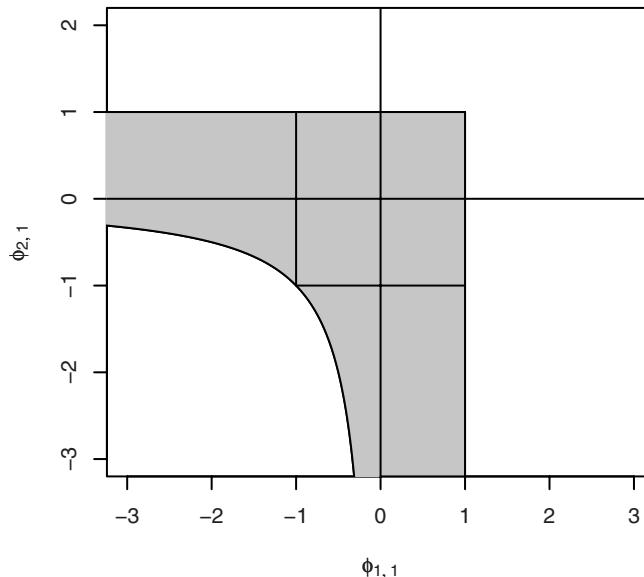
Now, the solution to the skeleton above can be readily obtained: Given a positive value for y_1 , $y_t = (-1.8) \times 0.5^{t-2} \times y_1$, for all $t \geq 2$. For negative y_1 , $y_t = 0.5^{t-1} \times y_1$. In both cases, $y_t \rightarrow 0$, as $t \rightarrow \infty$. The origin is said to be an *equilibrium point* as $y_t \equiv 0$, for all t , if $y_1 = 0$. The origin is then said to be a *globally exponentially stable limit point*, as the skeleton approaches it exponentially fast for any nonzero initial value. It can be shown (Chan and

Tong, 1985) that the origin is a globally exponentially stable limit point for the skeleton if the parameters satisfy the constraints

$$\phi_{1,1} < 1, \phi_{2,1} < 1, \phi_{1,1}\phi_{2,1} < 1 \quad (15.4.5)$$

in which case the first-order TAR model is ergodic and hence stationary. Exhibit 15.10 shows the region of stationarity shaded in gray. Note that the region of stationarity is substantially larger than the region defined by the linear time series inspired constraints $|\phi_{1,1}| < 1, |\phi_{2,1}| < 1$, corresponding to the region bounded by the inner square in Exhibit 15.10. For parameters lying strictly outside the region defined by the constraints (Equations (15.4.5)), the skeleton is unstable and the TAR model is nonstationary. For example, if $\phi_{2,1} > 1$, then the skeleton will escape to positive infinity for all sufficiently large initial values. On the boundary of the parametric region defined by (15.4.5), the intercept terms of the TAR model are pivotal in determining the stability of the skeleton and the stationarity of the TAR models; see Chan et al. (1985). In practice, we can check if the skeleton is stable numerically by using several different initial values. A stable skeleton gives us more confidence in assuming that the model is stationary.

Exhibit 15.10 Stationarity Region for the First-Order TAR Model (Shaded)



15.5 Threshold Models

The first-order (self-exciting) threshold autoregressive model can be readily extended to higher order and with a general integer delay:

$$Y_t = \begin{cases} \phi_{1,0} + \phi_{1,1} Y_{t-1} + \cdots + \phi_{1,p_1} Y_{t-p_1} + \sigma_1 e_t & \text{if } Y_{t-d} \leq r \\ \phi_{2,0} + \phi_{2,1} Y_{t-1} + \cdots + \phi_{2,p_2} Y_{t-p_2} + \sigma_2 e_t & \text{if } Y_{t-d} > r \end{cases} \quad (15.5.1)$$

Note that the autoregressive orders p_1 and p_2 of the two submodels need not be identical, and the delay parameter d may be larger than the maximum autoregressive orders. However, by including zero coefficients if necessary, we may and shall henceforth assume that $p_1 = p_2 = p$ and $1 \leq d \leq p$, which simplifies the notation. The TAR model defined by Equation (15.5.1) is denoted as the $\text{TAR}(2; p_1, p_2)$ model with delay d .

Again, the stability of the associated skeleton, obtained by setting the threshold to zero and suppressing the noise terms and the intercepts, implies that the TAR model is ergodic and stationary. However, the stability of the associated skeleton is now much more complex in the higher-order case so much so that the necessary and sufficient parametric conditions for the stationarity of the TAR model are still unknown. Nonetheless, there exist some simple sufficient conditions for the stationarity of a TAR model. For example, the TAR model is ergodic and hence asymptotically stationary if $|\phi_{1,1}| + \cdots + |\phi_{1,p}| < 1$ and $|\phi_{2,1}| + \cdots + |\phi_{2,p}| < 1$; see Chan and Tong (1985).

So far, we have considered the case of two regimes defined by the partition $-\infty < r < \infty$ of the real line, so that the first (second) submodel is operational if Y_{t-d} lies in the first (second) interval. The extension to the case of m regimes is straightforward and effected by partitioning the real line into $-\infty < r_1 < r_2 < \cdots < r_{m-1} < \infty$, and the position of Y_{t-d} relative to these thresholds determines which linear submodel is operational. We shall not pursue this topic further but shall restrict our discussion to the case of two regimes.

15.6 Testing for Threshold Nonlinearity

While Keenan's test and Tsay's test for nonlinearity are designed for detecting quadratic nonlinearity, they may not be sensitive to threshold nonlinearity. Here, we discuss a likelihood ratio test with the threshold model as the specific alternative. The null hypothesis is an $\text{AR}(p)$ model versus the alternative hypothesis of a two-regime TAR model of order p and with constant noise variance, that is; $\sigma_1 = \sigma_2 = \sigma$. With these assumptions, the general model can be rewritten as

$$Y_t = \phi_{1,0} + \phi_{1,1} Y_{t-1} + \cdots + \phi_{1,p} Y_{t-p} + \{\phi_{2,0} + \phi_{2,1} Y_{t-1} + \cdots + \phi_{2,p} Y_{t-p}\} I(Y_{t-d} > r) + \sigma e_t \quad (15.6.1)$$

where the notation $I(\cdot)$ is an indicator variable that equals 1 if and only if the enclosed expression is true. Moreover, in this formulation, the coefficient $\phi_{2,0}$ represents the change in the intercept in the upper regime relative to that of the lower regime, and similarly interpreted are $\phi_{2,1}, \dots, \phi_{2,p}$. The null hypothesis states that $\phi_{2,0} = \phi_{2,1} = \cdots = \phi_{2,p} = 0$. While the delay may be theoretically larger than the autoregressive order, this is seldom the case in practice. Hence, it is assumed that $d \leq p$ throughout this section, and

under this assumption and assuming the validity of linearity, the large-sample distribution of the test does not depend on d .

In practice, the test is carried out with fixed p and d . The likelihood ratio test statistic can be shown to be equivalent to

$$T_n = (n - p) \log \left\{ \frac{\hat{\sigma}^2(H_0)}{\hat{\sigma}^2(H_1)} \right\} \quad (15.6.2)$$

where $n - p$ is the effective sample size, $\hat{\sigma}^2(H_0)$ is the maximum likelihood estimator of the noise variance from the linear AR(p) fit and $\hat{\sigma}^2(H_1)$ from the TAR fit with the threshold searched over some finite interval. See the next section for a detailed discussion on estimating a TAR model. Under the null hypothesis that $\phi_{2,0} = \phi_{2,1} = \dots = \phi_{2,p} = 0$, the (nuisance) parameter r is absent. Hence, the sampling distribution of the likelihood ratio test under H_0 is no longer approximately χ^2 with p degrees of freedom. Instead, it has a nonstandard sampling distribution; see Chan (1991) and Tong (1990). Chan (1991) derived an approximation method for computing the p -values of the test that is highly accurate for small p -values. The test depends on the interval over which the threshold parameter is searched. Typically, the interval is defined to be from the $a \times 100$ th percentile to the $b \times 100$ th percentile of $\{Y_t\}$, say from the 25th percentile to the 75th percentile. The choice of a and b must ensure that there are adequate data falling into each of the two regimes for fitting the linear submodels.

The reader may wonder why the search of the threshold is restricted to some finite interval. Intuitively, such a restriction is desirable, as we want enough data to estimate the parameters for the two regimes under the alternative hypothesis. A deeper reason is mathematical in nature. This restriction is necessary because if the true model is linear, the threshold parameter is undefined, in which case an unrestricted search may result in the threshold estimator being close to the minimum or maximum data values, making the large-sample approximation ineffective.

We illustrate the likelihood ratio test for threshold nonlinearity using the (square-root-transformed) relative sunspot data and the (log-transformed) predator data. Recall that both Keenan's test and Tsay's test suggested that these data are nonlinear. Setting $p = 5$, $a = 0.25$, and $b = 0.75$ for the sunspot data, we tried the likelihood ratio test for threshold nonlinearity with different delays from 1 to 5, resulting in the test statistics being 46.9, 111.3, 99.1, 85.0, and 45.1, respectively.[†] Repeating the test with $a = 0.1$ and $b = 0.9$ yields identical results for this case. All the tests above have p -values less than 0.000, suggesting that the data-generating mechanism is highly nonlinear. Notice that the test statistic attains the largest value when $d = 2$; hence we may tentatively estimate

[†] The R code to carry out these calculations is as follows:

```
> pvaluem=NULL
> for (d in 1:5) { res=tlrt(sqrt(spots),p=5,d=d,a=0.25,b=0.75)
> pvaluem= cbind( pvaluem, c(d,res$test.statistic,res$p.value)) }
> rownames(pvaluem)=c('d','test statistic','p-value')
> round(pvaluem, 3)
```

the delay to be 2. But delay 3 is very competitive.

Next, consider the predator series, with $p = 4$, $a = 0.25$, $b = 0.75$, and $1 \leq d \leq 4$. The test statistics and their p -values, enclosed in parentheses, are found to equal 19.3 (0.026), 28.0 (0.001), 32.0 (0.000), and 16.2 (0.073), respectively. Thus, there is some evidence that the predator series is nonlinear, with the delay likely to be 2 or 3. Note that the test is not significant for $d = 4$ at the 5% significance level.[†]

15.7 Estimation of a TAR Model

Because the stationary distribution of a TAR model does not have a closed-form solution, estimation is often carried out conditional on the $\max(p, d)$ initial values, where p is the order of the process and d the delay parameter. Moreover, the noise series is often assumed to be normally distributed, and we will make this assumption throughout this section. The normal error assumption implies that the response is conditionally normal, but see Samia, Chan and Stenseth (2007) for some recent work on the nonnormal case. If the threshold parameter r and the delay parameter d are known, then the data cases can be split into two parts according to whether or not $Y_{t-d} \leq r$. Let there be n_1 data cases in the lower regime. With the data in the lower regime, we can regress Y_t on its lags 1 to p to find the estimates of $\hat{\phi}_{1,0}, \hat{\phi}_{1,1}, \dots, \hat{\phi}_{1,p}$ and the maximum likelihood noise variance estimate $\hat{\sigma}_1^2$; that is, the sum of squared residuals divided by n_1 . The number n_1 and the parameter estimates for the lower regime generally depend on r and d ; we sometimes write the more explicit notation, for example $n_1(r, d)$, below for clarity. Similarly, using the data, say n_2 of them, falling in the upper regime, we can obtain the parameter estimates $\hat{\phi}_{2,0}, \hat{\phi}_{2,1}, \dots, \hat{\phi}_{2,p}$ and $\hat{\sigma}_2^2$. Clearly, $n_1 + n_2 = n - p$, where n is the sample size. Substituting these estimates into the log-likelihood function yields the so-called *profile log-likelihood function* of (r, d) :

$$\begin{aligned} l(r, d) = & -\frac{n-p}{2}\{1 + \log(2\pi)\} - \frac{n_1(r, d)}{2}\log((\hat{\sigma}_1(r, d))^2) \\ & - \frac{n_2(r, d)}{2}\log((\hat{\sigma}_2(r, d))^2) \end{aligned} \quad (15.7.1)$$

The estimates of r and d can be obtained by maximizing the profile likelihood function above. The optimization need only be searched with r over the observed Y 's and integer d between 1 and p . This is because, for fixed d , the function above is constant between two consecutive observations.

However, without some restrictions on the threshold parameter, the (conditional) maximum likelihood method discussed above will not work. For example, if the lower regime contains only one data case, the noise variance $\hat{\sigma}_1^2 = 0$ so that the conditional log-likelihood function equals ∞ , in which case the conditional maximum likelihood estimator is clearly inconsistent. This problem may be circumvented by restricting the

[†] The R code for this calculation is similar to that shown on the previous page. The details may be found in the R code scripts for Chapter 15 available on the textbook Website.

search of the threshold to be between two predetermined percentiles of Y ; for example, between the tenth and ninetieth percentiles.

Another approach to handle the aforementioned difficulty is to estimate the parameters using the conditional least squares (CLS) approach. The CLS approach estimates the parameters by minimizing the predictive sum of squared errors, or equivalently conditional maximum likelihood estimation for the case of homoscedastic (constant-variance) Gaussian errors; that is, $\sigma_1 = \sigma_2 = \sigma$ so that maximizing the log-likelihood function is equivalent to minimizing the conditional residual sum of squares:

$$L(r, d) = \sum_{t=p+1}^n \{ (Y_t - \phi_{1,0} - \phi_{1,1}Y_{t-1} - \cdots - \phi_{1,p}Y_{t-p})^2 I(Y_{t-d} \leq r) + (Y_t - \phi_{2,0} - \phi_{2,1}Y_{t-1} - \cdots - \phi_{2,p}Y_{t-p})^2 I(Y_{t-d} > r) \} \quad (15.7.2)$$

where $I(Y_{t-d} \leq r)$ equals 1 if $Y_{t-d} \leq r$ and 0 otherwise; the expression $I(Y_{t-d} > r)$ is similarly defined. Again, the optimization need only be done with r searched over the observed Y 's and d an integer between 1 and p . The conditional least squares approach has the advantage that the threshold parameter can be searched without any constraints. Under mild conditions, including stationarity and that the true conditional mean function is a discontinuous function, Chan (1993) showed that the CLS method is consistent; that is, the estimator approaches the true value with increasing sample size. As the delay is an integer, the consistency property implies that the delay estimator is eventually equal to the true value with very large sample size. Furthermore, the sampling error of the threshold estimator is of the order $1/n$, whereas the sampling error of the other parameters is of order $1/\sqrt{n}$. The faster convergence of the threshold parameter and the delay parameter to their true values implies that in assessing the uncertainty of the autoregressive parameter estimates, the threshold and the delay may be treated as if they were known. Consequently, the autoregressive parameter estimators from the two regimes are approximately independent of each other, and their sampling distributions are approximately the same as those from the ordinary least squares regression with data from the corresponding true regimes. These large-sample distribution results can be lifted to the case of the conditional maximum likelihood estimator provided the true parameter satisfies the regularity conditions alluded to before. Finally, we note that the preceding large-sample properties of the estimator are radically different if the true conditional mean function is continuous; see Chan and Tsay (1998).

In practice, the AR orders in the two regimes need not be identical or known. Thus, an efficient estimation procedure that also estimates the orders is essential. Recall that for linear ARMA models, the AR orders can be estimated by minimizing the AIC. For fixed r and d , the TAR model is essentially fitting two AR models of orders p_1 and p_2 , respectively, so that the AIC becomes

$$\text{AIC}(p_1, p_2, r, d) = -2l(r, d) + 2(p_1 + p_2 + 2) \quad (15.7.3)$$

where the number of parameters, excluding r, d, σ_1 , and σ_2 , equals $p_1 + p_2 + 2$. Now, the minimum AIC (MAIC) estimation method estimates the parameters by minimizing the AIC subject to the constraint that the threshold parameter be searched over some inter-

val that guarantees any regimes have adequate data for estimation. Adding 2 to the minimum AIC so found is defined as the nominal AIC of the estimated threshold model, based on the naive idea of counting the threshold parameter as one additional parameter. Since the threshold parameter generally adds much flexibility to the model, it is likely to add more than one degree of freedom to the model. An asymptotic argument suggests that it may be equivalent to adding three degrees of freedom to the model; see Tong (1990, p. 248).

We illustrate the estimation methods with the predator series. In the estimation, the maximum order is set to be $p = 4$ and $1 \leq d \leq 4$. This maximum order is the AR order determined by AIC, which is likely to be not smaller than the order of the true TAR model. Alternatively, the order may be determined by cross-validation, which is computer-intensive; see Cheng and Tong (1992). Using the MAIC method with the search of threshold roughly between the tenth and ninetieth percentiles, the table in Exhibit 15.11 displays the nominal AIC value of the estimated TAR model for $1 \leq d \leq 4$. The nominal AIC is smallest when $d = 3$, so we estimate the delay to be 3. The table in Exhibit 15.12 summarizes the corresponding model fit.

Exhibit 15.11 Nominal AIC of the TAR Models Fitted to the Log(predator) Series for $1 \leq d \leq 4$

d	AIC	\hat{r}	\hat{p}_1	\hat{p}_2
1	19.04	4.15	2	1
2	12.15	4.048	1	4
3	10.92	4.661	1	4
4	18.42	5.096	3	4

```

> AICM=NULL
> for(d in 1:4)
+   {predator.tar=tar(y=log(predator.eq),p1=4,p2=4,d=d,a=.1,b=.9)
> AICM=rbind(AICM,
+   c(d,predator.tar$AIC,signif(predator.tar$thd,4),
+     predator.tar$p1,predator.tar$p2))}
> colnames(AICM)=c('d','nominal AIC','r','p1','p2')
> rownames(AICM)=NULL
> AICM

```

Although the maximum autoregressive order is 4, the MAIC method selects order 1 for the lower regime and order 4 for the upper regime. The submodel in each regime is estimated by ordinary least squares (OLS) using the data falling in that regime. Hence a less biased estimator of the noise variance may be estimated by the within-regime residual sum of squared errors normalized by the effective sample size which equals the number of data in that regime minus the number of autoregressive parameters (including the intercept) of the corresponding submodel. The “unbiased” noise variance $\tilde{\sigma}_i^2$ of the i th regime relates to its maximum likelihood counterpart by the formula

$$\tilde{\sigma}_i^2 = \frac{n_i}{n_i - p_i - 1} \hat{\sigma}_i^2, \quad (15.7.4)$$

where p_i is the autoregressive order of the i th submodel. Moreover, $(n_i - p_i - 1)\tilde{\sigma}_i^2/\sigma_i^2$ is approximately distributed as χ^2 with $n_i - p_i - 1$ degrees of freedom. For each regime, the t -statistics and corresponding p -values reported in Exhibit 15.12 are identical with the computer output for the case of fitting an autoregressive model with the data falling in that regime. Notice that the coefficients of lags 2 and 3 in the upper regime are not significant, while that of lag 4 is mildly significant at the 5% significance level. Hence, the model for the upper regime may be approximated by a first-order autoregressive model. We shall return to this point later.

Exhibit 15.12 Fitted TAR(2;1,4) Model for the Predator Data: MAIC Method

	Estimate	Std. Error	<i>t</i> -statistic	<i>p</i> -value
\hat{d}	3			
\hat{r}	4.661			
Lower Regime ($n_1 = 30$)				
$\hat{\phi}_{1,0}$	0.262	0.316	0.831	0.41
$\hat{\phi}_{1,1}$	1.02	0.0704	14.4	0.00
$\tilde{\sigma}_1^2$	0.0548			
Upper Regime ($n_2 = 23$)				
$\hat{\phi}_{2,0}$	4.20	1.28	3.27	0.00
$\hat{\phi}_{2,1}$	0.708	0.202	3.50	0.00
$\hat{\phi}_{2,2}$	-0.301	0.312	-0.965	0.35
$\hat{\phi}_{2,3}$	0.279	0.406	0.686	0.50
$\hat{\phi}_{2,4}$	-0.611	0.273	-2.24	0.04
$\tilde{\sigma}_2^2$	0.0560			

```
> predator.tar.1=tar(y=log(predator.eq), p1=4, p2=4, d=3, a=.1, b=.9,
+ print=T)
> tar(y=log(predator.eq), p1=1, p2=4, d=3, a=.1, b=.9, print=T,
+ method='CLS') # re-do the estimation using the CLS method
> tar(y=log(predator.eq), p1=4, p2=4, d=3, a=.1, b=.9, print=T,
+ method='CLS') # the CLS method does not estimate the AR orders
```

The threshold estimate is 4.661, roughly the 57th percentile. In general, a threshold estimate that is too close to the minimum or the maximum observation may be unreliable due to small sample size in one of the regimes, which, fortunately, is not the case

here. Exhibit 15.12 does not report the standard error of the threshold estimate because its sampling distribution is nonstandard and rather complex. Similarly, the discreteness of the delay estimator renders its standard error useless. However, a parametric bootstrap may be employed to draw inferences on the threshold and the delay parameters. An alternative is to adopt the Bayesian approach of Geweke and Terui (1993). In contrast, the fitted AR(4) model has the coefficient estimates of lags 1 to 4 equal to 0.943 (0.136), -0.171 (0.188), -0.1621 (0.186), and -0.238 (0.136), respectively, with their standard errors enclosed in parentheses; the noise variance is estimated to be 0.0852, which is substantially larger than the noise variances of the TAR(2;1,4) model. Notice that the AR(4) coefficient estimate is close to being nonsignificant, and the AR(2) and AR(3) coefficient estimates are not significant.

An interesting question concerns the interpretation of the two regimes. One way to explore the nature of the regimes is to identify which data value falls in which regime in the time series plot of the observed process. In the time series plot in Exhibit 15.2 on page 387, data falling in the lower regime (that is, those whose lag 3 values are less than 4.661) are drawn as solid circles, whereas those in the upper regime are displayed as open circles. The plot reveals that the estimated lower regime corresponds to the increasing phase of the predator cycles and the upper regime corresponds to the decreasing phase of the predator cycles. A biological interpretation is the following. When the predator number was low one and a half days earlier, the prey species would have been able to increase in the intervening period so that the predator species would begin to thrive. On the other hand, when the predator numbered more than $106 \approx \exp(4.661)$ one and a half days earlier, the prey species crashed in the intervening period so that the predator species would begin to crash. The increasing phase (lower regime) of the predator population tends to be associated with a robust growth of the prey series that may be less affected by other environmental conditions. On the other hand, during the decreasing phase (upper regime), the predator species would be more susceptible to environmental conditions, as they were already weakened by having less food around. This may explain why the lower regime has a slightly smaller noise variance than the upper regime; hence the slight conditional heteroscedasticity. The difference of the noise variance in the two regimes is unlikely to be significant, although the conditional heteroscedasticity is more apparent in the TAR(2;1,1) model to be discussed below. In general, the regimes defined by the relative position of the lag d values of the response are proxies for some underlying latent process that effects the switching between the linear submodels. With more substantive knowledge of the switching mechanism, the threshold mechanism may, however, be explicitly modeled.

While the interpretation of the regimes above is based on the time series plot, it may be confirmed by examining the fitted submodels. The fitted model of the lower regime implies that on the logarithmic scale

$$Y_t = 0.262 + 1.02Y_{t-1} + 0.234e_t \quad (15.7.5)$$

The lag 1 coefficient is essentially equal to 1 and suggests that the predator species had a (median) growth rate of $(\exp(0.262) - 1)100\% \approx 30\%$ every half day, although the intercept is not significant at the 5% level. This submodel is explosive because $Y_t \rightarrow \infty$ as $t \rightarrow \infty$ if left unchecked.

Interpretation of the fitted model of the upper regime is less straightforward because it is an order 4 model. However, it was suggested earlier that it may be approximated by an AR(1) model. Taking up this suggestion, we reestimated the TAR model with the maximum order being 1 for both regimes.[†] The threshold estimate is unchanged. The lower regime gains one data case, with less of an initial data requirement, but the autoregressive coefficients are almost unchanged. The fitted model of the upper regime becomes

$$Y_t = 0.517 + 0.807 Y_{t-1} + 0.989 e_t \quad (15.7.6)$$

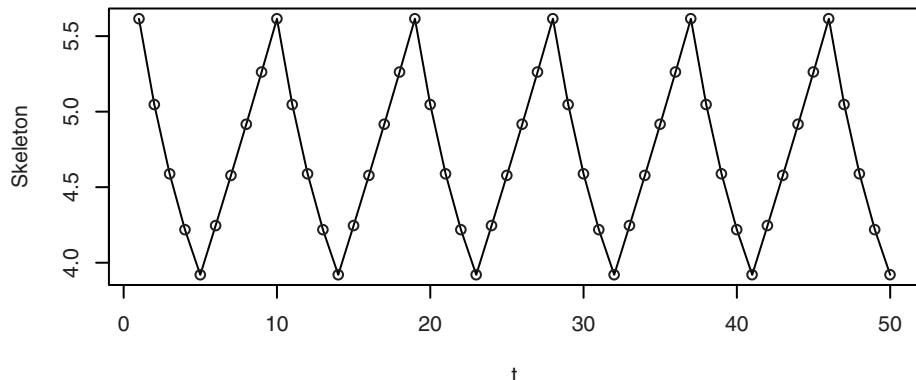
which is a stationary submodel. The growth rate on the logarithmic scale equals

$$Y_t - Y_{t-1} = 0.517 - 0.193 Y_{t-1} + 0.989 e_t \quad (15.7.7)$$

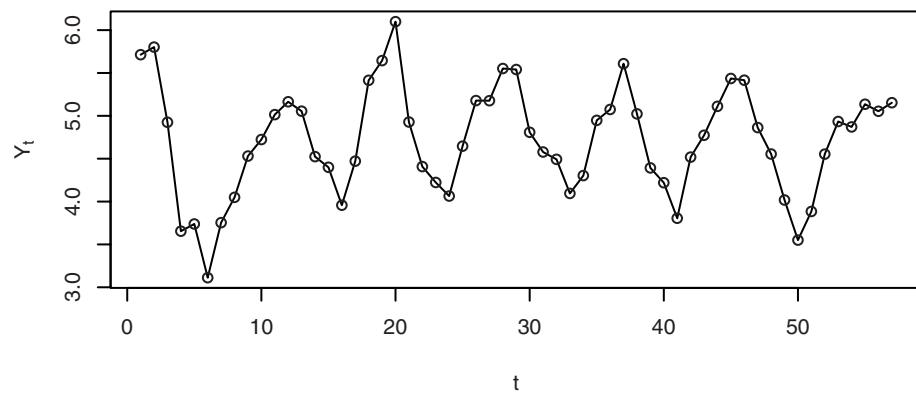
which has a negative median since $Y_{t-1} > 4.661$ on the upper regime. Notice that the conditional heteroscedasticity is more apparent now than the fitted TAR(2;1,4) model. The (nominal) AIC of the TAR(2;1,1) model with $d = 3$ equals 14.78, which is, however, not directly comparable with 10.92 of the TAR(2;1,4) model because of the difference in sample size. Models with different sample sizes may be compared by their nominal AIC per observation. In this case, the normalized AIC increases from $0.206 = 10.92/53$ to $0.274 = 14.78/54$ when the order is decreased from 4 to 1, suggesting that the TAR(2;1,4) model is preferable to the TAR(2;1,1) model.

Another way to assess a nonlinear model is to examine the long-term (asymptotic) behavior of its skeleton. Recall that the skeleton of a model is obtained by suppressing the noise term from the model; that is, replacing the noise term by 0. The skeleton may diverge to infinity, or it may converge to a limit point, a limit cycle, or a strange attractor; see Chan and Tong (2001) for definitions and further discussion. The skeleton of a stationary ARMA model always converges to some limit point. On the other hand, the skeleton of a stationary nonlinear model may display the full complexity of dynamics alluded to earlier. The skeleton of the fitted TAR(2;1,4) model appears to converge to a limit cycle of period 10, as shown in Exhibit 15.13. The limit cycle is symmetric in the sense that its increase phase and decrease phase are of the same length. The apparent long-run stability of the skeleton suggests that the fitted TAR(2;1,4) model with $d = 3$ is stationary. In general, with the noise term in the model, the dynamic behavior of the model may be studied by simulating some series from the stochastic model. Exhibit 15.14 shows a typical realization from the fitted TAR(2;1,4) model.

[†] `predator.tar.2=tar(log(predator.eq), p1=1, p2=1, d=3, a=.1, b=.9, print=T)`

Exhibit 15.13 Skeleton of the TAR(2;1,4) Model for the Predator Series


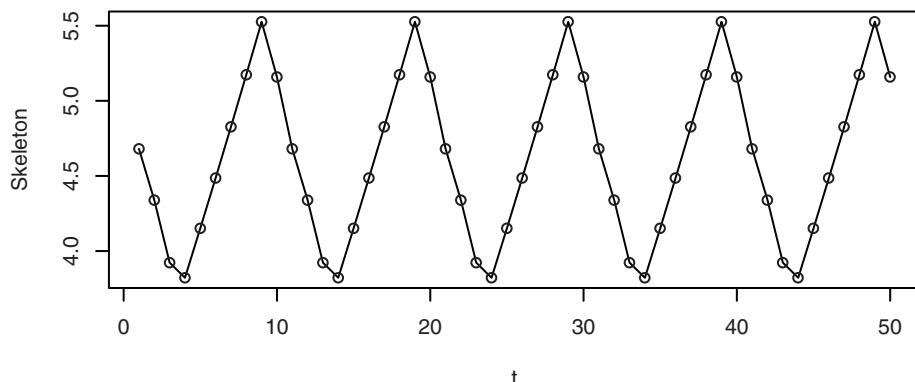
```
> tar.skeleton(predator.tar.1)
```

Exhibit 15.14 Simulated TAR(2;1,4) Series


```
> set.seed(356813)
> plot(y=tar.sim(n=57,object=predator.tar.1)$y,x=1:57,
       ylab=expression(Y[t]),xlab=expression(t),type='o')
```

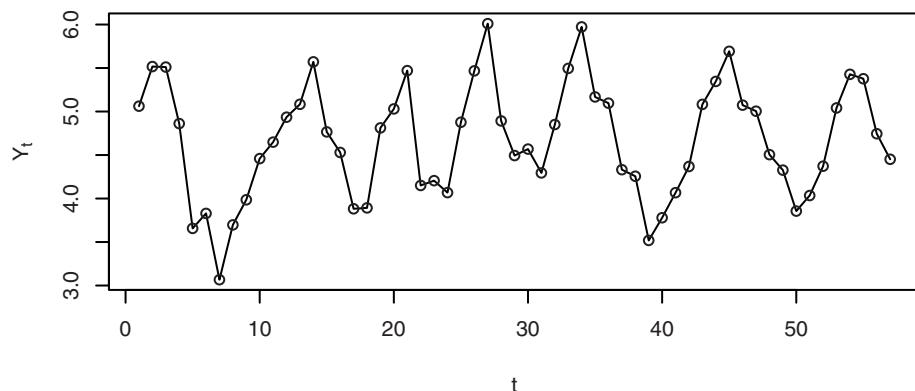
The limit cycle of the skeleton of the fitted TAR(2;1,1) model with $d = 3$ is asymmetric, with the increase phase of length 5 and the decrease phase of length 4; see Exhibit 15.15. A realization of the fitted TAR(2;1,1) model is shown in Exhibit 15.16.

Exhibit 15.15 Skeleton of the First-Order TAR Model for the Predator Series



```
> predator.tar.2=tar(log(predator.eq), p1=1, p2=1, d=3, a=.1, b=.9,
+ print=T)
> tar.skeleton(predator.tar.2)
```

Exhibit 15.16 Simulation of the Fitted TAR(2;1,1) Model



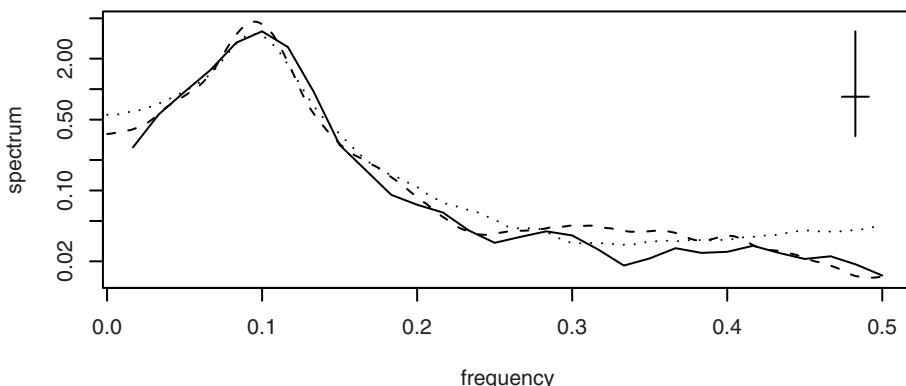
```
> set.seed(356813)
> plot(y=tar.sim(n=57, object=predator.tar.2)$y, x=1:57,
+ ylab=expression(Y[t]), xlab=expression(t), type='o')
```

For the predator data, excluding the two initial transient cycles and the last incomplete cycle, the table in Exhibit 15.17 lists the length of the successive increasing and decreasing phases. Observe that the mean length of the increasing phases is 5.4 and that of the decreasing phases is 4.6.

Exhibit 15.17 Length of the Increasing and Decreasing Phases of the Predator Series

Phase	
Increasing	Decreasing
6	4
7	5
5	4
4	5
5	5

There is some evidence of asymmetry with a longer increase phase than the decrease phase. Based on the cycle length analysis, the TAR(2;1,1) model appears to pick up the asymmetric cycle property better than the TAR(2;1,4) model, but the latter model gets the cycle length better matched to the observed average cycle length. A more rigorous comparison between the cyclical behavior of a fitted model and that of the data can be done by comparing the spectral density of the data with that of a long realization from the fitted model. Exhibit 15.18 plots the spectrum of the data using a modified Daniell window with a (3,3) span. Also plotted is the spectrum of the fitted TAR(2;1,4) model (dashed line) and that of the fitted TAR(2;1,1) model (dotted line), both of which are based on a simulated realization of size 10,000, a modified Daniell window with a (200,200) span, and 10% tapering. It can be seen that the spectrum of the TAR(2;1,4) model follows that of the predator series quite closely and is slightly better than the simplified TAR(2;1,1) model.

Exhibit 15.18 Spectra of Log(predator) Series, Dashed Line for TAR(2;1,1), Dotted Line for TAR(2;1,4)


```
> set.seed(2357125)
> yy.1.4=tar.sim(predator.tar.1,n=10000)$y
> yy.1=tar.sim(predator.tar.2,n=10000)$y
> spec.1.4=spec(yy.1.4,taper=.1, span=c(200,200),plot=F)
```

```
> spec.1=spec(yy.1,taper=.1, span=c(200,200),plot=F)
> spec.predator=spec(log(predator.eq),taper=.1,
+ span=c(3,3),plot=F)
> spec.predator=spec(log(predator.eq),taper=.1,span=c(3,3),
+ ylim=range(c(spec.1.4$spec,spec.1$spec,spec.predator$spec)))
> lines(y=spec.1.4$spec,x=spec.1.4$freq,lty=2)
> lines(y=spec.1$spec,x=spec.1$freq,lty=3)
```

We note that the conditional least squares method with the predator data yields the same threshold estimate for $d = 3$ and hence also the other parameter estimates, although this need not always be the case. Finally, a couple of clarifying remarks on the predator series analysis are in order. As the experimental prey series is also available, a bivariate time series analysis may be studied. But it is not pursued here since nonlinear time series analysis with multiple time series is not a well-charted area. Moreover, real biological data are often observational, and abundance data of the prey population are often much noisier than those of the predator population because the predator population tends to be fewer in number than the prey population. Furthermore, predators may switch from their favorite prey food to other available prey species when the former becomes scarce, rendering a more complex prey-predator system. For example, in a good year, hares may be seen hopping around in every corner in the neighborhood, whereas it is rare to spot a lynx, their predator! Thus, biological analysis often focuses on the abundance data of the predator population. Nonetheless, univariate time series analysis of the abundance of the predator species may shed valuable biological insights on the prey-predator interaction; see Stenseth et al. (1998, 1999) for some relevant discussion on a panel of Canadian lynx series. For the lynx data, a TAR(2;2,2) model with delay equal to 2 is the prototypical model, with delay 2 lending some nice biological interpretations. We note that, for the predator series, delay 2 is very competitive; see Exhibit 15.11, and hence may be preferred on biological grounds. In one exercise, we ask the reader to fit a TAR model for the predator series with delay set to 2 and interpret the findings by making use of the framework studied in Stenseth et al. (1998, 1999).

15.8 Model Diagnostics

In Section 15.7, we introduced some model diagnostic techniques; for example, skeleton analysis and simulation. Here, we discuss some formal statistical approaches to model diagnostics via residual analysis. The raw residuals are defined as subtracting the fitted value from the data, where the t th fitted value is the estimated conditional mean of Y_t given past values of Y 's; that is, the residuals $\hat{\varepsilon}_t$ are given by

$$\begin{aligned}\hat{\varepsilon}_t = & Y_t - \{\hat{\phi}_{1,0} + \hat{\phi}_{1,1}Y_{t-1} + \cdots + \hat{\phi}_{1,p}Y_{t-p}\}I(Y_{t-\hat{d}} \leq \hat{r}) \\ & - \{\hat{\phi}_{2,0} + \hat{\phi}_{2,1}Y_{t-1} + \cdots + \hat{\phi}_{2,p}Y_{t-p}\}I(Y_{t-\hat{d}} > \hat{r})\end{aligned}\tag{15.8.1}$$

These are the same as the raw residuals from the fitted submodels. The standardized residuals are obtained by normalizing the raw residuals by their appropriate standard deviations:

$$\hat{e}_t = \frac{\hat{\varepsilon}_t}{\hat{\sigma}_1 I(Y_{t-\hat{d}} \leq \hat{r}) + \hat{\sigma}_2 I(Y_{t-\hat{d}} > \hat{r})} \quad (15.8.2)$$

that is, raw residuals from the lower (upper) regime are normalized by the noise standard deviation estimate of the lower (upper) regime. As in the linear case, the time series plot of the standardized residuals should look random, as they should be approximately independent and identically distributed if the TAR model is the true data mechanism; that is, if the TAR model is correctly specified. As before, we look for the presence of outliers and any systematic pattern in such a plot, in which case it may provide a clue for specifying a more appropriate model. The independence assumption of the standardized errors can be checked by examining the sample ACF of the standardized residuals. Non-constant variance may be checked by examining the sample ACF of the squared standardized residuals or that of the absolute standardized residuals.

Here, we consider the generalization of the portmanteau test based on some overall measure of the magnitude of the residual autocorrelations. The reader may want to review the discussion in Section 12.5 on page 301, where we explain that even if the model is correctly specified, the residuals are generally dependent and so are their sample autocorrelations. Unlike the case of linear ARIMA models, the dependence of the residuals necessitates the employment of a (complex) quadratic form of the residual autocorrelations:

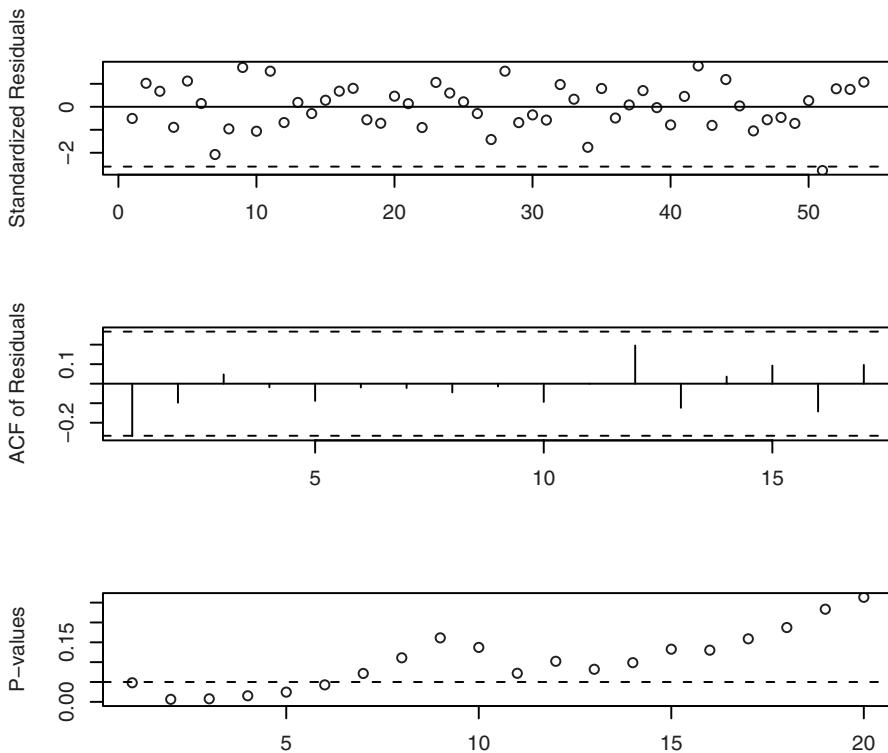
$$B_m = n_{\text{eff}} \sum_{i=1}^m \sum_{j=1}^m q_{i,j} \hat{\rho}_i \hat{\rho}_j \quad (15.8.3)$$

where $n_{\text{eff}} = n - \max(p_1, p_2, d)$ is the effective sample size, $\hat{\rho}_i$ the i th-lag sample autocorrelation of the standardized residuals, and $q_{i,j}$ some model-dependent constants given in Appendix L on page 421. If the true model is a TAR model, $\hat{\rho}_i$ are likely close to zero and so is B_m , but B_m tends to be large if the model specification is incorrect. The quadratic form is designed so that B_m is approximately distributed as χ^2 with m degrees of freedom. Mathematical theory predicts that the χ^2 distribution approximation is generally more accurate with larger sample size and relatively small m as compared with the sample size.

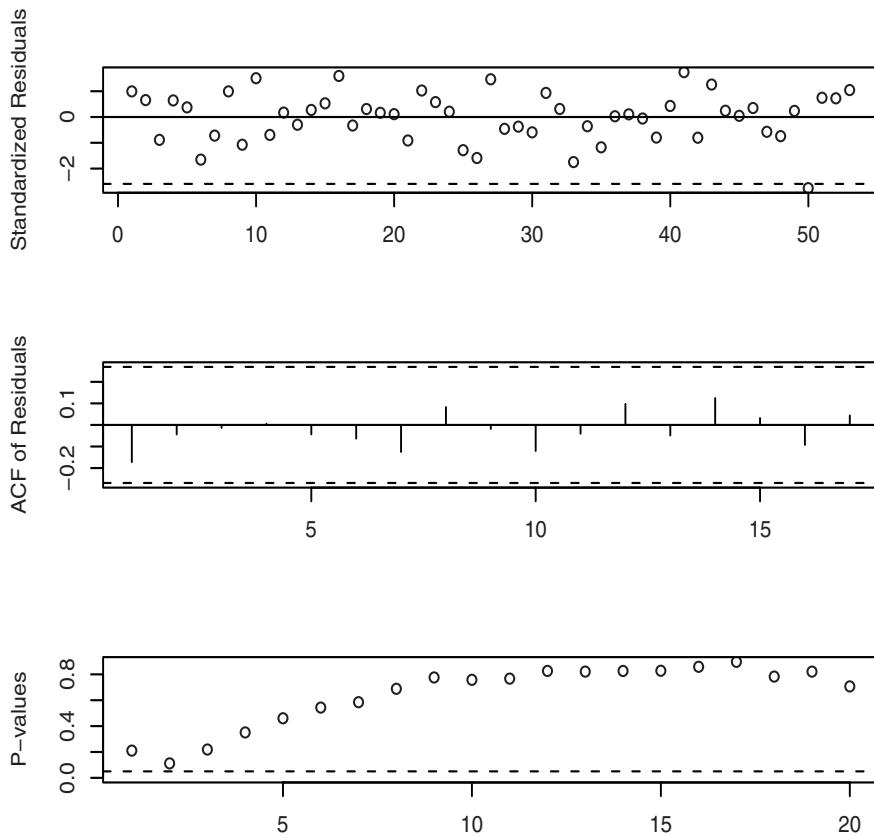
In practice, the p -value of B_m may be plotted against m over a range of m values to provide a more comprehensive assessment of the independence assumption on the standardized errors. The bottom figure of Exhibit 15.19 reports the portmanteau test of the TAR(2;1,1) model fitted to the predator series discussed earlier for $1 \leq m \leq 12$. The top figure there is the time series plot of the standardized residuals. Except for a possible outlier, the plot shows no particular pattern. The middle figure is the ACF plot of the standardized residuals. The confidence band is based on the simple $1.96/\sqrt{n}$ rule and should be regarded as a rough guide on the significance of the residual ACF. It suggests that the lag 1 residual autocorrelation is significant. The more rigorous portmanteau tests are all significant for $m \leq 6$, suggesting a lack of fit for the TAR(2;1,1) model. Similar diagnostics for the TAR(2;1,4) model are shown in Exhibit 15.20. Now, the only potential problem is a possible outlier. However, the fitted model changed little upon deleting the last four data points, including the potential outlier; hence we conclude that

the fitted TAR(2;1,4) model is fairly robust. Exhibit 15.21 displays the QQ normal score plot of the standardized residuals, which is apparently straight and hence the errors appear to be normally distributed. In summary, the fitted TAR(2;1,4) model provides a good fit to the predator series.

Exhibit 15.19 Model Diagnostics of the First-Order TAR Model: Predator Series

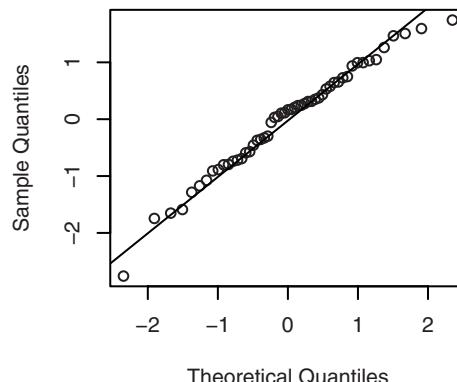


```
> win.graph(width=4.875,height=4.5)
> tsdiag(predator.tar.2,gof.lag=20)
```

Exhibit 15.20 Model Diagnostics for the TAR(2;1,4) Model: Predator Series

```
> tsdiag(predator.tar.1,gof.lag=20)
```

Exhibit 15.21 QQ Normal Plot of the Standardized Residuals



```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(predator.tar.1$std.res); qqline(predator.tar.1$std.res)
```

15.9 Prediction

In this section, we consider the problem of predicting future values from a TAR process. In practice, prediction is based on an estimated TAR model. But, as in the case of ARIMA models, the uncertainty due to parameter estimation is generally small compared with the natural variation of the underlying process. So, we shall proceed below as if the fitted model were the true model. The uncertainty of a future value, say $Y_{t+\ell}$, is completely characterized by its conditional probability distribution given the current and past data Y_t, Y_{t-1}, \dots , referred to as the ℓ -step-ahead predictive distribution below. For ARIMA models with normal errors, all predictive distributions are normal, which greatly simplifies the computation of a predictive interval, as it suffices to find the mean and variance of the predictive distribution. However, for nonlinear models, the predictive distributions are generally nonnormal and often intractable. Hence, a prediction interval may have to be computed by brute force via simulation. The simulation approach may be best explained in the context of a first-order nonlinear autoregressive model:

$$Y_{t+1} = h(Y_t, e_{t+1}) \quad (15.9.1)$$

Given $Y_t = y_t, Y_{t-1} = y_{t-1}, \dots$, we have $Y_{t+1} = h(y_t, e_{t+1})$ so a realization of Y_{t+1} from the one-step-ahead predictive distribution can be obtained by drawing e_{t+1} from the error distribution and computing $h(y_t, e_{t+1})$. Repeating this procedure independently B times, say 1000 times, we get a random sample of B values from the one-step-ahead predictive distribution. The one-step-ahead predictive mean may be estimated by the sample mean of these B values. However, it is important to inspect the shape of the one-step-ahead predictive distribution in order to decide how best to summarize the predictive information. For example, if the predictive distribution is multimodal or very

skewed, the one-step-ahead predictive mean need not be an appropriate point predictor. A generally useful approach is to construct a 95% prediction interval for Y_{t+1} ; for example, the interval defined by the 2.5th percentile to the 97.5th percentile of the simulated B values.

The simulation approach can be readily extended to finding the ℓ -step-ahead predictive distribution for any integer $\ell \geq 2$ by iterating the nonlinear autoregression.

$$\left. \begin{array}{l} Y_{t+1} = h(Y_t, e_{t+1}) \\ Y_{t+2} = h(Y_{t+1}, e_{t+2}) \\ \vdots \\ Y_{t+\ell} = h(Y_{t+\ell-1}, e_{t+\ell}), \end{array} \right\} \quad (15.9.2)$$

where $Y_t = y_t$ and $\{e_{t+1}, \dots, e_{t+\ell}\}$ is a random sample of ℓ values drawn from the error distribution. This procedure may be repeated B times to yield a random sample from the ℓ -step-ahead predictive distribution, with which we can compute prediction intervals of $Y_{t+\ell}$ or any other predictive summary statistic.

Indeed, the ℓ -tuple $(Y_{t+1}, \dots, Y_{t+\ell})$ is a realization from the joint predictive distribution of the first ℓ -step-ahead predictions. So, the procedure above actually yields a random sample of B vectors from the joint predictive distribution of the first ℓ -step-ahead predictions.

Henceforth in this section, we focus on the prediction problem when the true model is a TAR model. Fortunately, the simulation approach is not needed for computing the one-step-ahead predictive distribution in the case of a TAR model. To see this, consider the simple case of a first-order TAR model. In this case, Y_{t+1-d} is known, so that the regime for Y_{t+1} is known. If $Y_{t+1-d} \leq r$, then Y_{t+1} follows the AR(1) model

$$Y_{t+1} = \phi_{1,0} + \phi_{1,1} Y_t + \sigma_1 e_{t+1} \quad (15.9.3)$$

Because $Y_t = y_t$ is fixed, the conditional distribution of Y_{t+1} is normal with mean equal to $\phi_{1,0} + \phi_{1,1}y_t$ and variance σ_1^2 . Similarly, if $Y_t > r$, Y_{t+1} follows the AR(1) model of the upper regime so that, conditionally, it is normal with mean $\phi_{2,0} + \phi_{2,1}y_t$ and variance σ_2^2 . A similar argument shows that, for any TAR model, the one-step-ahead predictive distribution is normal. The predictive mean is, however, a piecewise linear function, and the predictive standard deviation is piecewise constant.

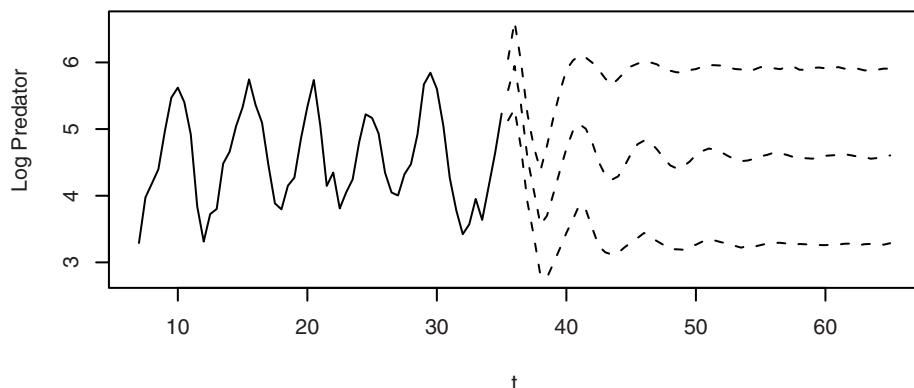
Similarly, it can be shown that if $\ell \leq d$, then the ℓ -step-ahead predictive distribution of a TAR model is also normal. But if $\ell > d$, the ℓ -step-ahead predictive distribution is no longer normal. The problem can be illustrated in the simple case of a first-order TAR model with $d = 1$ and $\ell = 2$. While Y_{t+1} follows a *fixed* linear model determined by the observed value of Y_t , Y_{t+2} may be in the lower or upper regime, depending on the random value of Y_{t+1} . Suppose that $y_t \leq r$. Now, Y_{t+1} falls in the lower regime if $Y_{t+1} = \sigma_1 e_{t+1} + \phi_{1,0} + \phi_{1,1}y_t \leq r$, which happens with probability $p_t = Pr(\sigma_1 e_{t+1} + \phi_{1,0} + \phi_{1,1}y_t \leq r)$ and in which case

$$\begin{aligned}
 Y_{t+2} &= \sigma_1 e_{t+2} + \phi_{1,0} + \phi_{1,1} Y_{t+1} \\
 &= \sigma_1 e_{t+2} + \phi_{1,1} \sigma_1 e_{t+1} + \phi_{1,1} \phi_{1,0} + \phi_{1,1}^2 y_t + \phi_{1,0}
 \end{aligned} \tag{15.9.4}$$

which is a normal distribution with mean equal to $\phi_{1,1} \phi_{1,0} + \phi_{1,1}^2 y_t + \phi_{1,0}$ and variance $\sigma_1^2 + \phi_{1,1}^2 \sigma_1^2$. On the other hand, with probability $1 - p_t$, Y_{t+1} falls in the upper regime, in which case the conditional distribution of Y_{t+2} is normal but with mean $\phi_{2,1}(\phi_{1,0} + \phi_{1,1} y_t) + \phi_{2,0}$ and variance $\sigma_2^2 + \phi_{2,1}^2 \sigma_1^2$. Therefore, the conditional distribution of Y_{t+2} is a mixture of two normal distributions. Note that the mixture probability p_t depends on y_t . In particular, the higher-step-ahead predictive distributions are nonnormal for a TAR model if $\ell > d$, and so we have to resort to simulation to find the predictive distributions.

As an example, we compute the prediction intervals for the logarithmically transformed predator data based on the fitted TAR(2;1,4) model with $d = 3$; see Exhibit 15.22, where the middle dashed line is the median of the predictive distribution and the other dashed lines are the 2.5th and 97.5th percentiles of the predictive distribution.

Exhibit 15.22 Prediction of the Predator Series



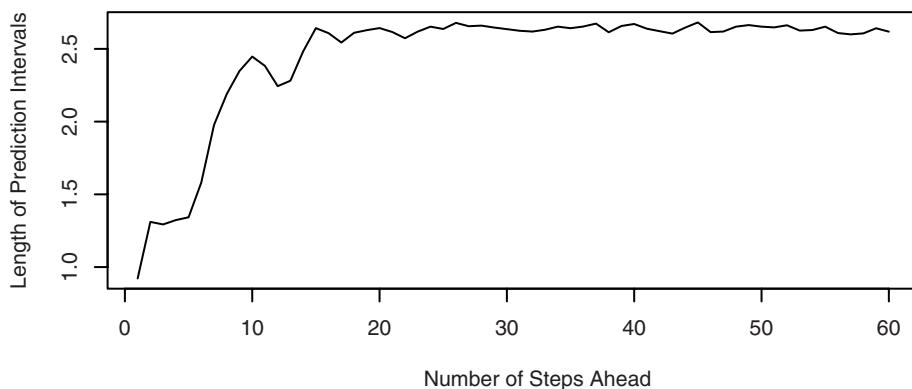
```

> set.seed(2357125)
> win.graph(width=4.875, height=2.5, pointsize=8)
> pred.predator=predict(predator.tar.1, n.ahead=60, n.sim=10000)
> yy=ts(c(log(predator.eq), pred.predator$fit), frequency=2,
  start=start(predator.eq))
> plot(yy, type='n', ylim=range(c(yy, pred.predator$pred.interval))),
  ylab='Log Predator', xlab=expression(t))
> lines(log(predator.eq))
> lines(window(yy, start=end(predator.eq)+c(0,1)), lty=2)
> lines(ts(pred.predator$pred.interval[2,],
  start=end(predator.eq)+c(0,1), freq=2), lty=2)
> lines(ts(pred.predator$pred.interval[1,],
  start=end(predator.eq)+c(0,1), freq=2), lty=2)

```

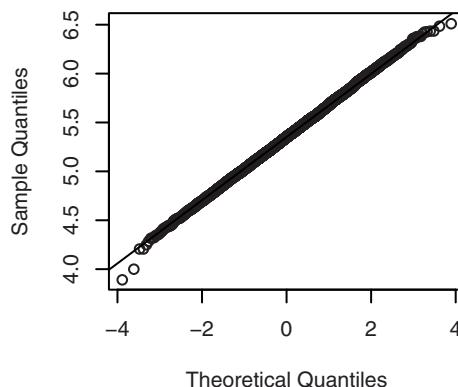
The simulation size here is 10,000. In practice, a smaller size such as 1000 may be adequate. The median of the predictive distribution can serve as a point predictor. Notice that the predictive medians display the cyclical pattern of the predator data initially and then approach the long-run median with increasing number of steps ahead. Similarly, the predictive intervals approach the interval defined by the 2.5th and 97.5th percentiles of the stationary distribution of the fitted TAR model. However, a new feature is that prediction need not be less certain with increasing number of steps ahead, as the length of the prediction intervals does not increase monotonically with increasing number of steps ahead; see Exhibit 15.23. This is radically different from the case of ARIMA models, for which the prediction variance always increases with the number of prediction steps ahead.

Exhibit 15.23 Width of the 95% Prediction Intervals Against Lead Time

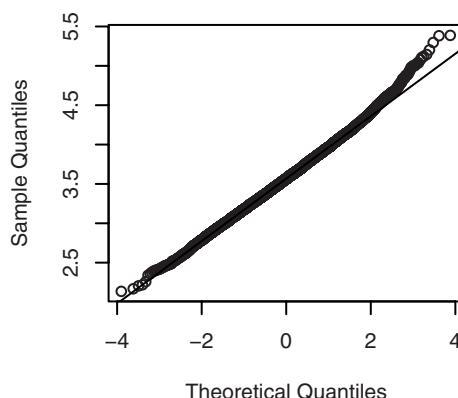


```
> plot(ts(apply(pred.predator$pred.interval, 2,
  function(x){x[2]-x[1]})),
  ylab='Length of Prediction Intervals',
  xlab='Number of Steps Ahead')
```

Recall that, for the TAR model, the prediction distribution is normal if and only if the number of steps ahead $\ell \leq d$. Exhibit 15.24 shows the QQ normal score plot of the three-step-ahead predictive distribution, which is fairly straight. On the other hand, the QQ normal score plot of the six-step-ahead predictive distribution (Exhibit 15.25) is consistent with nonnormality.

Exhibit 15.24 QQ Normal Plot of the Three-Step-Ahead Predictive Distribution

```
> win.graph(width=2.5,height=2.5,pointsize=8)
> qqnorm(pred.predator$pred.matrix[,3])
> qqline(pred.predator$pred.matrix[,3])
```

Exhibit 15.25 QQ Normal Plot of the Six-Step-Ahead Predictive Distribution

```
> qqnorm(pred.predator$pred.matrix[,6])
> qqline(pred.predator$pred.matrix[,6])
```

15.10 Summary

In this chapter, we have introduced an important nonlinear times serie model—the threshold model. We have shown how to test for nonlinearity and, in particular, for threshold nonlinearity. We then proceeded to consider the estimation of the unknown parameters in these models using both the minimum AIC (MAIC) criterion and the conditional least squares approach. As with all models, we learned how to criticize them through various model diagnostics, including an extended portmanteau test. Finally, we demonstrated how to form predictions from threshold models, including the calculation and display of prediction intervals. Several substantial examples were used to illustrate the methods and techniques discussed.

EXERCISES

- 15.1** Fit a TAR model for the predator series with delay set to 2, and interpret the findings by making use of the framework studied in Stenseth et al. (1998, 1999). (You may first want to check whether or not their framework is approximately valid for the TAR model.) Also, compare the fitted model with the TAR(2;1,4) model with delay 3 reported in the text. (The data file is named *veilleux*.)
- 15.2** Fit a TAR model to the square-root-transformed relative sunspot data, and examine its goodness of fit. Interpret the fitted TAR model. (The data file is named *spots*.)
- 15.3** Predict the annual relative sunspot numbers for ten years using the fitted model obtained in Exercise 15.2. Draw the prediction intervals and the predicted medians. (The data file is named *spots*.)
- 15.4** Examine the long-run behavior of the skeleton of the fitted model for the relative sunspot data. Is the fitted model likely to be stationary? Explain your answer.
- 15.5** Simulate a series of size 1000 from the TAR model fitted to the relative sunspot data. Compute the spectrum of the simulated realization and compare it with the spectrum of the data. Does the fitted model capture the correlation structure of the data?
- 15.6** Draw the lagged regression plots for the square-root-transformed hare data. Is there any evidence that the hare data are nonlinear? (The data file is named *hare*.)
- 15.7** Carry out formal tests (Keenan's test, Tsay's test, and threshold likelihood ratio test) for nonlinearity for the hare data. Is the hare abundance process nonlinear? Explain your answer. (The data file is named *hare*.)
- 15.8** Assuming that the hare data are nonlinear, fit a TAR model to the hare data and examine the goodness of fit. (The data file is named *hare*.)

- 15.9** This exercise assumes that the reader is familiar with Markov chain theory. Consider a simple TAR model that is piecewise constant:

$$Y_t = \begin{cases} \phi_{1,0} + \sigma_1 e_r & \text{if } Y_{t-1} \leq r \\ \phi_{2,0} + \sigma_2 e_r & \text{if } Y_{t-1} > r \end{cases}$$

where $\{e_t\}$ are independent standard normal random variables. Let $R_t = 1$ if $Y_t \leq r$ and 2 otherwise, which is a Markov chain.

- (a) Find the transition probability matrix of R_t and its stationary distribution.
- (b) Derive the stationary distribution of $\{Y_t\}$.
- (c) Find the lag 1 autocovariance of the TAR process.

Appendix L: The Generalized Portmanteau Test for TAR

The basis of the portmanteau test is the result that, if the TAR model is correctly specified, $\hat{\rho}_1, \hat{\rho}_2, \dots, \hat{\rho}_m$ are approximately jointly normally distributed with zero mean and covariances $Cov(\hat{\rho}_i, \hat{\rho}_j) = q_{ij}$, where Q is an $m \times m$ matrix whose (i,j) element equals q_{ij} and whose formula is given below; See Chan (2008) for a proof of this result. It can be shown that $Q = I - UV^{-1}U^T$ where I is an $m \times m$ identity matrix,

$$U = E \left\{ \begin{bmatrix} e_{t-1} \\ e_{t-2} \\ \vdots \\ e_{t-m} \end{bmatrix} \left[I_r, Y_{t-1}I_r, \dots, Y_{t-p}I_r(1-I_t), Y_{t-1}(1-I_t), \dots, Y_{t-p_2}(1-I_t) \right] \right\}$$

where $I_t = I(Y_{t-d} \leq r)$, the expectation of a matrix is taken elementwise, and

$$V = E \left\{ \begin{bmatrix} I_t \\ Y_{t-1}I_t \\ \vdots \\ Y_{t-p_1}I_t \\ (1-I_t) \\ Y_{t-1}(1-I_t) \\ \vdots \\ Y_{t-p_2}(1-I_t) \end{bmatrix} \left[I_r, Y_{t-1}I_r, \dots, Y_{t-p}I_r(1-I_t), Y_{t-1}(1-I_t), \dots, Y_{t-p_2}(1-I_t) \right] \right\}$$

These expectations can be approximated by sample averages computed with the true errors replaced by the standardized residuals and the unknown parameters by their estimates. For example, $E\{e_{t-1}I(Y_{t-d} \leq r)\}$ can be approximated by

$$\frac{1}{n} \sum_{t=1}^n \hat{e}_{t-1} I(Y_{t-d} \leq \hat{r})$$

where the initial standardized residuals $\hat{e}_t = 0$ for $t \leq \max(p_1, p_2, d)$.

APPENDIX: AN INTRODUCTION TO R

Introduction

All of the plots and numerical output displayed in this book were produced with the R software, which is available at no cost from the R Project for Statistical Computing. The software is available under the terms of the Free Software Foundation's GNU General Public License in source code form. It runs on a wide variety of operating systems, including Windows, Mac OS, UNIX, and similar systems, including FreeBSD and Linux. R is a language and environment for statistical computing and graphics, provides a wide variety of statistical methods (time series analysis, linear and nonlinear modeling, classical statistical tests, and so forth) and graphical techniques, and is highly extensible. In particular, one of the authors (KSC) has produced a large number of new or enhanced R functions specifically tailored to the methods described in this book. They are available for download in an R package named TSA on the R Project Website at www.r-project.org. The TSA functions are listed on page 468.

Important references for learning much more about R are also available at the R-Project Website, including *An Introduction to R: Notes on R, a Programming Environment for Data Analysis and Graphics*, Version 2.4.1 (2006-12-18), by W. N. Venables, D. M. Smith, and the R Development Core Team, (2006), and *R: A Language and Environment for Statistical Computing Reference Index*, Version 2.4.1 (2006-12-18), by The R Development Core Team (2006a).

The R software is the GNU implementation of the famed S language. It has been under active development by the R team, with contributions from many statisticians all over the world. R has become a versatile and powerful platform for doing statistical analysis. We shall confine our discussion to the Windows version of R. To obtain the software, visit the Website at www.r-project.org. Click on CRAN on the left-side of the screen under Download. Scroll down the list of CRAN Mirror sites and click on one of them nearest to you geographically. Click on the link for Windows (or Linux or MacOS X as appropriate) and click on the link named **base**. Finally, click on the link labeled R-2.6.1-win32.exe. (This file indicates release 2.6.1, the latest available release as of this writing. Newer versions come out frequently.) Save the file somewhere convenient, for example, on your desktop. When the download finishes, double-click the program icon and proceed with installing the software. (The discussion that follows assumes that you accept all of the defaults during installation.) At the end of this appendix, on page 468, you will find a listing and brief description of all the new or enhanced functions that are contained in the TSA package.

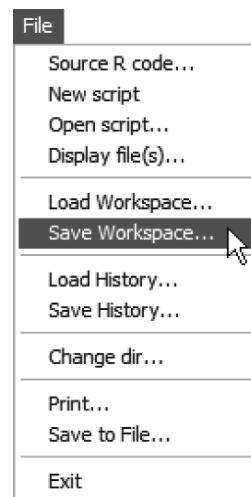
Before you start the R software for the first time, you should create a folder or directory, say **Rwork**, to hold data files that you will use with R for this project or course. This will be the working directory whenever you use R for this particular project or course. This directory is to contain the **workspace**, a file that contains all the objects (variables and functions) created in an R session. You should create separate

working directories for different projects or different courses.[†] After R is successfully installed on your computer, there will be an R shortcut icon on your desktop. If you have created your working directory, start R by clicking the R icon (shown at the right). When the software has loaded, you will have a console window similar to the one shown in Exhibit 1 with a bottom line that reads > followed by a large rectangular cursor (probably in red). This is the R prompt. You may enter commands at this prompt, and they will be carried out when you press the Enter key. Several tasks are available through the menus.

The first task is to save your workspace in the working directory you created. To do so, select the File menu and then click on the choice Save workspace... .[‡] You now may either browse to the directory Rwork that you created (which may take many steps) or type in the full path name; for example “C:\Documents and Settings\JoeStudent\My Documents\Course156\Rwork”. If your working directory is on a USB flash drive designated as drive E, you might simply enter “E:Rwork”. Click OK, and from this point on in this session, R will use the folder Rwork as its working directory.

You exit R by selecting Exit on the File menu. Every time you exit R, you will receive a message as to whether or not to Save the workspace image. Click Yes to save the workspace, and it will be saved in your current working directory. The next time you want to resume work on that same project, simply navigate to that working directory and locate the R icon there attached to the file named .RData. If you double-click this icon, R will start with this directory already selected as the working directory and you can get right to work on that project. Furthermore, you will receive the message [Previously saved workspace restored].

Exhibit 1 shows a possible screen display after you have started R, produced two different graphs, and worked with R commands in a script window using the R editor. Numerical results in R are displayed in the console window. Commands may be entered (keyed) in either the console window and executed immediately or (better) in a script window (the R editor) and then submitted to be run in R. The Menu bar and buttons will change depending on which window is currently the “focus.”

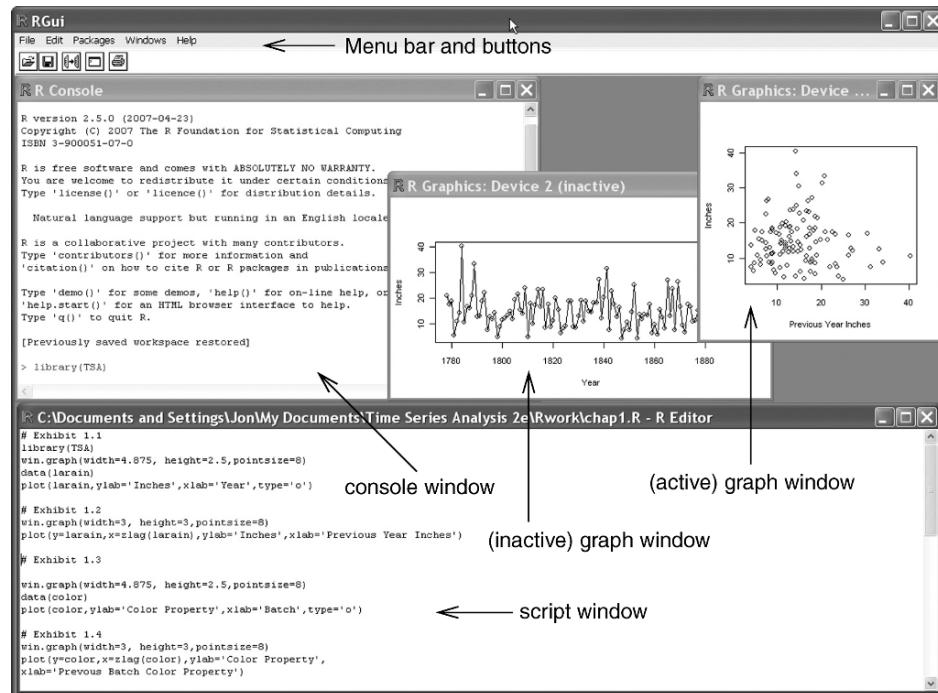


[†] If you work in a shared computer lab, check with the lab supervisor for information about starting R and about where you may save your work.

[‡] If you neglected to create a working directory *before* starting R, you may do so at this point. Navigate to a suitable place, click the Create new folder button, and create the folder Rwork now.

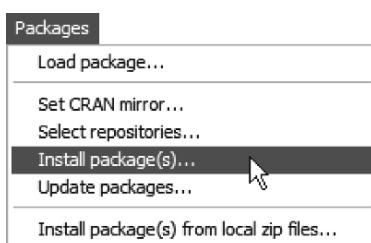
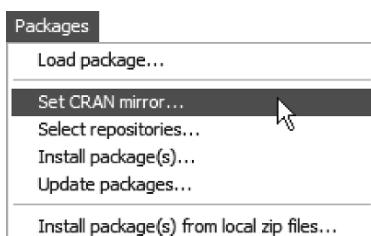


Exhibit 1 Windows Graphical User Interface for the R Software



A particularly useful feature of R is its ease of including supplementary tools in the form of libraries or packages. For example, all the datasets and the new or enhanced R functions used in this book are collected into a package called TSA that can be downloaded and installed in R. This can be done by clicking the Packages menu and then selecting Set CRAN mirror. Again select a mirror site that is closest to you geographically, and a window containing the names of all available packages will pop up.

In addition to our TSA package, you will need to install packages named leaps, locfit, MASS, mgcv, tseries, and uroot. Click the Packages menu once more, click Install package(s), and scroll through the window. Hold down the Ctrl key and click on each of these seven package names. When you have all seven selected, click OK, and they will be installed on your system by R. You only have to install them



once (but, of course, they may be updated in the future and some of them may be incorporated into the core of R and not need to be installed separately).

We will go over commands selected from the various chapters as a tutorial for R, but before delving into those, we first present an overview of R. R is an object-oriented language. The two main objects in R are data and functions. R admits many data structures. The simplest data structure is a vector that contains raw data. To create a data vector named `Dat` containing, say, 31, 4, 15, and 93, after the `>` prompt in the console window, enter the following command

```
Dat=c(31,4,15,93)
```

and then press the Enter key. The equal sign symbol signifies assigning the object on its right-hand side to the object on its left-hand side. The expression `c(31,4,15,93)` stands for concatenating the numbers within the parentheses to make a vector. So, the command creates an object named `Dat` that is a vector containing the numbers 31, 4, 15, and 93. R is case-sensitive, so the objects named `Dat` and `DAt` are different. To reveal the contents of an object, simply type the name of the object and press the Enter key. So, typing `Dat` in the R console window (and pressing the Enter key) will display the contents of `Dat`. If you subsequently enter `DAt` at the R prompt, it will complain by returning an error message saying that object "DAt" is not found. The name of an object is a string of characters that may contain letters, numerals, and the period sign, but the leading character is required to be a letter.[†] For example, `Abc123.a` is a valid name for an R object but `12a` is not. R has some useful built-in objects, for example `pi`, which contains the numerical value of π required for trigonometric operations such as computing the area of a circle.

For us, the most useful data structure is a time series. A time series is a vector with additional information on the epoch of the first datum and the number of data per a basic unit of time interval. For example, suppose we have quarterly data starting from the second quarter of 2006: 12, 31, 22, 24, 30. This time series can be created as follows:

```
> Dat2=ts(c(12,31,22,24,30), start=c(2006,2), frequency=4)
```

Its content can be verified by the command

```
> Dat2
      Qtr1  Qtr2  Qtr3  Qtr4
2006       12     31     22
2007       24     30
```

Larger datasets already in a data file (raw data separated by spaces, tabs, or line breaks) can be loaded into R by the command

```
> Dat2=ts(scan('file1'), start=c(2006,2), frequency=4)
```

where it is assumed that the data are contained in the file named `file1` in the same directory where you start up R (or the one changed into via the `change dir` command). Notice that the file name, `file1`, is surrounded by single quotes (''). In R, all

[†] Certain names should be avoided, as they have special meanings in R. For example, the letter T is short for true, F for false, and C for concatenate or combine.

character variables must be so enclosed. You may, however, use either single quotes or double quotes ("") as long as you use them in pairs.

Datasets with several variables may be read into R by the `read.table` function. The data must be stored in a table form: The first row contains the variable names, and starting from the second line, the data are stored so that data from each case make up a row in the order of the variable names. The relevant command is

```
Dat3=read.table('file2',header=T)
```

where `file2` is the name of the file containing the data. The argument `header=T` specifies that the variable names are in the first line of the file. For example, let the contents of a file named `file2` in your working directory be as follows:

```
Y X  
1 2  
3 7  
4 8  
5 9  
> Dat3=read.table('file2',header=T)  
> Dat3  
Y X  
1 1 2  
2 3 7  
3 4 8  
4 5 9
```

Note that in displaying `Dat3`, R adds the row labels, defaulted to be from 1 to the number of data cases. The output of `read.table` is a `data.frame`, which is a data structure for a table of data. More discussion on `data.frame` can be found below. Presently, it suffices to remember that the variables inside a `data.frame` are not accessible. Think of `Dat3` as a closed suitcase. It has to be opened before its variables are accessible in an R session. The command to “open” a `data.frame` is to attach it:

```
> Y  
Error: object "Y" not found  
> attach(Dat3)  
> Y  
[1] 1 3 4 5  
> X  
[1] 2 7 8 9
```

R can also read in data from an Excel file saved in the `csv` (*comma-separated values*) format, with the first row containing the variable names. Suppose `file2.csv` contains a spreadsheet containing the same information as in `file2`. The commands for reading in the data from `file2.csv` are similar to the one for a text file.

```
> Dat4=read.csv('file2.csv',header=T)  
> Dat4  
Y X  
1 1 2  
2 3 7  
3 4 8  
4 5 9
```

The functions `scan`, `read.table`, and `read.csv` have many other useful options. Use R Help to learn more about them. For example, run the command `?read.table`, and a window showing detailed information for the `read.table` command will open. Remember that prefacing the question mark to any function name will display the function's details in a new Help window.

Functions in R are similar to functions in the programming language C. A function is invoked by typing its name followed by a list of arguments enclosed by parentheses. For example, the concatenate function has the name “`c`” and its purpose is to create a vector obtained by concatenating the arguments supplied to the function.

```
> c(12, 31, 22, 24, 30)
```

Note that there can be no space between the left parenthesis and the function name. Even if the argument list is empty, the parentheses must be included in invoking a function. Try the command

```
> c
```

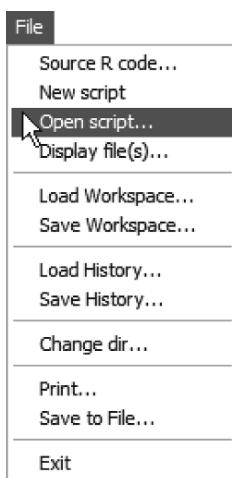
R now sees the name of an object and will simply display its contents by printing the entire set of commands making up the function in the console window. R has many useful built-in functions, including `abs`, `log`, `log10`, `exp`, `sin`, `cos`, `sqrt`, and so forth, that are useful for manipulating data. (The function `abs` computes the absolute value; `log` does the log-transformation with base e , while `log10` uses base 10; `exp` is the exponentiation function, `sin` and `cos` are the trigonometric functions; and `sqrt` computes the square root.) These functions are applied to a vector or a time series element by element. For example, `log(Dat2)` log-transforms each element of the time series `Dat2` and transfers the time series structure to the transformed data.

```
> Dat2=ts(c(12,31,22,24,30), start=c(2006,2), frequency=4)
> log(Dat2)
      Qtr1     Qtr2     Qtr3     Qtr4
2006      2.484907 3.433987 3.091042
2007      3.178054 3.401197
```

Furthermore, vectors and time series can be manipulated algebraically with the usual addition (+), subtraction (-), multiplication (*), division (/), or power (^ or **) carried out element by element. For example, applying the transformation $y = 2x^3 - x + 7$ to `Dat2` and saving the transformed data to a new time series named `new.Dat2` can be easily carried out by the command

```
new.Dat2= 2*Dat2^3-Dat2+7
```

Chapter 1 R Commands



Now, we are ready to check out selected R commands used in Chapter 1 of the book. Script files of the commands used in each of the fifteen chapters are available for download at www.stat.uiowa.edu/~kchan/TSA.htm. The script files contain the R commands needed to carry out the analyses shown in the chapters. They also contain a limited amount of additional explanation. Download the scripts and save them in your working directory. You may then open them within R in an R editor (script) window and you will save much typing! Once they are downloaded, script files may be opened by either clicking the open file button or by using the file menu shown at the left.

Exhibit 2 A Script Window with Chapter 1 Scripts Displayed

```

# Exhibit 1.1
library(TSA)
win.graph(width=4.875, height=2.5, pointsize=8)
data(larain)
plot(larain, ylab='Inches', xlab='Year', type='o')

# Exhibit 1.2
win.graph(width=3, height=3, pointsize=8)
plot(y=larain, x=zlag(larain), ylab='Inches', xlab='Previous Year Inches')

# Exhibit 1.3
win.graph(width=4.875, height=2.5, pointsize=8)
data(color)
plot(color, ylab='Color Property', xlab='Batch', type='o')

```

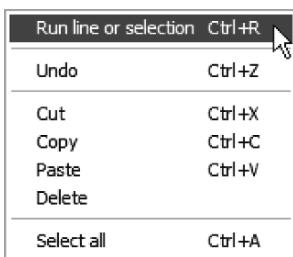


Exhibit 2 shows a portion of the script file for Chapter 1 in a script window. The first four commands have been highlighted by dragging the mouse pointer across them. They can now all be executed by either pressing Control-R (Ctrl-R) or by right-clicking the highlighted group and choosing Run from the choices displayed, as shown at the left. If the cursor is in a single command line with no highlighting, that one command may be executed similarly.

At the beginning of each session with R, you need to load the TSA library. The following command will accomplish this (but you may wish to investigate the `.First` function that can automate some startup tasks).

```
library(TSA)
```

The TSA package contains all datasets and functions needed for repeating the analyses and doing the exercises.

```
# Exhibit 1.1 on page 2.  
win.graph(width=4.875, height=2.5, pointsize=8)
```

Comments may be interspersed in the R codes to improve their readability. The `#` sign in a R command signifies that what follows the sign are comments, and hence ignored by R. The first R command opening with the `#` sign is therefore a comment. The second R command opens a window for graphics that is 4.875 inches wide and 2.5 inches tall with characters printed with point size 8. The chosen setting and similar settings produce time sequence plots that are appropriate for inclusion in the book. Other settings will be appropriate for other purposes. For example, quantile-quantile plots are best viewed with a 1:1 aspect ratio (`height = width`). For exploratory data analysis, you will want larger graphics windows to use the full resolution of your computer screen to see more detail. The command `win.graph` can be safely omitted altogether. If there is currently no open graphics window, R will open a graphics window whenever a graphics command is issued. You can resize this window in the usual ways by dragging edges or corners.

```
data(larain)
```

This loads the time series `larain` into the R session and makes it available for further analysis such as

```
plot(larain, ylab='Inches', xlab='Year', type='o')
```

`Plot` is a function. It draws the time sequence plot for `larain`. The argument `ylab='Inches'` specifies “Inches” as the label for the *y*-axis. Similarly, the label for the *x*-axis is “Year.” The argument `type` indicates how the data are displayed in the plot. For `type='o'`, the individual data points are *overplotted* on the curve; `type='b'` (for *both*) is another option that superimposes the data points on the curve, but with the curve broken around the data points. For `type='l'`, only the *line* segments connecting the points are shown. (Note: This character (1) is an “el,” not a one.) To show only the data *points*, supply the argument `type='p'`. To learn more about the `plot` function and the full options for the `type` argument, run the command

```
?plot
```

A Help window on the `plot` function will then pop up for your browsing. Try it now. What will be plotted if the option `type='h'` is used instead of `type='o'`? All graphs may be saved (File > Save as > ...) in any of several graphics formats: jpeg, pdf, etc. Saved graphs may then be imported into most word-processing programs to create high-quality reports.

```
# Exhibit 1.2 on page 2.  
win.graph(width=3, height=3, pointsize=8)  
plot(y=larain, x=zlag(larain), ylab='Inches',
```

```
    xlab='Previous Year Inches')
```

The `plot` function is a multipurpose function. It can do many different kinds of plots, depending on the set of arguments passed to it and their attributes. Here, it draws the scatter diagram of `larain` against its lag 1 values through the arguments `y=larain` (that is, `larain` on the *y*-axis) and `x=zlag(larain)` (that is, the lag 1 of `larain` is on the *x*-axis). Note that `zlag` is a function in the `TSA` package. Run the command `?zlag` to learn what you can do with it.

```
# Exhibit 1.3 on page 3.
```

```
data(color)
```

```
plot(color,ylab='Color Property',xlab='Batch',type='o')
```

Here we have supplied four arguments to the `plot` function to draw the time sequence plot of the time series `color`. The first argument is simply `color`, but the other supplied arguments are of the form name of the argument = argument value so the first supplied argument is an unnamed argument, while the other arguments are named arguments. You may wonder how an unnamed argument is interpreted by R. To understand this, use the `?plot` command to check that the argument list of the `plot` function is `x`, `y`, and You may guess that the `x` argument represents the *x*-variable, and the `y` argument for the *y*-variable in a plot. The ellipsis (...) argument stands for all other allowable arguments, which must, however, be specified with the name of the argument. (Again, consult the pages of the `plot` function to figure out which other arguments besides `x` and `y` may be passed to `plot`.) Any unnamed argument is interpreted to be the value for the argument whose order matches that of the unnamed argument supplied to the function. For example, `color` appears as the first argument supplied to the `plot` function, so R interprets it as the value for the `x` argument. Now there is no value supplied to the `y` argument. In this case, `plot` will examine the nature of the *x*-variable to determine what actions to be taken. Since `color` is a time series, `plot` draws a time sequence plot of `color`. To reinforce understanding, now try the following command in which `color` appears twice in the argument list, as the first and second arguments.

```
plot(color, color, ylab='Color Property',
      xlab='Batch',type='o')
```

Guess what will be drawn by R? Now, `color` is interpreted as the *x*-variable and also the *y*-variable; hence a 45 degree line is drawn. However, the line seems to be of nonuniform thickness. (Can you see this?) Why? It is because seeing that the variables are time series, `plot` draws the line by connecting data points in the order they are recorded, with the order of the data points marked in the plot. This feature can be useful in some analyses but in this case this feature is distracting. A remedy is to strip the time series attribute from the *x*-variables before plotting. (`Plot` takes the clue of how to do the plot from the attribute of the *x*-variable.) To temporarily turn `color` into a raw data vector, use the command

```
as.vector(color)
```

Now, try the command

```
plot(as.vector(color), color, ylab='Color Property',
      xlab='Batch',type='o')
```

```
# Exhibit 1.4 on page 4.
plot(y=color,x=zlag(color),ylab='Color Property',
     xlab='Previous Batch Color Property')
```

The `zlag` function outputs an ordinary vector; that is, `zlag(color)` is the lag 1 of `color`, but with its time series attribute stripped.

```
# Exhibit 1.9 on page 7.
plot(oilfilters,type='l',ylab='Sales')
```

`Plot` is a high-level graphics function and, as such, it will replace what is currently in the graphics window or create a new graphics window if none exists. Recall that the argument `type='l'` instructs `plot` to just draw the line segments connecting the individual time series points.

`Month=c('J','A','S','O','N','D','J','F','M','A','M','J')`

creates a vector named `Month` that contains 12 elements that represent the 12 months of the year beginning with July.

```
points(oilfilters,pch=Month)
```

`Points` is a low-level graphics function that draws on top of an existing graph. Since `oilfilters` is a time series, `points` plots `oilfilters` against time order, but the argument `pch=Month` instructs the `points` function to plot the data points using the successive values of the `Month` vector as plotting symbols. So, the first point plotted is plotted as a J, the second as an A, and so forth. When the values of `Month` are used up, they are recycled; think of `Month` being replicated as `Month, Month, Month, ...`, to make up any deficiency. So, the 13th data point is plotted as a J and the 14th as an A. What letter is used for the 30th data point?

Alternatively, the exhibit can be reproduced by the following commands

```
plot(oilfilters,type='l',ylab='Sales')
points(y=oilfilters,x=time(oilfilters),
       pch=as.vector(season(oilfilters)))
```

The `time` function outputs the epochs when the time series values were collected. The `season` function returns the month of the data in `oilfilters`; `season` is a smart function, as it returns the quarter of the data for quarterly data and so forth. The `pch` argument expects a vector as its value, but the output of the `season` function has been designed to be a `factor` object; hence the application of the `as.vector` function to `season(oilfilters)` strips its factor attribute. (See more about `factor` objects on page 435.)

A good way to appreciate the natural variation in a stochastic process is draw realizations from the process and plot them in a time sequence plot. For example, the independent and identically normally distributed process is often used as a data generating mechanism for completely random data; that is, data with no temporal structure. In other words, such data constitute a random sample from a normal distribution that are drawn sequentially over time. Simulating data from such a process and viewing their time sequence plots is a valuable exercise that can train our eyes to differentiate whether a time series is random or dependent over time, c.f. Exercise 1.3. The R command for simulating and storing in a variable named `y` a random sample of size, say $n = 48$, from

a standard normal distribution is

```
y=rnorm(48)
```

The data can then be plotted using the command

```
plot(y, type='p', ylab='IID Normal Data')
```

Try the `type='o'` option in the above command. Which plotting option do you find better to see the randomness in the data? Notice that executing the command `y=rnorm(48)` again will yield a different time series realization of the random process. The `set.seed` command discussed below addresses the issue of how to make simulations in R “reproducible.”

Data can be simulated from other distributions. For example, the command `rt(n=48, df=5)` simulates 48 independent observations from a *t*-distribution with 5 degrees of freedom. Similarly, `rchisq(n=48, df=2)` simulates a realization of size 48 from the chi-square distribution with 2 degrees of freedom.

Chapter 2 R Commands

We show some R code to simulate your own random walk with, say, 60 independent standard normal errors.

```
# Exhibit 2.1 on page 14.  
n=60
```

This assigns the value of 60 to the object named `n`.

```
set.seed(12345)
```

This initializes the random number generator so that the simulation is reproducible if needed.

```
sim.random.walk=ts(cumsum(rnorm(n)), freq=1, start=1)
```

The expression `rnorm(n)` generates `n` independent values from the standard normal distribution. The function `cumsum` then computes the vector of cumulative sums of the normally distributed sample, resulting in a random walk realization. The random walk realization is then given the attribute of a time series and saved into the object named `sim.random.walk`.

```
plot(sim.random.walk, type='o', ylab='Another Random Walk')  
plots the simulated random walk.
```

Chapter 3 R Commands

We now move to discuss some of the R commands appearing in Chapter 3.

```
# Exhibit 3.1 on page 31.  
data(rwalk)
```

This command loads the time series `rwalk`, which is a random walk realization.

```
model1=lm(rwalk~time(rwalk))
```

The function `lm` fits a *linear model* (a regression model) with its first argument being a formula. A formula is an expression including a tilde sign (~), the left-hand side of which is the response variable and the right-hand side are the covariates or explanatory variables (separated by plus signs if there are two or more covariates). By default, the intercept term is included in the model. The intercept can be removed by including the term “`-1`” on the right-hand side of the tilde sign. Recall that `time(rwalk)` yields a time series of the time epochs at which the random walk was sampled. So the command `lm(rwalk~time(rwalk))` fits a time trend regression model to the `rwalk` series. The model fit is saved as the object named `model1`.

```
summary(model1)
```

The function `summary` prints out a summary of the fitted model passed to it. Hence the command above prints out the fitted time trend regression model for `rwalk`.

```
# Exhibit 3.2 on page 31.
plot(rwalk,type='o',ylab='y')
abline(model1)
```

The function `abline` is a low-level graphics function. If a fitted simple regression model is passed to it, it adds the fitted straight line to an existing graph. Any straight line of the form $y = \beta_0 + \beta_1x$ can be superimposed on the graph by running the command

```
abline(a=beta0,b=beta1)
```

For example, the following command adds a 45 degree line on the current graph.

```
abline(a=0,b=1)
```

Recall the `lm` function can fit multiple regression models, with the covariates or explanatory variables specified one by one, on the right side of the tilde sign (~) in the formula. The covariates must be separated with a plus sign (+). Suppose we want to fit a quadratic time trend model to the `rwalk` series. We need to create a new covariate that contains the square of the time indices. The quadratic variable may be created before invoking the `lm` function. Or it may be created on the fly when invoking the `lm` function. The latter approach is illustrated here.

```
model1a=lm(rwalk~time(rwalk)+I(time(rwalk)^2))
```

Notice that the expression `time(rwalk)^2` is enclosed within the `I` function which instructs R to create a new variable by executing the command passed into the `I` function. The fitted quadratic trend model can be inspected with the `summary` function.

```
> summary(model1a)
Call:
lm(formula = rwalk ~ time(rwalk) + I(time(rwalk)^2))
Residuals:
    Min      1Q  Median      3Q     Max 
-2.696232 -0.768018  0.008256  0.853365  2.344685 
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.4272911  0.4534893 -3.147  0.00262 ** 
time(rwalk)   0.1746746  0.0343028  5.092 4.16e-06 *** 
I(time(rwalk)^2) -0.0006654  0.0005451 -1.221  0.22721  
---

```

```
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 1.132 on 57 degrees of freedom
Multiple R-Squared: 0.8167, Adjusted R-squared: 0.8102
F-statistic: 127 on 2 and 57 DF, p-value: < 2.2e-16
```

The `summary` function repeats the function call to the `lm` function. It then prints out the five-number numerical summary of the residuals, followed by a table of the parameter estimates with their standard errors, *t*-values and *p*-values. All significant covariates are marked with asterisks (*); more asterisks means higher significance, that is, smaller *p*-value, as explained in the line labeled as `Signif. codes`. Finally, it outputs the residual standard error, that is, the noise standard deviation estimate, and the multiple R-squared of the fitted model. Clearly, the quadratic term is not significant so that it is not needed, as is also obvious from the time plot of the series.

The reader may wonder why the `I` function is needed. This is because without the `I` function, R interprets the term `time(rwalk) + time(rwalk)^2` using the formula convention (`run ?formula` to learn more about the formula convention), which results in fitting the linear trend model! Refit the quadratic trend model but now omit the `I` function in the R command, and compare the model fit with those of the linear and quadratic trend models.

```
# Exhibit 3.3 on page 32.
data(tempdub)
```

This loads the `tempdub` series. You can learn more about the dataset `tempdub` by running the command `?tempdub`.

```
month.=season(tempdub)
```

The expression `season(tempdub)` outputs the monthly index of `tempdub` as a factor, and saves it into the object `month..`. The first period sign (.) is part of the name (`month..`) and is included to make the printout from later commands more clear.

We now digress to explain what a factor is. A `factor` is a kind of data structure for handling qualitative (nominal) data that do not have a natural ordering like numbers do. However, for purposes of summary and graphics, the user may supply the `levels` argument to indicate an ordering among the factor values. For example, the following command creates a factor containing the qualitative variable `sex`, with the default ordering using the dictionary order.

```
> sex=factor(c('M','F','M','M','F'))
> sex
[1] M F M M F
Levels: F M
```

We can change the ordering as follows:

```
> sex=factor(c('M','F','M','M','F'),levels=c('M','F'))
> sex
[1] M F M M F
Levels: M F
```

Note the swap of F and M in the levels. The function `table` counts the frequencies of the two sexes.

```
> table(sex)
sex M F
 3 2
```

The printout lists the frequencies of the values according to the order supplied in the level argument. Now, we return to the R scripts in Chapter 3.

```
model2=lm(tempdub~month.-1)
```

Recall that month is a factor containing the month of the data. When a formula contains a factor covariate, the function `lm` replaces the factor variable by a set of indicator variables corresponding to each distinct level (value) of the factor. Here, `month.` has 12 distinct levels: Jan, Feb,..., and so forth. So, in place of `month.`, `lm` creates 12 monthly indicator variables and replaces `month.` by the 12 indicator variables. Because these 12 indicator variables are linearly dependent (they add up to a vector of all ones), the intercept term has to be removed to avoid multicollinearity. The expression “`-1`” in the formula takes care of this. The fitted model corresponds to fitting a mean separately for each month. If the expression “`-1`” is omitted, `lm` deals with the multicollinearity by omitting the first indicator variable; that is, the indicator variable for January will be deleted. In such a fitted model, the intercept represents the overall January mean and the coefficients for other months are the deviations of their means from the January mean.

```
summary(model2)
```

A summary of the fitted regression model is printed out with this command. Many variables derived from the fitted model can also be easily obtained. For example, the fitted values can be printed as

```
fitted(model2)
```

whereas residuals are obtained by using

```
residuals(model2)
```

```
# Exhibit 3.4 on page 33.
```

```
model3=lm(tempdub~month.) # intercept is automatically
                           included so one month (January) is dropped
summary(model3)
```

```
# Exhibit 3.5 on page 35.
```

```
har.=harmonic(tempdub,1)
```

The first pair of harmonic functions (sine and cosine pairs) can be constructed by the `harmonic` function, which takes a time series as its first argument and the number of harmonic pairs as its second argument. Run `?harmonic` to learn more about this function. The output of the harmonic function is a matrix that is saved into an object named `har..` Again, the first period is part of the name and included to make the later printouts clearer.

```
model4=lm(tempdub~har.)
summary(model4)
```

We now briefly discuss the use of matrices in R. A matrix is a rectangular array of numbers. It can be created by the `matrix` function. Here is an example:

```
> M=matrix(1:6,ncol=2)
> M
[,1] [,2]
[1,]     1     4
[2,]     2     5
[3,]     3     6
```

The `matrix` function expects a vector as its first argument, and it uses the values in the supplied vector to fill up a matrix column by column. The column dimension of a matrix is specified by the `ncol` argument and the row dimension by the `nrow` argument. The expression `1:6` stands for the vector containing the integers from 1 to 6. So the `matrix` function creates a matrix consisting of two columns using the six numbers 1, 2, 3, 4, 5, and 6. Since the row dimension is missing, R assumes that the matrix has six elements and hence the missing row dimension is set to 2. The dimensions of a matrix can be extracted using the `dim` function.

```
> dim(M)
[1] 3 2
```

This displays the row and column dimensions of `M` as a vector. The function `apply` can process a matrix column by column, with each column operated by a supplied function. For example, the column means of `M` can be computed as follows:

```
> apply(M, 2, mean)
[1] 2 5
```

The first argument of the `apply` function is the matrix on which it processes, and the second argument is `MARGIN`, which should be set to 1 for row processing or 2 for column processing. The third argument is `FUN`, which takes the user-specified function. The example above instructs R to process `M` column by column and apply the `mean` function to each column. How would you modify the preceding R command to compute the row sums of `M`?

```
# Exhibit 3.6 on page 35.
plot(ts(fitted(model4), freq=12, start=c(1964,1)),
      ylab='Temperature', type='l',
      ylim=range(c(fitted(model4), tempdub)))
points(tempdub)
```

The `ylim` option ensures that the y-axis has a range that includes both the raw data and the fitted values.

```
# Exhibit 3.8 on page 43.
plot(y=rstudent(model3), x=as.vector(time(tempdub)),
      xlab='Time', ylab='Standardized Residuals', type='o')
```

The expression `rstudent(model3)` returns the (externally) Studentized residuals from the fitted model. To compute the (internally) standardized residuals, use the command `rstandard(model3)`.

```
# Exhibit 3.11 on page 45.
hist(rstudent(model3), xlab='Standardized Residuals')
```

The function `hist` draws a histogram of the data passed to it as the first argument. Note that the default heading of the histogram says that the plot is a histogram of

`rstudent(model3)`. While the default main label correctly depicts what is plotted, it is often desirable to have a less technical but more descriptive label; for example, setting the option `main='Histogram of the Standardized Residuals'`.

```
# Exhibit 3.12 on page 45.
qqnorm(rstudent(model3))
```

The expression `rstudent(model3)` extracts the standardized residuals of `model3`. The `qqnorm` function then plots the Q-Q normal scores plot of the residuals. A reference straight line can be superimposed on the Q-Q normal score plot by running the command `qqline(rstudent(model3))`.

```
# Exhibit 3.13 on page 47.
acf(rstudent(model3))
```

The `acf` function computes the sample autocorrelation function of the time series supplied to the function. The maximum number of lags is determined automatically based on the sample size. It can, however, be changed to, say, 30 by setting the option `max.lag=30` when calling the function.

The Shapiro-Wilk test and the runs test on the residuals can be carried out respectively by the following commands.

```
shapiro.test(rstudent(model3))
runs(rstudent(model3))
```

These commands compute the test statistics as well as their corresponding *p*-values.

Chapter 4 R Commands

```
# Exhibit 4.2 on page 59.
data(ma1.2.s)
plot(ma1.2.s, ylab=expression(Y[t]), type='o')
```

The software R can display mathematical symbols in a graph. The option `ylab=expression(Y[t])` specifies that the y label is Y with t as its subscript, all in math font. Typesetting a formula does require some additional work. Read the help pages for `legend` (`?legend`) and run the command `demo(mathplot)` to learn more about this topic.

An MA(1) series with MA coefficient equal to $\theta_1 = -0.9$ and of length $n = 100$ can be simulated by the following commands.

```
set.seed(12345)
```

This command initializes the seed of the random number generator so that a simulation can be reproduced if needed. Without this command, the random generator will initialize “randomly,” and there is no way to reproduce the simulation. The argument 12345 can be replaced by other numbers to obtain different random numbers.

```
y=arima.sim(model=list(ma=-c(-0.9)), n=100)
```

The `arima.sim` function simulates a time series from a given ARIMA model passed into the function as a list that contains the AR and MA parameters as vectors. The simulated model above is an MA(1) model, so there is no AR part in the model list. The soft-

ware R uses a plus convention in parameterizing the MA part, so we have to add a minus sign before the vector of MA values to agree with our parameterization. The sample size is determined by the value of the argument n . So, the command above instructs R to simulate a realization of size 100 from an MA(1) model with $\theta_1 = -0.9$.

We now digress to explain some pertinent facts about `list`. A list is the most flexible data structure in R. You may think of a list as a cabinet with many drawers (elements or components), each of which contains data with possibly different data structures. For example, an element of a list can be another list! The elements of a list are ordered according to the order they are entered. Also, elements can be named to facilitate their easy retrieval. A list can be created by the `list` function with elements supplied as its arguments. The elements may be passed into the `list` function in the form of `name = value`, delimited by commas. Below is an example of a list containing three elements named `a`, `b`, and `c`, where `a` is a three-dimensional vector, `b` is a number, and `c` is a time series.

```
> list1=list(a=c(1,2,3),b=4,c=ts(c(5,6,7,8),
  start=c(2006,2),frequency=4))
> list1
$a
[1] 1 2 3
$b
[1] 4
$c   Qtr1 Qtr2 Qtr3 Qtr4
2006      5   6   7
2007      8
```

To retrieve an element of a list, run the command `listname$elementname`, for example

```
> list1$c
  Qtr1 Qtr2 Qtr3 Qtr4
2006      5   6   7
2007      8
```

Data of irregular structure can be stored as a list. The output of a function is often a list. Simply entering the name of a list may result in dazzling output if the printed list is large. An alternative is to first explore the structure of a list by the function `str` (`str` stands for *structure*). An example follows.

```
> str(list1)
List of 3
 $ a: num [1:3] 1 2 3
 $ b: num 4
 $ c: Time-Series [1:4] from 2006 to 2007: 5 6 7 8
```

This shows that `list1` has three elements and describes these elements briefly.

Chapter 5 R Commands

```
# Exhibit 5.4 on page 91.
plot(diff(log(oil.price)),ylab='Change in Log(Price)',
  type='l')
```

The function `diff` outputs the first difference of the supplied time series. Higher-order differences can be computed by supplying the `differences` argument. For example, the second difference of `log(oil.price)` can be computed by the command

```
diff(log(oil.price), differences=2)
```

A useful convention of R is that the name of an argument in a function can be abbreviated if it does not result in ambiguity. For example, the previous command can be shortened to

```
diff(log(oil.price), diff=2)
```

Note that the second argument of the `diff` function is the lag argument. By default, `lag=1` and the `diff` function computes regular differences—first or higher differences. Later, when we deal with seasonal time series data, it will sometimes be desirable to consider seasonal differences. For example, we may want to subtract this month's number from the number of the same month one year ago; that is, the differences are computed with a lag of 12 months. This can be done by specifying `lag=12`. As an illustration, computing the seasonal differences of period 12 can be done by issuing the command `diff(tempdub, lag=12)`. What will be computed by the command `diff(log(oil.price), 2)`? One of the authors (KSC) committed a serious error, more than once, when he tried to compute the second regular differences of some time series by running a similar command with unnamed arguments. Instead of the second regular differences, the first seasonal differences of lag 2 were actually computed by the command with unnamed arguments! Imagine his frustrations of many anxious hours, all because the data analysis from the flawed computations seriously conflicted with expectations based on theory! The moral is that passing unnamed arguments to a function is risky unless you know the positions of the relevant arguments very well. It is well to remember that unnamed arguments, if present, should appear together in the beginning part of the argument list, and there should be no unnamed argument after a named one. Indeed, mixed arguments (some named and some unnamed in a haphazard order) may result in erroneous interpretation by R. The order of the arguments in a function can be quickly checked by running the command `args(function.name)` or `?function.name`, where `function.name` should be replaced by the name of the function you are checking.

```
# Exhibit 5.11 on page 102.
library(MASS)
```

This loads the library MASS. Run the command `library(help=MASS)` to see the content of this library.

```
boxcox(lm(electricity~1))
```

The function `boxcox` computes the maximum likelihood estimate of the power transformation on the response variable to make a linear regression model appropriate for the data. The first argument is a fitted model by the `lm` function. By default, the `boxcox` function produces a plot of the log-likelihood function of the power parameter. The MLE of the power parameter is the value that maximizes the plotted likelihood curve. Here the model is that some power transform of electricity is given by the model of a constant mean plus normally distributed white noise. But we already know that elec-

tricity is serially correlated, so this method is not entirely correct, as the autocorrelation in the series is not accounted for.

For time series analysis, a more appropriate model is that some power transform of the time series variable follows an AR model. The function `BoxCox.ar` implements this approach. It has two drawbacks in that it is much more computer-intensive and that other covariates cannot be included in the model in the current version of the function. The first argument of `BoxCox.ar` is the name of the time series variable. The AR order may be supplied by the user through the `order` argument. If the AR order is missing, the function estimates the AR order by minimizing the AIC for the log-transformed data. Both `boxcox` and `BoxCox.ar` require the response variable to be positive.

```
BoxCox.ar(electricity)
```

This plots the log-likelihood function of the power parameter for the model that accounts for autocorrelation in the data.

Chapter 6 R Commands

```
# Exhibit 6.9 on page 120.  
acf(ma2.s, ci.type='ma', xaxp=c(0,20,10))
```

The argument `ci.type='ma'` instructs R to plot the sample ACF with the confidence band for the k th lag ACF computed based on the assumption of an $\text{MA}(k - 1)$ model. See Equation (6.1.11) on page 112 for details.

```
# Exhibit 6.11 on page 121.  
pacf(ar1.s, xaxp=c(0,20,10))
```

This calculates and plots the sample PACF function. Run the command `?pacf` to learn more about the `xaxp` argument.

```
# Exhibit 6.17 on page 124.  
eacf(arma11.s)
```

This computes the sample EACF function (extended autocorrelation function) of the data `arma11.s`. The maximum AR and MA orders can be set via the `ar.max` and `ma.max` arguments. Their default values are seven and thirteen, respectively. For example, `eacf(arma11.s, ar.max=10, ma.max=10)` computes the EACF with maximum AR and MA orders of 10. The EACF function prints a table of symbols with X standing for a significant value and O a nonsignificant value.

```
library(uroot)
```

This loads the `uroot` library and the following commands illustrate the computation of the Dickey-Fuller unit-root test.

```
ar(diff(rwalk))
```

This command finds the AR order for the differenced series, which is order 8, by the minimum AIC criterion.

```
ADF.test(rwalk, selectlags=list(mode=c(1,2,3,4,5,6,7,8),
  Pmax=8), itsd=c(1,0,0))
```

This computes the ADF test for the data `rwalk`. The `selectlags` argument takes a list as its value. The `mode` argument specifies which lags must be included, and if it is absent, then the `Pmax` argument sets the maximum lag and the `ADF.test` function determines which lags to include in the test using several methods by setting the `mode` to `signif`, `aic`, or `bic`. The option `signif` is the default value for `mode`, which estimates a subset AR model by retaining only significant lags. The argument `itsd` expects a vector; the first two elements are binary, indicating whether to include a constant term (if the first element is 1) or a linear time trend (if the second element is 1); and the third element zero if there are no more covariates to include in the model. See the help pages for the `ADF.test` function to learn more about it. Hence, the R command instructs `ADF.test` to carry out the test with the null hypothesis that the model has a unit root and an intercept term. The alternative is that the model is stationary, so a small *p*-value implies stationarity!

```
ADF.test(rwalk, selectlags=list(Pmax=0), itsd=c(1,0,0))
```

In comparison, the preceding command carries out the ADF test with the null hypothesis being that the model has a unit root, an intercept but no other lags, whereas the alternative specifies that the model is a stationary AR(1) model with an intercept. If `itsd=c(0,0,0)`, then the alternative model is a centered stationary AR(1) model, that is, with zero mean. Such a hypothesis is not relevant unless the data are already mean-corrected.

```
# Exhibit 6.22 on page 132.
set.seed(92397)
test=arima.sim(model=list(ar=c(rep(0,11),.8),
  ma=c(rep(0,11),0.7)),n=120)
```

This simulates a subset ARMA model. Here `rep(0,11)` stands for a sequence of 11 zeros.

```
res=armasubsets(y=test,nar=14,nma=14,y.name='test',
  ar.method='ols')
```

The `armasubsets` function computes various subset ARMA models, with the maximum AR and MA orders specified by the `nar` and `nma` arguments, both set as 14 in the example above. The associated AR models are estimated by the default method of `ols` (ordinary least squares).

```
plot(res)
```

The `plot` function is a smart function. Seeing that `res` is the output from the `armasubsets` function, it draws a table indicating several of the best subset ARMA models.

Chapter 7 R Commands

Below is a function that computes the method-of-moments estimator of the MA(1) coefficient of an MA(1) model. It is a simple example of an R function. Simply copy and

paste it into the R console. Press the enter key to compile the code, and the function `estimate.ma1.mom` will be created and then be available for use in your workspace. This function only exists in the particular workspace where it was created.

```
estimate.ma1.mom=function(x){r=acf(x,plot=F)$acf[1];
  if (abs(r)<0.5) return((-1+sqrt(1-4*r^2))/(2*r))
  else return(NA)}
```

Readers uninterested in the specifics of R programming may skip down to the material on Exhibit 7.1. The syntax of an R function takes the form

```
function.name = function(argument list) {function body}
```

where `function body` is a set of R statements (commands). Normally, complete R commands are separated by line breaks. Alternatively, they may be separated by the semicolon symbol (`;`). If an R command is incomplete, R will assume that it is to be continued on the next line and so forth until R reads a complete command. So the function above has a single argument called `x` and contains two commands. The first one is

```
r=acf(x,plot=F)$acf[1]
```

which instructs R to compute the acf of `x` without plotting the values, extract the first element of the computed sample acf function (that is, the lag 1 autocorrelation) and then save it in an object called `r`. The object `r` is a local object; it only exists within the `estimate.ma1.mom` function environment. The second command is

```
if (abs(r)<0.5)
  return((-1+sqrt(1-4*r^2))/(2*r)) else return(NA)
```

Note the line break after the `if` clause and the second half of the command. Since the `if` clause alone is incomplete, R assumes that it is to be continued on the next line. With the second line, R finds a complete R command and so concludes the two lines of commands together as a complete command. In other words, R sees the next command as equivalent to the following one line:

```
if (abs(r)<0.5) return((-1+sqrt(1-4*r^2))/(2*r)) else return(NA)
```

The function `abs` computes the absolute value of the argument passed to it, whereas `sqrt` is the function that computes the square root of its argument. Now, we are ready to interpret the second command: if the absolute value of `r`, the lag 1 autocorrelation of `x`, is less than 0.5 in magnitude, the function returns the number

$$(-1 + \sqrt{1 - 4r^2})/(2r)$$

which is the method-of-moments estimator of the MA(1) coefficient θ_1 ; otherwise the function returns NA (see Equation (7.1.4) on page 150). The symbol NA is the code standing for a missing value in R. (NA stands for *not available*.) In this example, R is specifically instructed what value to return to the user. However, the default procedure is that a function returns the value created by the last command in the function body. R provides a powerful computer language for doing statistics. Please consult the documents on the R Website to learn more about R programming.

```
# Exhibit 7.1 on page 152.
data(ma1.2.s)
```

This loads a simulated MA(1) series.

```
estimate.ma1.mom(ma1.2.s)
```

This computes the MA(1) coefficient estimate by the method of moments using the user-created `estime.ma1.mom` function above!

```
data(ar1.s)
```

This loads a simulated AR(1) series from the TSA package.

```
ar(ar1.s,order.max=1,AIC=F,method='yw')
```

This computes the AR coefficient estimates for the `ar1.s` series. The `ar` function estimates the AR model for the centered data (that is, mean-corrected data), so the intercept must be zero and not estimated or printed out in the output. The `ar` function requires the user to specify the maximum AR order through the `order.max` argument. The AR order may be estimated by choosing the order, between 0 and the maximum order, whose model has the smallest AIC. This option can be specified by setting the `AIC` argument to take the true value, that is, `AIC=T`. Or we can switch off order selection by specifying `AIC=F`. In the latter case, the AR order is set to the maximum AR order. The `ar` function can estimate the AR model using a number of methods, including solving the Yule-Walker equations, ordinary least squares, and maximum likelihood estimation (assuming normally distributed white noise error terms). These correspond to setting the option `method='yw'`, `method='ols'`, or `method='mle'`, respectively. In particular, the preceding R command fits an AR(1) model for the `ar1.s` series by solving the Yule-Walker equation.

We digress briefly to discuss the concept of a logical variable, which can take the value TRUE or FALSE. These values can be abbreviated as T and F. In binary representation, T is also represented by 1 and F by 0. R adopts the useful convention that a logical variable appearing in an arithmetic expression will be automatically converted to 1 if it is a T and 0 otherwise.

```
# Exhibit 7.6, page 165.
data(arma11.s)
arima(arma11.s, order=c(1,0,1),method='CSS')
```

The `arima` function estimates an ARIMA(p,d,q) model for the time series passed to it as the first argument. The ARIMA order is specified by the `order` argument, `order=c(p,d,q)`, so the command above fits an ARMA(1,1) model to the data. Estimation can be carried out by the conditional sum-of-squares method (`method='CSS'`) or maximum likelihood (`method='ML'`). The default estimation method is maximum likelihood, with initial values determined by the CSS method. The `arima` function prints out a summary of the fitted model. The fitted model may also be saved as an object that can be further manipulated, for example, for model diagnostics. By default, if $d = 0$, a stationary ARMA model will be fitted. Also, the fitted model is in the centered form; that is, an ARMA model fitted to the series minus its sample mean. The intercept term reported in the output of the `arima` function is a misnomer, as it is in fact the mean! However, the mean so estimated generally differs slightly from the sample mean.

```
# Exhibit 7.10 on page 168.
res=arima(sqrt(hare),order=c(3,0,0))
```

This saves the fitted AR(3) model in the object named `res`. The output of the `arima` function is a `list`. Run the command `str(res)` to find out what is saved in `res`. You will find that most of the things in `res` are not directly useful. Instead, the output of the `arima` function has to be processed by other functions for more informed summaries. For example, (raw) residuals from the fitted model can be computed by the `residuals` function via the command `residuals(res)`. Fitted values can be obtained by running `fitted(res)`. Other useful functions for processing a fitted ARIMA model from the `arima` function will be discussed below.

The empirical approach of using the bootstrap to do inference is illustrated below.

```
set.seed(12345)
```

This initializes the seed of the random number generator so that the simulation study can be repeated.

```
coefm.cond.norm=arima.boot(res,cond.boot=T,is.normal=T,
  B=1000,init=sqrt(hare))
```

The `arima.boot` function carries out a bootstrap analysis based on a fitted ARIMA model. Its first argument is a fitted ARIMA model, that is, the output from the `arima` function. Four different bootstrap methods are available: The bootstrap series can be initialized by a supplied value (`cond.boot=T`) or not (`cond.boot=F`), and a nonparametric bootstrap (`is.normal=F`) or a parametric bootstrap assuming normal innovations (`is.normal=T`) can be used. For a conditional bootstrap, the initial values can be supplied as a vector (the `arima.boot` function will use the initial values from the supplied vector). The bootstrap sample size, say 1000, is specified by the `B=1000` option. The function `arima.boot` outputs a matrix with each row being the bootstrap estimate of the ARIMA coefficients obtained by maximum likelihood estimation with the bootstrap data. So, if `B=1000` and the model is an AR(3), then the output is a 1000 by 4 matrix where each row consists of the bootstrap AR(1), AR(2), and AR(3) coefficients plus the mean estimate in that order ($\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3, \hat{\mu}$).

```
signif(apply(coefm.cond.norm,2,function(x)
  {quantile(x,c(.025,.975),na.rm=T)}),3)
```

This is a compound R statement. It is equivalent to the two commands

```
temp=apply(coefm.cond.norm,2,function(x)
  {quantile(x,c(.025,.975),na.rm=T)})
signif(temp,3)
```

except that the temporary variable `temp` is not created in the original compound statement. Recall that the `apply` function is a general-purpose function for processing a matrix. Here the `apply` function processes the matrix `coefm.cond.norm` column by column, with each column supplied to the no-name user-supplied function

```
function(x){quantile(x,c(.025,.975),na.rm=T)}
```

This no-name function has one input, called `x`, that is processed by the `quantile` function. The `quantile` function takes a vector and computes the sample quantiles with the corresponding probability specified in the second argument. The third argu-

ment of the quantile function is specified as `na.rm=T` (`na` stands for not available and `rm` means remove), which means that any missing values in the input are discarded before computing the quantiles. This specification is pivotal because by default any quantile of a dataset with some missing values is defined to be a missing value (`NA`) in R. (Some bootstrap series may have convergence problems upon fitting an ARIMA model and hence the output of the `bootstrap` function may contain some missing values.) To return to the interpretation of the command on the right-hand side of `temp`, it instructs R to compute the 2.5th and 97.5th percentiles of each bootstrap coefficient estimate. To enable precise calculations, R maintains many significant digits in the numbers stored in an object. The printed version, however, usually requires fewer significant digits for clarity. This can be done by the `signif` function. The `signif` function outputs the object passed into it as first argument, but only to the number of significant digits specified in the second argument, which is three in the example. Altogether, the compound R command computes the 95% bootstrap confidence intervals for each AR coefficient.

Chapter 8 R Commands

```
# Exhibit 8.2 on page 177.
data(hare)
m1.hare=arima(sqrt(hare),order=c(3,0,0))
m1.hare
```

This prints the fitted AR(3) model for the square-root-transformed hare data. The AR(2) coefficient estimate ($\hat{\phi}_2$) turns out not to be significant. Note that the AR(2) coefficient is the second element in the coefficient vector, as shown in the printout of the fitted model. A constrained ARIMA model with some elements fixed at certain values can be fitted by using the `fixed` argument in the `arima` function. The `fixed` argument should be a vector of the same length as the coefficient vector and its elements set to `NA` for all of the free elements but set to zero (or another fixed value) for all of the constrained coefficients. For example, here the AR(2) coefficient is constrained to be zero ($\phi_2 = 0$) and hence `fixed=c(NA, 0, NA, NA)`, that is, the AR(1), AR(3), and the “intercept” term are free parameters, whereas the AR(2) is fixed at 0. Remember that the “intercept” term is last. Below is the command for fitting the constrained AR(3) model for the hare data.

```
m2.hare=arima(sqrt(hare),order=c(3,0,0),
               fixed=c(NA,0,NA,NA))
m2.hare
```

Note that the intercept term is actually the mean in the centered form of the ARMA model; that is, if $y = \sqrt{hare} - \text{intercept}$, then the model is

$$y_t = 0.919y_{t-1} - 0.5313y_{t-3} + e_t$$

so the “true” estimated intercept equals $5.6889*(1 - 0.919 + 0.5313) = 3.483$, as stated in the text!

```
plot(rstandard(m2.hare),
      ylab='Standardized Residuals', type='b')
```

The function `rstandard` computes the standardized residuals; that is, the raw residuals normalized by the estimated noise standard deviation.

```
abline(h=0)
```

adds a horizontal line to the plot with zero y-intercept. Use the help in R to find out how to add a vertical line with x -intercept = 10.

```
# Exhibit 8.12 on page 185 (prefaced by some commands in
# Exhibit 8.1 on page 176)
data(color)
m1.color=arima(color,order=c(1,0,0))
tsdiag(m1.color,gof=15,omit.initial=F)
```

The `tsdiag` function in the `TSA` package has been modified from that in the `stats` package of R. It performs model diagnostics on a fitted model. The argument `gof` specifies the maximum number of lags in the `acf` function used in the model diagnostics. Setting the argument `omit.initial=T` omits the few initial residuals from the analysis. This option is especially useful for checking seasonal models where the initial residuals are close to zero by construction and including them may skew the model diagnostics. In the example, the `omit.initial` argument is set to be `F` so that the diagnostics are done with all residuals. Recall that the Ljung-Box (portmanteau) test statistic equals the weighted sum of the squared residual autocorrelations from lags 1 to K, say; see Equation (8.1.12) on page 184. Assuming that the ARIMA orders are correctly specified, the validity of the approximate chi-square distribution for the Ljung-Box test statistic requires that K be larger than the lag beyond which the original time series has negligible autocorrelation. The modified `tsdiag` function in the `TSA` package checks this requirement; consequently the Ljung-Box test is only computed for sufficiently large K. If the required K is larger than the specified maximum lag, `tsdiag` will return an error message. This problem can be solved by increasing the maximum lag asked for. Use `?tsdiag` to learn more about the modified `tsdiag` function.

Chapter 9 R Commands

```
# Exhibit 9.2 on page 205.
```

```
data(tempdub)
tempdub1=ts(c(tempdub, rep(NA, 24)), start=start(tempdub),
freq=frequency(tempdub))
```

This appends two years of missing values to the `tempdub` data, as we want to forecast the temperature for two years into the future. The function `start` extracts the starting date of a time series. The function `frequency` extracts the frequency of the time series passed to it, here being 12. Hence, `tempdub1` contains the Dubuque temperature series augmented by two years of missing data, with the same starting date and frequency of sampling per unit time interval.

```
har.=harmonic(tempdub, 1)
```

This creates the first pair of harmonic functions.

```
m5.tempdub=arima(tempdub,order=c(0,0,0),xreg=har.)
```

This fits the harmonic regression model using the `arima` function. The covariates are passed to the function through the `xreg` argument. In the example, `har.` is the covariate and the `arima` function fits a linear regression model of the response variable on the covariate, with the errors assumed to follow an ARIMA model. Because the specified ARIMA orders $p = d = q = 0$, the presumed error structure is white noise; that is, the `arima` function fits an ordinary linear regression model of `tempdub` on the first pair of harmonic functions. Note that the result is the same as that from the fit using the `lm` function, which can be verified by the following commands:

```
har.=harmonic(tempdub,1); model4=lm(tempdub~har.)
summary(model4)
```

The `xreg` argument expects the covariate input either as a matrix or a `data.frame`. A `data.frame` can be thought of as a matrix made up by binding together several covariates column by column. It can be created by the `data.frame` function with multiple arguments, each of which takes the form `covariate.name = R statement` for computing the covariate. If the `covariate.name` is omitted, the `R statement` becomes the covariate name, which may be undesirable for a complex defining statement. If the `R statement` is a matrix, its columns are taken as covariates with the column names taken as the covariate names. Consider the example of augmenting the harmonic regression model above by a linear time trend. The augmented model can be fitted by the command

```
arima(tempdub,order=c(0,0,0),
      xreg=data.frame(har.,trend=time(tempdub)))
m5.tempdub
```

This prints the fitted model.

We now illustrate prediction with an example.

```
newhar.=harmonic(ts(rep(1,24), start=c(1976,1), freq=12),1)
```

This creates the harmonic functions over two years starting from January 1976. Remember that the `tempdub` series ends in December 1975.

```
plot(m5.tempdub,n.ahead=24,n1=c(1972,1),newxreg=newhar.,
     col='red', type='b',ylab='Temperature',xlab='Year')
```

This computes and plots the forecasts based on the fitted model passed as the first argument. Here, we specify a forecast for 24 steps ahead through the argument `n.ahead=24`. The covariate values over the period of forecast have to be supplied by the `newxreg` argument. The `newxreg` argument should match the `xreg` argument in terms of the covariates except that their values are from different periods. The plot may be drawn with a starting date different from the start date of the time series data by using the `n1` argument. Here, `n1=c(1972,1)` specifies January 1972 as the start date for the plot. For nonseasonal data (that is, frequency = 1), `n1` should be a scalar. The `col` and `type` arguments refer to the color and style of the plotted lines.

```
# Exhibit 9.3 on page 206.
data(color)
m1.color=arima(color,order=c(1,0,0))
```

```
plot(m1.color,n.ahead=12,col='red',type='b',xlab='Year',
     ylab='Temperature')
abline(h=coef(m1.color)
      [names(coef(m1.color))=='intercept'])
```

The final command adds the horizontal line at the estimated mean (intercept). This is a complex statement. The expression `coef(m1.color)` extracts the coefficient vector. The components of the coefficient vector are named. The names of a vector can be extracted by the `names` function, so `names(coef(m1.color))` returns the vector of names of the components of the coefficient vector. The `==` operator compares the two vectors on its two sides element by element, resulting in a vector consisting of TRUEs and FALSEs depending on whether the elements are equal or not. (If the vectors under comparison are of unequal length, R recycles the shorter one repeatedly to match the longer one.) Hence, the command

```
[names(coef(m1.color))=='intercept']
```

returns a vector with the TRUE value in the position in which the “intercept” component lies and with all other elements FALSE. Finally, the intercept coefficient estimate is extracted by the “bracket” operation:

```
coef(m1.color)[names(coef(m1.color))=='intercept']
```

The operation within brackets subsets a vector using one of two mechanisms. Let `v` be a vector. A subvector of it can be formed by the command `v[s]`, where `s` is a Boolean vector, (that is, consisting of TRUEs and FALSEs) that is of the same length as `v`. The vector `v[s]` is then a sub-vector of `v` consisting of those elements of `v` for which the corresponding element in `s` is TRUE; elements in `v` whose corresponding element in `s` is FALSE are discarded from `v[s]`.

A second way to subset a vector is to construct `s` so that it contains the position of the elements to be retained and `v[s]` will return the desired subvector. A variation of this approach is to form a subvector by deletion. Unwanted elements are designated by giving their positions multiplied by -1. An illustration follows.

```
> v=1:5
```

This creates a vector containing the first five positive integers.

```
> v
[1] 1 2 3 4 5
> names(v)
NULL
```

By default, the components of `v` are unnamed, so `names(v)` returns an empty vector denoted by the object `NULL`.

```
> names(v)=c('A','B','C','D','E')
```

This is the method of assigning names to the components of a vector.

```
> v
A B C D E
1 2 3 4 5
```

The command

```
> names(v)=='C'
```

```
[1] FALSE FALSE TRUE FALSE FALSE
finds which components of names (v) is "C."
```

The command

```
> v[names(v) == 'C']
C
3
```

subsets v by Boolean extraction.

The command

```
> v[3]
C
3
```

subsets v by supplying the positions of the retained elements.

The command

```
> v[-3]
A B D E
1 2 4 5
```

subsets v by supplying the positions of the unwanted elements.

Chapter 10 R Commands

The theoretical ACF of a stationary ARMA process can be computed by the ARMAacf function. The ar parameter vector, if present, is to be passed into the function via the ar argument. Similarly, the ma parameter vector is passed into the function via the ma argument. The maximum lag may be specified by the lag.max argument. Setting the pacf argument to TRUE computes the theoretical pacf; otherwise the function computes the theoretical acf. Consider as an example the seasonal MA model:

$$Y_t = (1 + 0.5B)(1 + 0.8B^{12})e_t$$

Note that $(1 + 0.5B)(1 + 0.8B^{12}) = (1 + 0.5B + 0.8B^{12} + 0.4B^{13})$ so the ma coefficients are specified by the option ma=c(0.5, rep(0, 10), 0.8, 0.4). Its theoretical ACF is displayed on the left side of Exhibit 10.3, which can be done by the following R commands.

```
plot(y=ARMAacf(ma=c(0.5,rep(0,10),0.8,0.4),
               lag.max=13)[-1],x=1:13,type='h',
               xlab='Lag k',ylab=expression(rho[k]),axes=F,ylim=c(0,0.6))
points(y=ARMAacf(ma=c(0.5,rep(0,10),0.8,0.4),
                  lag.max=13)[-1],x=1:13,pch=20)
abline(h=0)
axis(1,at=1:13,
      labels=c(1,NA,3,NA,5,NA,7,NA,9,NA,11,NA,13))
axis(2)
text(x=7,y=.5,labels=expression(list(theta&=&-0.5,
                                       Theta&=&-0.8)))
```

As the labeling of the figure requires Greek alphabets and subscripts, the label information has to be passed via the expression function. Run the help menu

?plotmath to learn more about how to do mathematical annotations in R.

```
# Exhibit 10.10 on page 237
m1.co2=arima(co2,order=c(0,1,1),
  seasonal=list(order=c(0,1,1),period=12))
```

The argument `seasonal` supplies the information on the seasonal part of the seasonal ARIMA model. It expects a `list` with the seasonal order supplied in the component named `order` and the seasonal period entered via the `period` component, so the command above instructs the `arima` function to fit a seasonal ARIMA $(0,1,1) \times (0,1,1)_{12}$ model to the `co2` series.

```
m1.co2
```

This prints a summary of the fitted seasonal ARIMA model.

Chapter 11 R Commands

```
# Exhibit 11.5 on page 255.
acf(as.vector(diff(diff(window(log(airmiles),
  end=c(2001,8)),12))),lag.max=48)
```

The expression `window(log(airmiles), end=c(2001, 8))` subsets the `log(airmiles)` time series by specifying a new end date of August 2001. The sub-time series is first seasonally differenced with lag 12 and then regularly differenced. The doubly differenced series is then passed to the `acf` function for computing the sample ACF out to 48 lags.

```
# Exhibit 11.6 on page 255.
air.m1=arimax(log(airmiles),order=c(0,1,1),seasonal=
  list(order=c(0,1,1),period=12),
  xtransf=data.frame(I911=1*(seq(airmiles)==69),
  I911=1*(seq(airmiles)==69)),
  transfer=list(c(0,0),c(1,0)),
  xreg=data.frame(Dec96=1*(seq(airmiles)==12),
  Jan97=1*(seq(airmiles)==13),
  Dec02=1*(seq(airmiles)==84)),method='ML')
```

The `arimax` function extends the `arima` function so that it can handle intervention analysis and outliers (both AO and IO) in time series. It is assumed that the intervention affects the mean function of the process, with the deviation from the unperturbed mean function modeled as the sum of the outputs of an ARMA filter of a number of covariates; the deviation is known as the transfer function. The covariates making up the transfer function are passed to the `arimax` function via the `xtransf` argument in the form of a matrix or a `data.frame`. For each such covariate, its contribution to the transfer function takes the form of a dynamic response given by

$$\frac{(a_0 + a_1 B + \dots + a_q B^q)}{(1 - b_1 B - b_2 B^2 - \dots - b_p B^p)} covariate_t$$

The transfer function is the sum of the dynamic responses, in the form of some ARMA filter, of all covariates in the `xtransf` argument. The ARMA order of the filter is

denoted by the vector $c(p, q)$. If $p = q = 0$ (that is, $c(p, q) = c(0, 0)$), the contribution of the covariate is of the form $a_0 covariate_t$. If $c(p, q) = c(1, 0)$, the output becomes

$$\frac{a_0}{(1 - b_1 B)} covariate_t = a_0(covariate_t + b_1 covariate_{t-1} + b_1^2 covariate_{t-2} + \dots)$$

The ARMA orders for the dynamic components of the transfer function are supplied via the `transf` argument as a `list` containing the vectors of ARMA orders in the order of the covariates defined in the `xtransf` argument. Hence, the options:

```
xtransf=data.frame(I911=1*(seq(airmiles)==69),
I911=1*(seq(airmiles)==69)),
transfer=list(c(0,0),c(1,0))
```

instruct the `arimax` function to create two identical covariates called `I911`, which is an indicator variable, say P_t , that equals 1 in September 2001 and 0 otherwise, and the transfer function is the sum of two ARMA filters of the 9/11 indicator variable of orders $c(0,0)$ and $c(1,0)$ respectively. Hence the transfer function equals

$$\omega_0 P_t + \frac{\omega_1}{(1 - \omega_2 B)} P_t$$

This is equivalent to an ARMA(1,1) filter of the form

$$\frac{\{(\omega_0 + \omega_1) - \omega_0 \omega_2 B\}}{(1 - \omega_2 B)} P_t$$

which can be specified by the following options

```
xtransf=data.frame(I911=1*(seq(airmiles)==69)),
transfer=list(c(1,1))
```

Additive outliers (AO) in a time series can be incorporated as indicator variables passed to the `xreg` argument. For example, three potential AOs are included in the model by the following supplied argument:

```
xreg=data.frame(Dec96=1*(seq(airmiles)==12),
Jan97=1*(seq(airmiles)==13),
Dec02=1*(seq(airmiles)==84))
```

Note that the first potential outlier occurs in December 1996. The corresponding indicator variable is labeled as `Dec96` and is computed by the formula `1*(seq(airmiles)==12)`, which results in a vector that equals 0 except its twelfth element, which equals 1, and the vector is of the same length as `airmiles`. Some specifics of this “simple” command follow. The function `seq` creates a vector consisting of the first n positive integers, where n is the length of the vector passed to the `seq` function. The expression `seq(airmiles)==12` creates a vector of the same length as `airmiles`, and its elements are all FALSE except that the twelfth element is TRUE. Then `1*(seq(airmiles)==12)` is an arithmetic expression for which R automatically converts any imbedded Boolean vector (`seq(airmiles)==12`) to a binary vector. Recall that the TRUE values are converted to 1s and the FALSE values to 0s.

Multiplying by 1 does not alter the converted binary vector. Indeed, multiplication is employed to trigger the conversion from the Boolean values to binary values.

For this example, the unperturbed process is assumed to be an IMA(1,1) process, as is evident from the supplied argument `order=c(0, 1, 1)`. In general, a seasonal ARIMA unperturbed process is specified in the same way that it is specified for the `arima` function.

```
air.ml
```

This prints out the fitted intervention model, as displayed below.

```
> air.ml
Call: arimax(x=log(airmiles), order=c(0, 1, 1), seasonal=
  list(order=c(0, 1, 1), period=12), xreg=data.frame(Dec96=
    1*(seq(airmiles)==12), Jan97=1*(seq(airmiles)==13),
    Dec02=1*(seq(airmiles)==84)), method='ML',
    xtransf=data.frame(I911=1*(seq(airmiles)==69), I911=1*
      (seq(airmiles)==69)), transfer=list(c(0, 0), c(1, 0)))

Coefficients:
          mal      smal   Dec96   Jan97   Dec02  I911-MA0  I911.1-AR1  I911.1-MA0
  -0.3825 -0.6499  0.0989 -0.0690  0.0810  -0.0949   0.8139   -0.2715
  s.e.  0.0926  0.1189  0.0228  0.0218  0.0202   0.0462   0.0978   0.0439
sigma^2 estimated as 0.000672: log likelihood=219.99, aic=-423.98
```

Note that the parameter in the transfer-function component defined by the first instance of the indicator variable `I911` is labeled as `I911-MA0`; that is, the MA(0) coefficient. The transfer-function components defined by the second instance of the indicator variable `I911` are labeled as `I911.1-AR1` and `I911.1-MA0`. These are the AR(1) and MA(0) coefficient estimates.

We can also try the equivalent parameterization of specifying an ARMA(1,1) filter on the 9/11 indicator variable.

```
> air.mla=arimax(log(airmiles), order=c(0, 1, 1),
  seasonal=list(order=c(0, 1, 1), period=12),
  xtransf=data.frame(I911=1*(seq(airmiles)==69)),
  transfer=list(c(1, 1)),
  xreg=data.frame(Dec96=1*(seq(airmiles)==12),
  Jan97=1*(seq(airmiles)==13),
  Dec02=1*(seq(airmiles)==84)), method='ML')

> air.mla
Call: arimax(x=log(airmiles), order=c(0, 1, 1), seasonal=
  list(order=c(0, 1, 1), period=12), xreg=data.frame(Dec96=1
  *(seq(airmiles)==12), Jan97=1*(seq(airmiles)==13), Dec02=
  1*(seq(airmiles)==84)), method='ML', xtransf=
  data.frame(I911=1*(seq(airmiles)==69)), transfer=
  list(c(1, 1)))

Coefficients:
          mal      smal   Dec96   Jan97   Dec02  I911-AR1  I911-MA0  I911-MA1
  -0.3601 -0.6130  0.0949 -0.0840  0.0802   0.8094  -0.3660   0.0741
  s.e.  0.0926  0.1261  0.0222  0.0229  0.0194   0.0924   0.0233   0.0424
sigma^2 estimated as 0.000648: log likelihood=221.76, aic=-427.52
```

Note that the parameter estimates of this model are similar to those of the previous model but this model has a better fit, which may happen as the optimization is done numerically.

```
# Exhibit 11.8 on page 256.
Nine11p=1*(seq(airmiles)==69)
```

This defines the 9/11 indicator variable.

```
plot(ts(Nine11p*(-0.0949)+ filter(Nine11p,filter=.8139,
  method='recursive',side=1)*(-0.2715),
  frequency=12,start=1996),type='h',ylab='9/11 Effects')
```

The command

```
Nine11p*(-0.0949)+filter(Nine11p,filter=.8139,
  method='recursive',side=1)*(-0.2715)
```

computes the estimated transfer function. Note that the command

```
filter(Nine11p,filter=.8139,method='recursive',side=1)
```

computes $(1 - 0.8139B)Nineteen11p$. The function `filter` performs an MA or AR filtering on the input sequence passed to it as the first argument. Suppose the input is a vector $x = c(x_1, x_2, \dots, x_n)$. Then the output $y = c(y_1, y_2, \dots, y_n)$ defined by the MA filter

$$y_t = c_0x_t + c_1x_{t-1} + \dots + c_qx_{t-q}$$

can be computed by the command

```
filter(x,filter=c(c0,c1,...,cq),side=1).
```

The argument `side=1` specifies that the MA operator works on current and past values when computing an output value. To compute y_1 , the value of x_0 is needed. Since the latter is not observed, the filter sets it to NA, and hence y_1 is also NA. In this case, y_2 , y_3 , and so forth can be computed. For an AR filtering with the output defined recursively by the equation

$$y_t = x_t + c_1y_{t-1} + \dots + c_py_{t-p}$$

the R command is

```
filter(x,filter=c(c1,c2,...,cp),method='recursive',
  side=1)
```

Note that, unlike the case of the MA filter, the filter vector starts with c_1 and there is no c_0 in the equation. The argument `method='recursive'` signifies an AR type of filtering. For the AR filter, the initial values cannot be set to NA, lest all output values be NA! The default initial values are zeros although other initial values may be specified via the `init` argument.

```
abline(h=0)
```

adds a horizontal line with zero y-intercept.

```
# Exhibit 11.9 on page 259.
set.seed(12345)
y=arima.sim(model=list(ar=.8,ma=.5),n.start=158,n=100)
```

This simulates an ARMA(1,1) series of sample size 100. To remove transient effects of the initial values, a burn-in of size 158 is specified. A large burn-in of the order of hundreds should generally ensure that the simulated process is approximately stationary. The number 158 is chosen for no particular good reason.

```
y[10]
```

This prints out the tenth simulated value.

```
y[10]=10
```

This alters the tenth value to be 10; that is, it becomes an additive outlier, mimicking the effect of a clerical recording mistake, for example!

```
y=ts(y,freq=1,start=1); plot(y,type='o')
acf(y)
pacf(y)
eacf(y)
```

This exploratory analysis suggests an AR(1) model.

```
m1=arima(y,order=c(1,0,0)); m1; detectAO(m1)
```

This detects the presence of any additive outliers (AO) in the fitted AR(1) model. The test requires an estimate of the standard deviation of the error (innovation) term, which by default is estimated by a robust estimation scheme, resulting in a more powerful test. The robust estimation scheme can be switched off by the argument `robust=F`, as illustrated in the command below.

```
detectAO(m1, robust=F)
```

This verifies that a nonrobust procedure is less powerful.

```
detectIO(m1)
```

This detects the presence of any innovative outliers (IO) in the fitted AR(1) model. As an AO is found in the tenth case, it is incorporated as an indicator covariate in the following model.

```
m2=arima(y,order=c(1,0,0),xreg=data.frame(AO=seq(y)==10))
m2
# Exhibit 11.10 on page 260
data(co2)
m1.co2=arima(co2,order=c(0,1,1),seasonal=list
             (order=c(0,1,1),period=12))
m1.co2
detectAO(m1.co2)
detectIO(m1.co2)
```

As an IO is found in the 57th data case, it is incorporated in the model.

```
m4.co2=arimax(co2,order=c(0,1,1),
               seasonal=list(order=c(0,1,1),period=12),io=c(57))
```

The epochs of IOs are passed to the `arimax` function via the `io` argument, which expects a `list` containing the positions of the IOs either as the time index of the IO or as a vector in the form of `c(year,month)` that gives the year and month of the IO for seasonal data; the latter format also works similarly for seasonal data of other types. For

a single IO, it is not necessary to enclose the single vector of index in a list before passing it to the `io` argument.

```
# Exhibit 11.11 on page 262.
set.seed(12345)
X=rnorm(105)
Y=zlag(X,2)+.5*rnorm(105)
```

The command `zlag(X, 2)` computes the second lag of `X`.

```
X=ts(X[-(1:5)], start=1, freq=1)
```

This omits the first five values of `X` and converts the remaining values to form a time series.

```
Y=ts(Y[-(1:5)], start=1, freq=1)
ccf(X,Y,ylab='CCF')
```

This computes the cross-correlation function of `X` and `Y`. The `ylab` argument is supplied in lieu of the default `y`-label of the `ccf` function that is “ACF”.

```
# Exhibit 11.14 on page 264.
data(milk)
data(electricity)
milk.electricity=ts.intersect(milk,log(electricity))
```

The `ts.intersect` function merges several time series into a matrix (panel) of time series over the time frame where each series has data. The object `milk.electricity` is a matrix of two time series, the first column of which is the milk series and the second the log of electricity, over the time period when these two series overlap.

```
plot(milk.electricity,yax.flip=T)
```

The option `yax.flip=T` flips the label for the y-axis for the series alternately so as to make the labeling clearer.

```
# Exhibit 11.15 on page 265.
ccf(milk.electricity[,1],milk.electricity[,2],
  main='milk & electricity',ylab='CCF')
```

The expression `milk.electricity[, 1]` extracts the milk series and `milk.electricity[, 2]` the log electricity series.

The `as.vector` function strips the time series attribute from the time series. This is done to nullify the default way that the `ccf` function plots the cross-correlations. You may want to repeat the command without the `as.vector` function to see the default labels of the lags according to the period of the data.

```
ccf((milk.electricity[,1]),(milk.electricity[,2]),
  main='milk & electricity',ylab='CCF')
```

The bracket operator extracts a submatrix from a matrix, say `M`, in the form of `M[v1, v2]`, where `v1` indicates which rows are kept and `v2` indicates which columns are retained. Consequently, the submatrix `M[v1, v2]` contains all elements of `M` in the intersection of the retained rows and columns. If `v1` (`v2`) is missing, then all rows (columns) are retained. Hence, `M[, 1]` is simply the submatrix consisting of the first column of `M`. However, R adopts the convention that a submatrix with a single row or column is “demoted” to a vector; that is, it loses one dimension. This convention makes

sense in most cases. However, if you do matrix algebra in R, this convention may result in strange error messages! To prevent automatic dimension reduction, use `M[v1, v2, drop=F]`. Instead of specifying which rows or columns are to be retained in the submatrix, you can specify which rows or columns are to be deleted by specifying the negative of their positions. Or `v1` (`v2`) can be specified as a Boolean vector, where the positions to be retained (eliminated) are denoted by `TRUE` (`FALSE`).

```
# Exhibit 11.16 on page 267.
me.dif=ts.intersect(diff(diff(milk,12)),
                     diff(diff(log(electricity),12)))
prewhiten(as.vector(me.dif[,1]),as.vector(me.dif[,2]),
          ylab='CCF')
```

The `prewhiten` function expects two time series input via the `x` and `y` arguments. Both series will be filtered according to an ARIMA model. The ARIMA model can be supplied via the `x.model` argument and should be the output of the `arima` function. If no ARIMA model is supplied, an AR model will be fitted to the `x` series, with the AR order selected by minimizing the AIC. The `prewhiten` function computes and plots the cross-correlation function (CCF) of the residuals of the `x` series and those of the `y` series from the same (supplied or fitted) model.

Chapter 12 R Commands

Below, we show how to implement the Jarque-Bera test for normality in two different ways. First, we show the direct approach.

`skewness(r.cref)`

This computes the skewness of the `r.cref` series.

`kurtosis(r.cref)`

This computes the kurtosis of the data.

`length(r.cref)*skewness(r.cref)^2/6`

The function `length` returns the length of the vector (time series) passed into it, so the expression above computes the first part of the Jarque-Bera statistic.

`length(r.cref)*kurtosis(r.cref)^2/24`

computes the second half of the Jarque-Bera statistic.

`JB=length(r.cref)*(skewness(r.cref)^2/6 +
kurtosis(r.cref)^2/24)`

The object `JB` then contains the Jarque-Bera statistic and the command `JB` prints out the statistic. The command `1-pchisq(JB, df=2)` computes the *p*-value of the Jarque-Bera test for normality. The function `pchisq` computes the cumulative probability of a chi-square distribution being less than or equal to the value in the first argument. The `df` argument of the `pchisq` function specifies the degrees of freedom for the chi-square distribution. Because the *p*-value equals the right tail area, it equals 1 minus the cumulative probability. Besides `pchisq`, other functions associated with the chi-square distribution include `qchisq`, which computes quantiles; `dchisq`, which

computes the probability density; and `rchisq`, which simulates realizations from the chi-square distributions. Use Help in R to learn more about these functions. For other probability distributions, similar functions are available. Associated with the normal distributions are `rnorm`, `pnorm`, `dnorm`, and `qnorm`. Check out the usages of the relevant functions for the binomial (`binom`), Poisson, and other distributions.

```
library(tseries)
```

This loads the `tseries` library, which contains a number of functions needed for the analysis reported in this chapter. Run `library(help=tseries)` for more information about the `tseries` package.

```
jarque.bera.test(r.cref)
```

This carries out the Jarque-Bera test for normality with the time series `r.cref`.

```
# Exhibit 12.9 on page 283.  
McLeod.Li.test(y=r.cref)
```

This performs the McLeod-Li test for presence of ARCH in the daily CREF returns. The first two arguments of the function are `object` and `y`, respectively. For the test with raw data, the time series is supplied to the function via the `y` argument. Then, the function computes the Box-Ljung statistics with the autocorrelations of the squared data to detect for conditional heteroscedascity. The test is carried out with the first m autocorrelations of the squared data, with m ranging from 1 to the maximum lag specified by the `gof.lag` argument. If the `gof.lag` argument is missing, the default is set to $n \log_{10}(n)$ where n is the sample size.

The McLeod-Li test can also be applied to residuals from an ARMA model fitted to the data. For example, the US dollar/Hong Kong dollar exchange rate data was found to admit an AR(1) + outlier model. The need for incorporating ARCH in the model for the exchange rate data can be tested by the command

```
McLeod.Li.test(arima(hkrate, order=c(1,0,0),  
xreg=data.frame(outlier1)))
```

Note that `object` is the first argument so in the above command, the fitted AR(1) + outlier model is passed into the function. The function then computes the test statistics based on the squared residuals from the fitted AR(1) + outlier model. If the `object` argument is supplied explicitly or implicitly, the `y` argument is ignored by the function even if it is supplied. Remember that to apply the test to raw data, the `y` argument must be supplied and the `object` argument suppressed.

```
# Exhibit 12.11 on page 286.  
set.seed(12345678)  
garch01.sim=garch.sim(alpha=c(.01,.9), n=500)
```

The `garch.sim` function simulates a GARCH process, with the ARCH coefficients supplied via the `alpha` argument and the GARCH coefficients via the `beta` argument. The sample size is passed into the function via the `n` argument. In the example above, `alpha=c(.01,.9)` specifies that the constant term is 0.01 and the ARCH(1) coefficient equals 0.9. So, `garch01.sim` saves a realization from an ARCH(1) process.

```
# Exhibit 12.25 on page 300.  
m1=garch(x=r.cref, order=c(1,1))
```

This fits a GARCH(1,1) model with the `r.cref` series. The `garch` function estimates a GARCH model by maximum likelihood. The time series is supplied into the function by the `x` argument and the GARCH order by the `order` argument. The `order` takes the form `c(p, q)` where `p` is the GARCH order and `q` the ARCH order.

```
summary(m1)
```

This summarizes the fitted GARCH(1,1) model. Ignore the Box-Ljung test results reported in the summary, as the generalized portmanteau tests should be used; see the book.

```
# Exhibit 12.29 on page 305.  
gBox(m1,method='squared')
```

The `gBox` function computes the generalized portmanteau test for checking whether or not there is any residual heteroscedasticity in the residuals of a fitted GARCH model. It requires supplying the fitted GARCH model from the `garch` function through the first argument (the `model` argument, the first argument of the function). By default, the tests are carried out with the squared residuals from the fitted GARCH model. To inspect absolute residuals, use the option `method='absolute'`. By default, the test is carried out for the ACF for lags from 1 to, say, `K`, where `K` runs from 1 to 20. The collection of `K`'s can be specified by the `lags` argument. For example, to carry out the test for `K` ranging from 1 to 30, supply the option `lags=1:30`.

```
gBox(m1, lags=20,plot=F,x=r.cref, method='squared')$pvalue
```

prints out the *p*-values of the generalized portmanteau test with the squared residuals and `K = 20`; that is, it tests any residual heteroscedasticity based on the first 20 lags of residual ACF of the squared residuals from the fitted GARCH model. Plotting is switched off by the `plot=F` option. The `gBox` function returns a list, an element of which is named `pvalue` and contains the *p*-values of the test for each `K`. Thus, the command prints out the *p*-value for the test with `K = 20`.

```
# Exhibit 12.30 on page 306.  
acf(abs(residuals(m1)),na.action=na.omit)
```

As the initial residuals from a fitted GARCH model may be missing, it is essential to instruct the ACF to omit all missing values through the argument `na.action=na.omit` (the preferred action when encountering a missing value is to omit it). If this argument is omitted, the `acf` function uses all data and will return missing values if there are any missing data.

Overfitting the GARCH(1,2) model to the CREF returns can be carried out by the following command

```
m2=garch(x=r.cref,order=c(1,2))  
summary(m2,diagnostics=F)
```

The summary is based on the `summary.garch` function in the `tseries` package. Note that the *p*-values of the Ljung-Box test from the summary are invalid; the generalized portmanteau tests should be used instead. Hence, the diagnostics are turned off.

```
AIC(m2)
```

This computes the AIC of the fitted GARCH model `m1`.

```
# Exhibit 12.31 on page 306.
gBox(m1,x=r.cref,method='absolute')
```

This carries out the generalized portmanteau test based on the absolute residuals.

```
shapiro.test(na.omit(residuals(m1)))
```

This computes the Shapiro-Wilk test for normality with the residuals from the fitted model m1. The function `na.omit` strips all missing values from the residuals. Thus, the test is carried out with the nonmissing residuals. Without preprocessing the residuals by the `na.omit` function, the test may return a missing value if some of the residuals are missing!

```
# Exhibit 12.32 on page 307.
plot((fitted(m1) [,1]) ^ 2, type='l',
      ylab='conditional variance', xlab='t')
```

The `fitted` function is a smart function that processes differently depending on the fitted model passed to it as the first argument. If the fitted model is some output from the `garch` function, the default output from the `fitted` function is a two-column matrix whose first column contains the one-step-ahead conditional standard deviations. Hence, their squares are the conditional variances. So `(fitted(m1) [,1]) ^ 2` computes the time series of estimated one-step-ahead conditional variances based on the model `m1`.

Chapter 13 R Commands

```
# Exhibit 13.3 on page 323.
```

The periodogram of a time series can be computed and plotted by the function `periodogram` into which the data are passed as its first argument.

```
sp=periodogram(y); abline(h=0);
axis(1,at=c(0.04167,.14583))
```

The function `periodogram` has several useful arguments. Setting `log='yes'` tells R to plot on a log scale, whereas `log='no'` (the default) says to plot on a linear scale. Other arguments for the `plot` function may be passed into the function to make better graphs. The function `axis` draws an axis with the first argument specifying the side on which the axis is drawn. The sides are labeled from 1 to 4 starting from the bottom in a clockwise direction. The vector of locations of the tick marks can be specified by the `at` argument. The command above instructs R to draw an (additional) axis on the bottom of the figure with tick marks placed at 0.04167 and 0.14583.

```
# Exhibit 13.9 on page 333.
theta=.9 # Reset theta for other MA(1) plots
ARMAspec(model=list(ma=-theta))
```

The function `ARMAspec` calculates and plots the theoretical spectral density function of the ARMA model supplied to the function as the first argument. Recall that R uses the plus convention in the MA specification, so the minus sign is added to theta. The format of the model is the same as that for the `arima` function.

Chapter 14 R Commands

```
# Exhibit 14.2 on page 353.
```

The `spec` function can estimate the spectral density function by locally averaging the periodogram via some suitable kernel function. The function `spec` has several useful arguments. Setting `log='yes'` tells R to plot on a log scale whereas `log='no'` says to plot on a linear scale. Data may be detrended (fitting a linear time trend) by setting `detrend=T`, and tapering may be enforced by setting `taper` to some fraction between 0 and 0.5. The default options are: `taper=0` and `detrend=F`.

```
k=kernel('daniell',m=15)
```

Here, the object `k` contains the Daniell kernel function with halfwidth 15. Use Help in R to learn more about the `kernel` function.

```
sp=spec(y,kernel=k,log='no',sub='',
         xlab='Frequency',ylab='Smoothed Sample Spectral Density')
```

Specifying the kernel to be the Daniell kernel function instructs R to compute and plot the spectral density estimate, where the estimate at a certain frequency is obtained by averaging the current (raw) periodogram value, the neighboring 15 periodogram values on its left, and another 15 periodogram values on its right. More or less local averaging can be specified through the `m` argument in the `kernel` function.

```
lines(sp$freq,ARMAAspec(model=list(ar=phi),freq=sp$freq,
                         plot=F)$spec,lty='dotted')
```

This adds the theoretical spectral density function.

```
# Exhibits 14.11 and 14.12, page 364.
# Spectral analysis of simulated series
set.seed(271435)
n=100
phi1=1.5; phi2=-.75 # Reset parameter values to obtain
# Exhibits 14.13 & 14.14
y=arima.sim(model=list(ar=c(phi1,phi2)),n=n)
```

This simulates an AR(2) time series of length 100.

```
sp1=spec(y,spans=3,sub='',lty='dotted', xlab='Frequency',
          ylab='Log(Estimated Spectral Density)')
```

This estimates the special density function using the modified Daniell kernel (the default kernel when the `kernel` argument is missing and the `spans` argument is supplied). The `spans` argument supplies the width of the kernel function; that is, it is twice the `m` argument in the `kernel` function plus 1. Here, `spans=3` specifies local averaging of three consecutive periodogram values. Note that local averaging may be repeated by passing a vector as the value of `spans`. For example, setting `spans=c(3,5)` performs local averaging twice. The estimated function obtained by local averaging with `spans=3` is then averaged again locally with `spans=5`. Repeated averaging with a modified Daniell (rectangular) kernel is similar to averaging using a bell-shaped kernel due to the Central Limit effect.

```
sp2=spec(y, spans=9, plot=F)
```

This computes the spectrum estimate using a wider window encompassing nine periodogram values without plotting via the `plot=F` argument. The output of the `spec` function is saved into an object named `sp2`.

```
sp3=spec(y, spans=15, plot=F)
```

This uses an even wider window. How many periodogram values are included in each local averaging?

```
lines(sp2$freq, sp2$spec, lty='dashed')
```

This plots the smoother spectrum estimate (`spans=9`) as a dashed line.

```
lines(sp3$freq, sp3$spec, lty='dotdash')
```

This plots the smoothest spectrum estimate (`spans=15`) as a dotdash line.

```
f=seq(0.001,.5,by=.001)
```

This creates an arithmetic sequence starting from 0.001 and ending at 0.5, with increments 0.001, which is then saved into the object `f`.

```
lines(f,ARMAspec(model=list(ar=c(phi1,phi2)), freq=f,
plot=F)$spec, lty='solid')
```

This plots the theoretical spectral density function for the specified ARMA model as connected line segments on top of the estimated spectral density plot.

```
# Exhibit 14.12 on page 365.
```

```
sp4=spec(y,method='ar',lty='dotted', xlab='Frequency',
ylab='Log(Estimated AR Spectral Density)')
```

This estimates the spectral density function using the theoretical spectral density function of an AR model fitted to the data by minimizing the AIC.

```
f=seq(0.001,.5,by=.001)
```

```
lines(f,ARMAspec(model=list(ar=c(phi1,phi2)),
freq=f,plot=F)$spec, lty='solid')
```

This plots the theoretical spectral density function.

```
sp4$method
```

This displays the order of the AR model selected.

Chapter 15 R Commands

```
# Exhibit 15.1 on page 386.
```

```
set.seed(2534567)
```

```
par(mfrow=c(3,2))
```

```
y=arima.sim(n=61,model=list(ar=c(1.6,-0.94),ma=-0.64))
```

This simulates an ARMA(2,1) series of sample size 61.

```
lagplot(y)
```

This plots the lagged regression plots, where the time series is plotted against its lags and a smooth curve is superimposed on each scatter diagram. The smooth curves are obtained by local linear fits to the data. By increasing the value specified in the `nn` argu-

ment (default `nn=0.7`), the local fitting scheme uses more local data, resulting in a smoother fit that is likely to be more biased but less variable due to more smoothing. On the contrary, decreasing the value in the `nn` argument leads to a rougher fit that is less biased but more variable due to less smoothing. The smooth curve in the scatter diagram of the time series response versus its lag j estimates the conditional mean response given its lag j as a function of the value of the lag j of the response. By default, `lagplot` plots the lagged regression plot for lags 1 to 6. More lags can be computed via the `lag.max` argument. For instance, `lag.max=12` computes the lagged regression plots for lags 1 through 12. Note that the `lagplot` function requires the installation of the `locfit` package of R.

```
# Exhibit 15.2 on page 387.  
data(veilleux)
```

The dataset `veilleux` is a matrix consisting of two time series. Its first column is the series of *Didinium* abundance and the second column the series of *Paramecium* abundance, each counted every 12 hours. The basic time unit is days, so these are series of frequency 2, as they are sampled twice per day.

```
predator=veilleux[,1]
```

This defines the predator series as the abundance series of *Didinium*.

```
plot(log(predator),lty=2,type='b',xlab='Day',  
     ylab='Log(predator)')
```

This plots the entire log-transformed predator series as a dashed line.

```
predator.eq=window(predator,start=c(7,1))
```

This subsets the “stationary” part of the predator series that appears to begin on the seventh day of the experiment. Subsequent analyses of the predator series reported in the text were done with this log-transformed stationary subseries.

```
lines(log(predator.eq))
```

This draws the stationary part as a solid line.

```
index1=zlag(log(predator.eq),3)<=4.661
```

The command `zlag(log(predator.eq),3)` returns the lag 3 of the (log-transformed) predator series. The expression `zlag(log(predator.eq),3)<=4.661` computes a Boolean vector whose elements are TRUE if and only if their corresponding element of the lag 3 of the predator series is less than or equal to 4.661. The Boolean vector is saved in an object named `index1`. Other comparison operators, including `>=`, `>`, `<`, and `==`, can be used to compare the vectors on the two sides of the comparison operator. In the example above, the left-hand side of `<=` is a vector, but its right-hand side is a scalar! The discrepancy is resolved by the recycling rule, that R replicates the shorter vector repeatedly to match its longer part. Note that the equality operator is denoted by the double equal sign `==`, as the single equal sign represents the assignment operator!

```
points(y=log(predator.eq)[index1],(time(predator.eq))
       [index1],pch=19)
```

This draws as solid circles (`pch=19`) those data points whose lag 3 of the predator abundance is less than or equal to 4.661. Run the command `?points` to learn other styles for plotting data points.

```
# Tests for nonlinearity, page 390.
Keenan.test(sqrt(spots))
```

This carries out Keenan's test for linearity. The working order of the AR process under the null hypothesis of linearity can be supplied via the `order` argument. For example, `order=2` sets the working AR order to 2. If the `order` argument is missing, the order is automatically determined by minimizing the AIC via the `ar` function. The `ar` function by default estimates the models by solving the Yule-Walker equations. But other estimation methods may be used by including the `method` argument when calling the `Keenan.test` function; for example, `method='mle'` specifies using maximum likelihood in the `ar` function.

```
Tsay.test(sqrt(spots)), page 390.
```

This implements Tsay's test for linearity; see Tsay (1986). The design of the `Tsay.test` function and its arguments are similar to those of the `Keenan.test` function.

```
# Exhibit 15.6 on page 400.
y=qar.sim(n=100,const=0.0,phi0=3.97,
           phi1=-3.97,sigma=0,init=.377)
```

The function `qar.sim` simulates a time series realization from a first-order quadratic AR model where `phi0` is the coefficient of the lag 1 and `phi1` is that of the square of lag 1. The default intercept is zero, otherwise it can be set by the `const` argument. The innovation standard deviation is passed into the function via the `sigma` argument. Here, `sigma=1` sets the standard deviation to be 1. The argument `n=15` sets the sample size to 15. Finally, the argument `init=.377` sets the initial value to be 0.377. The default initial value is 0.

```
plot(x=1:100,y=y,type='l',ylab=expression(Y[t]),xlab='t')
```

The output of the `qar.sim` function is a vector. To draw the time sequence plot, both the *x*-variable and the *y*-variable have to be specified.

```
# Exhibit 15.8 on page 411.
set.seed(1234579)
y=tar.sim(n=100,Phi1=c(0,0.5),Phi2=c(0,-1.8),p=1,d=1,
           sigma1=1,thd=-1,sigma2=2)$y
```

The function `tar.sim` simulates time series realizations from a two-regime TAR model. The order of the model is specified by the `p` argument, so `p=1` specifies a first-order model. The delay is passed into the function by the `d` argument, so `d = 1` specifies the delay to be 1. The AR coefficient vector for the lower (upper) regime, with the intercept being the first component, is supplied via the `Phi1` (`Phi2`) argument. The `thd=-1` argument imposes the threshold parameter of -1. The innovation standard deviations for the lower and upper regimens are specified via the `sigma1` and `sigma2`

arguments, respectively. The simulated TAR model in the example is conditionally heteroscedastic, as the innovation standard deviation for the upper regime is twice that for the lower regime. The sample size is set to 100 by the `n=100` argument.

The likelihood ratio test for threshold nonlinearity, assuming normally distributed innovations, can be carried out by the `tlrt` function, with which the data enter into the function as the first argument. Other required information includes the `order` and `delay` arguments. Also, the threshold parameter must be searched over a finite interval from the a times 100 percentile to the b times 100 percentile of the data. Often, data have to be transformed before testing for nonlinearity, which can be specified by supplying the transformed data or supplying the raw data with the `transform` argument set to one of the available options: '`no`' (means no transformation, the default), '`log`', '`log10`', or '`sqrt`'. For example, the following command does the likelihood ratio test of the null hypothesis that the square root transformation of relative sunspot data is an AR(5) process versus the alternative that it follows a threshold model with delay 1, order 5, and with the threshold parameter searched from the first to the third quartile of the (transformed) data.

```
tlrt(sqrt(spots), p=5, d=1, a=0.25, b=0.75)
```

The `tlrt` function outputs a list containing the test statistic and its p -value. In practice, the true delay of the threshold model is unknown, although it is likely to be between 1 and the order of the model. (The delay may be specified to some value greater than the order if this is deemed appropriate.) The command above can be replicated a number of times for each possible delay value. A more elegant way is to use a `for` loop as follows.

```
# Tests for threshold nonlinearity, page 400.
pvaluem=NULL
```

This defines an empty object named `pvaluem`.

```
for (d in 1:5)
  {res=tlrt(sqrt(spots), p=5, d=d, a=0.25, b=0.75); pvaluem=
  cbind(pvaluem, c(d, res$test.statistic, res$p.value))}
```

The statements within the curly brackets are repeated for each value the variable `d` takes sequentially from the vector `1 : 5`, which contains the first five positive integers. Thus, `d` is first set to 1, and the likelihood ratio test for threshold nonlinearity is carried out, with its output stored in an object named `res`. The command `c(d, res$test.statistic, res$p.value)` creates a vector containing the value 1, the likelihood ratio test statistic, and its p -value. The vector so created is then augmented to the right-hand side of `pvaluem` to form a matrix. So, after the first loop, `pvaluem` is a matrix consisting of the test results for `d=1`. Then the loop sets `d` to the second value, namely 2; carries out the threshold likelihood ratio test for `d=2`; augments the test results for `d=2` to the right-hand side of `pvaluem`; and so forth until the loop exhausts all possible values for `d` and `n` and then R exits from the loop.

```
rownames(pvaluem)=c('d','test statistic','p-value')
```

This labels the rows of the `pvaluem` matrix, with the first row labeled as “`d`”, the second “`test statistic`”, and the third row “`p-value`”.

```
round(pvaluem,3)
```

This prints out the matrix (table) of test results, with the numbers rounded to three decimal places. Note that the computational efficiency of the R code above can be improved by declaring pvaluem as a matrix with appropriate dimension (for example, pvaluem=matrix('NA',nrow=3,ncol=5)) in which the test results are saved.

```
# Exhibit 15.12 on page 405.
predator.tar.1=tar(y=log(predator.eq),p1=4,p2=4,d=3,a=.1,
b=.9,print=T)
```

This fits a threshold model with the (log-transformed) predator.eq series with maximum AR order to be 4 for both lower and upper regimes, d=3, and the threshold parameter searched from the tenth to the ninetieth percentiles. The fitted model is printed out if the print argument is set to T. By default, the function uses the MAIC (minimum AIC) method for estimation, with the AR orders estimated as well. Another method of estimation is conditional least squares, which can be specified by the method='CLS', as illustrated in the next command.

In the command below, we repeat the estimation but using the CLS method. Note that the CLS method does not estimate the AR orders of the two regimes. Instead, the AR orders are set as the maximum orders specified through the p1 and p2 arguments! That is why the values of p1 and p2 are set differently from the previous command and in fact set as the orders estimated from the model using the MAIC method.

```
tar(y=log(predator.eq),p1=1,p2=4,d=3,a=.1,b=.9,print=T,
method='CLS')
```

```
# Exhibit 15.13 on page 408.
tar.skeleton(predator.tar.1)
```

This computes the skeleton of a TAR model supplied as the first argument, with a default sample size of 500 values, a burn-in of 500 values, and plots the time sequence plot of the last 50 values of the skeleton. The TAR model is usually the output of that of the object argument of the tar function. Alternatively, the model parameters can be specified in a format similar to the tar.sim function. The function also prints a summary statement on the long-run behavior of the skeleton.

```
# Exhibit 15.14 on page 408.
set.seed(356813)
plot(y=tar.sim(n=57,object=predator.tar.1)$y,x=1:57,
ylab=expression(Y[t]),xlab=expression(t),type='o')
```

This plots a simulated time series from the fitted TAR(2;1,4) model to the predator series. The fitted model is supplied via the object argument.

```
# Exhibit 15.20 on page 414.
tsdiag(predator.tar.1,gof.lag=20)
```

This carries out several model diagnostics on the fitted TAR(2;1,4) model to the predator series. The function plots a time sequence plot of the standardized residuals, the residual ACF, and the *p*-value plots of the generalized portmanteau tests. The argument gof.lag=20 specifies that the last two plots use a maximum lag of 20.

```
# Exhibit 15.21 on page 415.
qqnorm(predator.tar.1$std.res)
```

This plots the quantile-quantile normal score plot for the standardized residuals from the TAR(2;1,4) model fitted to the predator series.

```
qqline(predator.tar.1$std.res)
```

adds the reference line on the Q-Q plot.

```
# Exhibit 15.22 on page 417.
```

```
set.seed(2357125)
pred.predator=predict(predator.tar.1,n.ahead=60,
n.sim=1000)
```

This simulates a time series from the conditional distribution of the future values given the data and a threshold model (usually the output of the tar function, here being `predator.tar.1`), with a forecast horizon of a maximum sixty-step-ahead predictions. The point predictors and their 95% prediction limits are computed by simulation. The simulation size is specified as `n.sim=1000`. The output of the `predict` function is a list that contains the prediction means as a vector in the component (element) named `fit` and the lower and upper prediction limits as a matrix in the `pred.interval` component. The function `predict` is a smart function and recognizes that the first argument is a TAR model, on the basis of which it computes the prediction. To learn more about the `predict` function for TAR models, run `?predict.TAR`. The extension TAR signifies the particular predict function for processing prediction based on a TAR model.

```
yy=ts(c(log(predator.eq),pred.predator$fit),frequency=2,
      start=start(predator.eq))
```

This augments the point prediction values to the data.

```
plot(yy,type='n',
      ylim=range(c(yy,pred.predator$pred.interval)),
      ylab='Log Prey', xlab=expression(t))
```

This sets up a plot of the data and the predicted future values without actual plotting (`type='n'`). We anticipate superimposing the prediction intervals, so the range of the y-axis is specified through the `ylim` argument to the vector containing the minimum and maximum of the combined vector of the observed + predicted values (`yy`) and the prediction limits (`pred.predator$pred.interval`), computed via the `range` function.

```
lines(log(predator.eq))
```

This draws the data as a solid line.

```
lines(window(yy, start=end(predator.eq)+c(0,1)),lty=2)
```

This adds the curve of the predicted values as a dashed line.

```
lines(ts(pred.predator$pred.interval[2,],
      start=end(predator.eq)+c(0,1),freq=2),lty=2)
```

This adds the upper prediction limits.

```
lines(ts(pred.predator$pred.interval[1,],
         start=end(predator.eq)+c(0,1),freq=2),lty=2)
```

This adds the lower prediction limits.

```
# Exhibit 15.24 on page 419.
qqnorm(pred.predator$pred.matrix[,3])
```

The output of the predict function is a list that contains another component, named pred.matrix, which is a matrix containing all simulated future values, with the first column consisting of the simulated one-step-ahead values, the second column those of the two-steps-ahead values, and so forth.

```
qqnorm(pred.predator$pred.matrix[,3])
```

This extracts all 1000 simulated three-steps-ahead values, which are then passed into the qqnorm function to make the Q-Q normal score plot for these data.

```
qqline(pred.predator$pred.matrix[,6])
```

This adds the reference straight line for checking the normality of the three-steps-ahead conditional distribution.

Finally, here is a listing and brief description of all the new or enhanced functions that are contained in the TSA package.

New or Enhanced Functions in the TSA Library

Function	Description
acf	Computes and plots the sample autocorrelation function starting with lag 1.
arima	This command has been amended to compute the AIC according to our definition.
arima.boot	Bootstraps time series according to a fitted ARMA(p, d, q) model.
arimax	Extends the arima function, allowing the incorporation of transfer functions and innovative and additive outliers.
ARMAspec	Computes and plots the theoretical spectrum of an ARMA model.
armasubsets	Finds “best subset” ARMA models.
BoxCox.ar	Finds a power transformation so that the transformed time series is approximately an AR process with normal error terms.
detectAO	Detects additive outliers in time series.
detectIO	Detects innovative outliers in time series.
eacf	Computes and displays the extended autocorrelation function of a time series.
garch.sim	Simulates a GARCH process.
gBox	Performs a goodness-of-fit test for fitted GARCH models.
harmonic	Creates a matrix of the first m pairs of harmonic functions for fitting a harmonic trend (cosine-sine trend, Fourier regression) model with a time series response.

New or Enhanced Functions in the TSA Library (Continued)

Function	Description
Keenan.test	Carries out Keenan's test for nonlinearity against the null hypothesis that the time series follows some AR process.
kurtosis	Calculates the (excess) coefficient of kurtosis.
lagplot	Computes and plots nonparametric regression functions of a time series against its various lags.
periodogram	Computes the periodogram of a time series.
LB.test	Computes the Ljung-Box or Box-Pierce tests checking whether or not the residuals from an ARIMA model appear to be white noise.
McLeod.Li.test	Perform the McLeod-Li test for conditional heteroscedascity (ARCH).
plot.Arima	Plots a time series and its predictions (forecasts) with 95% prediction bounds based on a fitted ARIMA model.
predict.TAR	Calculates predictions based on a fitted TAR model. The errors are assumed to be normally distributed and the predictive distributions are approximated by simulation.
prewhiten	Bivariate time series are prewhitened according to an AR model fitted to the x -component of the bivariate series. Alternatively, if an ARIMA model is provided, it is used to prewhiten both series. The CCF of the prewhitened bivariate series is then computed and plotted.
qar.sim	Simulates a first-order quadratic AR model with normally distributed white noise error terms.
rstandard.Arima	Computes internally standardized residuals from a fitted ARIMA model.
runs	Tests the independence of a sequence of values by checking whether there are too many or too few runs above (or below) the median.
season	Extracts season information from a time series and creates a vector of the season information. For example, for monthly data, the function outputs a vector containing the months of the data.
skewness	Calculates the skewness coefficient of a dataset.
spec	Allows the user to invoke either the spec.pgram function or the spec.ar function in the stats package. The seasonal attribute of the data, if it exists, is suppressed for our preferred way of presenting the output. Alters defaults to demean=T, detrend=F, taper=0, and permits plotting of confidence interval bands.
summary.armasubsets	Summary method for class armasubsets, that is useful for ARMA subset selection.
tar	Estimates a two-regime TAR model.

New or Enhanced Functions in the TSA Library (Continued)	
Function	Description
<code>tar.sim</code>	Simulates a two-regime TAR model.
<code>tar.skeleton</code>	Obtains the skeleton of a TAR model by suppressing the noise term in the TAR model.
<code>tlrt</code>	Carries out the likelihood ratio test for threshold nonlinearity, with the null hypothesis being a normal AR process and the alternative hypothesis being a TAR model with homogeneous, normally distributed errors.
<code>Tsay.test</code>	Carries out Tsay's test for quadratic nonlinearity in a time series.
<code>tsdiag.Arima</code>	Modifies the <code>tsdiag</code> function of the <code>stats</code> package suppressing initial residuals and displaying Bonferroni bounds. It also checks the condition for the validity of the chi-square asymptotics for the portmanteau tests.
<code>tsdiag.TAR</code>	Displays the time series plot and the sample ACF of the standardized residuals. Also, portmanteau tests for detecting autocorrelations in the standardized residuals are computed and displayed.
<code>zlag</code>	Computes the lag of a vector, with missing elements replaced by NA.

DATASET INFORMATION

Filename/ Variable(s)	Description and Source	Page(s)
airmiles	Monthly U.S. airline passenger-miles: 01/1996–05/2005. Source: www.bts.gov/xml/air_traffic/src/index.xml#MonthlySystem	249
airpass	Monthly total international airline passengers from 01/1960–12/1971. Source: Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. <i>Time Series Analysis: Forecasting and Control</i> , second edition, Prentice-Hall, Engelwood Cliffs, NJ, 1994.	104
beersales	Monthly U.S. beer sales (in millions of barrels), 01/1975–12/1990. Source: Frees, E. W., <i>Data Analysis Using Regression Models</i> , Prentice-Hall, Engelwood Cliffs, NJ, 1996.	51
bluebird: (log.sales & price)	Weekly unit sales of Bluebird standard potato chips (New Zealand) and their price for 104 weeks. From the website of Dr. Andrew Balemi. Source: www.stat.auckland.ac.nz/~balemi/Assn3.xls	267
bluebirdlite: (log.sales & price)	Weekly unit sales of Bluebird Lite potato chips (New Zealand) and their price for 104 weeks. From the website of Dr. Andrew Balemi. Source: www.stat.auckland.ac.nz/~balemi/Assn3.xls	276
boardings: (log.boarding s & log.price)	Monthly public transit boardings (mostly buses and light rail), Denver, Colorado region, 08/2000–03/2006. Source: Personal communication from Lee Cryer, Project Manager, Regional Transportation District, Denver, Colorado. Denver gasoline prices were obtained from the Energy Information Administration, U.S. Department of Energy, Washington, D.C., at www.eia.doe.gov	248, 271, 273
co2	Monthly carbon dioxide levels in northern Canada, 01/1994–12/2004. Source: http://cdiac.ornl.gov/ftp/trends/co2/altso.co2	234, 234
color	Color properties from 35 consecutive batches of an industrial chemical process. Source: Cryer, J. D. and Ryan, T. P., “The estimation of sigma for an X chart”, <i>Journal of Quality Technology</i> , 22, No. 3, 187–192.	3, 134, 147, 165, 176, 194
CREF	Daily values of one unit of the CREF (College Retirement Equity Fund) Stock fund, 08/26/04–08/15/06. Source: www.tiaa-cref.org/performance/retirement/data/index.html	278
cref.bond	Daily values of one unit of the CREF (College Retirement Equity Fund) Bond fund, 08/26/04–08/15/06. Source: www.tiaa-cref.org/performance/retirement/data/index.html	316
days	Accounts receivable data. Number of days until a distributor of Winegard Company products pays their account. Source: Personal communication from Mark Selergren, Vice President, Winegard, Inc., Burlington, Iowa.	147, 174, 217, 276

Filename/ Variable(s)	Description and Source (Continued)	Page(s)
deere1	82 consecutive values for the amount of deviation (in 0.000025 inch units) from a specified target value that an industrial machining process at Deere & Co. produced under certain specified operating conditions. Source: Personal communication from William F. Fulkerson, Deere & Co. Technical Center, Moline, Illinois.	146, 275
deere2	102 consecutive values for the amount of deviation (in 0.0000025 inch units) from a specified target value that another industrial machining process produced at Deere & Co. Source: Personal communication from William F. Fulkerson, Deere & Co. Technical Center, Moline, Illinois.	146
deere3	57 consecutive values from a complex machine tool at Deere & Co. The values given are deviations from a target value in units of ten millionths of an inch. The process employs a control mechanism that resets some of the parameters of the machine tool depending on the magnitude of deviation from target of the last item produced. Source: Personal communication from William F. Fulkerson, Deere & Co. Technical Center, Moline, Illinois.	147, 174, 190, 217
eeg	An electroencephalogram (EEG) is a noninvasive test used to detect and record the electrical activity generated in the brain. These data were measured at a frequency of 256 per second and came from a patient suffering a seizure. This is a portion of a series on the website of Professor Richard Smith, University of North Carolina. His source: Professors Mike West and Andrew Krystal, Duke University. Source: http://www.stat.unc.edu/faculty/rs/s133/Data/datadoc.html	380
electricity	Monthly U.S. electricity generation (in millions of kilowatt hours) of all types: coal, natural gas, nuclear, petroleum, and wind, 01/1973–12/2005. Source: www.eia.doe.gov/emeu/mer/elect.html	99, 214, 247, 264, 380
euph	A digitized sound file of about 0.4 seconds of a Bb just below middle C played on a euphonium by one of the authors (JDC), a member of the group <i>Tempered Brass</i> .	374
flow	Flow data (in cubic feet per second) for the Iowa River measured at Wapello, Iowa, for the period 09/1958–08/2006. Source: http://waterdata.usgs.gov/ia/nwis/sw	372, 381
gold	Daily price of gold (in U.S. dollars per troy ounce), 01/04/2005–12/30/2005. Source: www.lbma.org.uk/2005dailygold.htm	105
google	Daily returns of Google stock from 08/20/04 to 09/13/06. Source: http://finance.yahoo.com/q/hp?s=GOOG	317

Filename/ Variable(s)	Description and Source (Continued)	Page(s)
hare	Annual Canadian hare abundance, 1905–1935. Source: Stenseth, N. C., Falck, W., Bjørnstad, O. N. and Krebs, C. J. (1997) “Population regulation in snowshoe hare and Canadian lynx: Asymmetric food web configurations between hare and lynx.” <i>Proceedings of the National Academy of Sciences, USA</i> , 94 , 5147–5152.	4, 136, 152, 176, 206
hours	Monthly average hours worked per week in the U.S. manufacturing sector for 07/1982–06/1987. Source: Cryer, J. D. <i>Time Series Analysis</i> , Duxbury Press, Boston, 1986.	51
JJ	Quarterly earnings per share for 1960Q1–1980Q4 of the U.S. company, Johnson & Johnson, Inc. From the web site of David Stoffer. Source: www.stat.pitt.edu/stoffer/tsa2/	105, 248
lrain	Annual rainfall totals for Los Angeles, California, 1878–1992. Source: Personal communication from Professor Donald Bentley, Pomona College, Claremont, California. For more data see www.wrh.noaa.gov/lox/climate/cvc.php	1, 49, 105, 133, 379
milk	Monthly U.S. milk production from 01/1994 to 12/2005. Source: National Agricultural Statistics Service: usda.mannlib.cornell.edu/MannUsda/viewDocumentInfo.do?documentID=1103	264, 374, 374
oil.price	Monthly spot price for crude oil, Cushing, OK (in U.S. dollars per barrel), 01/1986–01/2006. U.S. Energy Information Administration. Source: tonto.eia.doe.gov/dnav/pet/hist/rwtcM.htm	87, 125, 153, 177, 276, 317
oilfilters	Monthly wholesale specialty oil filter sales, Deere & Co., 07/1983–06/1987. Source: Personal communication from William F. Fulkerson, Deere & Co. Technical Center, Moline, Illinois.	6
prescrip	Monthly U.S. average prescription costs for the months 08/1986 – 03/1992. Source: Frees, E. W., <i>Data Analysis Using Regression Models</i> , Prentice-Hall, Engelwood Cliffs, NJ, 1996.	52
retail	Monthly total UK (United Kingdom) retail sales (non-food stores in billions of pounds), 01/1983–12/1987. Source: www.statistics.gov.uk/statbase/TSDdownload1.asp	52
robot	Final position in the “x” direction of an industrial robot put through a series of planned exercises many times. Source: Personal communication from William F. Fulkerson, Deere & Co. Technical Center, Moline, Illinois.	147, 174, 190, 217, 370
SP	Quarterly S&P Composite Index, 1936Q1–1977Q4, Source: Frees, E. W., <i>Data Analysis Using Regression Models</i> , Prentice-Hall, Engelwood Cliffs, NJ, 1996.	104

Filename/ Variable(s)	Description and Source (Continued)	Page(s)
spots	Annual American (relative) sunspot numbers collected from 1945 to 2005. The annual (relative) sunspot number is a weighted average of solar activities measured from a network of observatories. Source: www.ngdc.noaa.gov/stp/SOLAR/ftpunspotnumber.html#american	392
spots1	Annual international sunspot numbers, NOAA National Geophysical Data Center. Source: ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/YEARLY.PLT	379
star	Brightness of a variable star at midnight on 600 successive nights. Source: www.statsci.org/data/general/star.html	325
tbone	A digitized sound file of about 0.4 seconds of a Bb just below middle C played on a tenor trombone by Chuck Kreeb, a member of <i>Tempered Brass</i> and a friend of one of the authors.	374
tempdub	Monthly average temperatures in Dubuque, Iowa, 1/1964–12/1975. Source: http://mesonet.agron.iastate.edu/climodat/index.phtml?station=ia2364&report=16	6, 213, 379
tuba	A digitized sound file of about 0.4 seconds of a Bb an octave and one whole step below middle C played on a BBb tuba by Linda Fisher, a member of <i>Tempered Brass</i> and a friend of one of the authors.	381
units	Annual sales of certain large equipment, 1983–2005. (Proprietary sales data from a large international company.)	276
usd.hkd	Daily exchange rates of U.S. dollar to Hong Kong dollar, 01/2005–03/2006. A data frame with 431 observations on the following six variables. r: daily returns of USD/HKD exchange rates v: estimated conditional variances based on an AR(1)+GARCH(3,1) hkrate: daily USD/HKD exchange rates outlier1: dummy variable of day 203, corresponding to July 22, 2005 outlier2: dummy variable of day 290, another possible outlier day: calendar day Source: www.oanda.com/convert/fxhistory	310
veilleux: Day, Didinium, Paramecium	A bivariate time series from an experiment studying prey-predator dynamics. The first time series consists of the number of prey individuals (<i>Didinium natsutum</i>) per ml measured every 12 hours over a period of 35 days. The second time series consists of the corresponding number of predators (<i>Paramecium aurelia</i>) per ml. Source: Veilleux, B. G. (1976) “The analysis of a predatory interaction between <i>Didinium</i> and <i>Paramecium</i> .” MSc thesis, University of Alberta, Canada. See also www.journals.royalsoc.ac.uk/content/lekv0yqp2ecpabvd/archive1.pdf	386

Filename/ Variable(s)	Description and Source (Continued)	Page(s)
wages	Monthly average hourly wages in the U.S. apparel industry: 07/1981–06/1987. Source: Cryer, J. D. <i>Time Series Analysis</i> , Duxbury Press, Boston, 1986.	51
winnebago	Monthly unit sales of recreational vehicles from Winnebago, Inc. from 11/1966 to 02/1972. Source: Roberts, H. V., <i>Data Analysis for Managers with Minitab</i> , second edition, The Scientific Press, Red- wood City, CA, 1991.	51, 104

BIBLIOGRAPHY

- Abraham, B. and Ledolter, J. (1983). *Statistical Methods for Forecasting*. New York: John Wiley & Sons.
- Akaike, H. (1973). "Maximum likelihood identification of Gaussian auto-regressive moving-average models." *Biometrika*, **60**, 255–266.
- Akaike, H. (1974). "A new look at the statistical model identification." *IEEE Transactions on Automatic Control*, **19**, 716–723.
- Andersen, T. G., Bollerslev, T., Christoffersen, P. F., and Diebold, F. X. (2006). "Volatility Forecasting." To appear in *Handbook of Economic Forecasting*, edited by Graham Elliott, Clive W. J. Granger, and Allan Timmermann, Amsterdam: North-Holland.
- Anderson, T. W. (1971). *The Statistical Analysis of Time Series*. New York: John Wiley & Sons.
- Banerjee, A., Dolado, J. J., Galbraith, J. W. and Hendry, D. F. (1993). *Cointegration, Error Correction, and the Econometric Analysis of Non-Stationary Data*. Oxford: Oxford University Press.
- Bartlett, M. S. (1946). "On the theoretical specification of sampling properties of auto-correlated time series." *Journal of the Royal Statistical Society B*, **8**, 27–41.
- Becuinj, M., Gourieroucx, S., and Monfort, A. (1980). "Identification of a mixed autoregressive-moving average process: The corner method." In *Time Series*, edited by O. D. Anderson, 423–436. Amsterdam: North-Holland.
- Bloomfield, P. (2000). *Fourier Analysis of Time Series: An Introduction*, 2nd ed. New York: John Wiley & Sons.
- Bollerslev, T. (1986). "Generalized autoregressive conditional heteroskedasticity." *Journal of Econometrics*, **31**, 307–327.
- Box, G. E. P. and Cox, D. R. (1964). "An analysis of transformations." *Journal of the Royal Statistical Society B*, **26**, 211–243.
- Box, G. E. P. and Pierce, D. A. (1970). "Distribution of residual correlations in autoregressive-integrated moving average time series models." *Journal of the American Statistical Association*, **65**, 1509–1526.
- Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis, Forecasting and Control*. San Francisco: Holden-Day.
- Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis, Forecasting and Control*, 2nd ed. New York: Prentice-Hall.

- Box, G. E. P. and Tiao, G. (1975). "Intervention analysis with applications to economic and environmental problems." *Journal of the American Statistical Association*, **70**, 70–79.
- Brillinger, D. R. (2001). *Time Series: Data Analysis and Theory*. Philadelphia, SIAM.
- Brock, W. A., Deckert, W. D., and Seheinkman, J. A. (1996). "A test for independence based on the correlation dimension." *Econometric Reviews*, **15**, 197–235.
- Brockwell, P. J. and Davis, R. A. (1991). *Time Series: Theory and Methods*. 2nd ed. New York: Springer.
- Brockwell, P. J. and Davis, R. A. (2002). *Introduction to Time Series and Forecasting*. 2nd ed. New York: Springer.
- Brown, R. G. (1962). *Smoothing, Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs, NJ: Prentice-Hall.
- Chan, K. S. (1991). "Percentage points of likelihood ratio tests for threshold autoregression." *Journal of the Royal Statistical Society B*, **53**, 3, 691–696.
- Chan, K. S. (1993). "Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model." *Annals of Statistics*, **21**, 1, 520–533.
- Chan, K. S. (2008). "A new look at model diagnostics, with applications to time series analysis." Unpublished manuscript.
- Chan, K. S., Mysterud, A., Oritsland, N. A., Severinsen, T., and Stenseth, N. C. (2005). "Continuous and discrete extreme climatic events affecting the dynamics of a high Arctic reindeer population." *Oecologia*, **145**, 556–563.
- Chan, K. S., Petruccelli, J. D., Tong, H. and Woolford, S. W. (1985). "A multiple threshold AR(1) model." *Journal of Applied Probability*, **22**, 267–279.
- Chan, K. S. and Tong, H. (1985). "On the use of the deterministic Lyapunov function for the ergodicity of stochastic difference equations." *Advances in Applied Probability*, **17**, 666–678.
- Chan, K. S. and Tong, H. (1994). "A note on noisy chaos." *Journal of the Royal Statistical Society B*, **56**, 2, 301–311.
- Chan, K. S. and Tong, H. (2001). *Chaos: A Statistical Perspective*. New York: Springer-Verlag.
- Chan, K. S. and Tsay, R. S. (1998). "Limiting properties of the least squares estimator of a continuous threshold autoregressive model." *Biometrika*, **85**, 413–426.
- Chan, W. S. (1999). "A comparison of some pattern identification methods for order determination of mixed ARMA models." *Statistics and Probability Letters*, **42**, 69–79.
- Chang, I., Tiao, G. C., and Chen, C. (1988). "Estimation of time series parameters in the presence of outliers." *Technometrics*, **30**, 2, 193–204.

- Chang, Y. and Park, J. Y. (2002). "On the asymptotics of ADF tests for unit roots." *Econometric Reviews*, **21**, 431–447.
- Chatfield, C. (2004). *The Analysis of Time Series*, 6th ed. London: Chapman and Hall.
- Chen, Y. T. and Kuan, C. M. (2006). "A generalized Jarque-Bera test of conditional normality." www.sinica.edu.tw/~ckuan/pdf/jb01.pdf
- Cheng, B. and Tong, H. (1992). "On consistent nonparametric order determination and chaos (Disc: pp. 451-474)." *Journal of the Royal Statistical Society B*, **54**, 427–449.
- Cline, D. B. H. and Pu, H. H. (2001). "Stability of nonlinear time series: What does noise have to do with it?." In *Selected Proceedings of the Symposium on Inference for Stochastic Processes*, IMS Lecture Notes Monograph Series, Volume 37, Edited by I. V. Basawa, C. C. Heyde, and R. L. Taylor, 151–170. Beachwood, OH: Institute of Mathematical Statistics.
- Cooley, J. W. and Tukey, J. W. (1965). "An algorithm for the machine calculation of complex Fourier series." *Mathematics of Computation*, **19**, 297–301.
- Cramér, H. and Leadbetter, M. R. (1967). *Stationary and Related Random Processes*. New York: John Wiley & Sons.
- Cryer, J. D. and Ledolter, J. (1981). "Small-sample properties of the maximum likelihood estimator in the first-order moving average model." *Biometrika*, **68**, 3, 691–694.
- Cryer, J. D., Nankervis, J. C., and Savin, N. E. (1989). "Mirror-Image and Invariant Distributions in Arma Models." *Econometric Theory*, **5**, 1, 36–52.
- Cryer, J. D., Nankervis, J. C., and Savin, N. E. (1990). "Forecast Error Symmetry in ARIMA Models." *Journal of the American Statistical Association*, **85**, 41, 724–728.
- Cryer, J. D. and Ryan, T. P. (1990). "The estimation of sigma for an X chart." *Journal of Quality Technology*, **22**, 3, 187–192.
- Davies, N., Triggs, C. M., and Newbold, P. (1977). "Significance levels of the Box-Pierce portmanteau statistic in finite samples." *Biometrika*, **64**, 517–522.
- Davison, A. C. and Hinkley, D. V. (2003). *Bootstrap Methods and Their Application*, 2nd ed. New York: Cambridge University Press.
- Diggle, P. J. (1990). *Time Series: A Biostatistical Introduction*. Oxford: Oxford University Press.
- Draper, N. R. and Smith, H. (1981). *Applied Regression Analysis*, 2nd ed. New York: John Wiley & Sons.
- Durbin, J. (1960). "The fitting of time series models." *Review of the International Institute of Statistics*, **28**, 233–244.

- Durbin, J. (1970). "Testing for serial correlation in least-squares regression when some of the regressors are lagged independent variables." *Econometrika*, **38**, 410-421.
- Durbin, J. and Koopman, S. J. (2001). *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.
- Durbin, J. and Watson, G. S. (1950). "Testing for serial correlation in least-squares regression: I." *Biometrika*, **37**, 409-428.
- Durbin, J. and Watson, G. S. (1951). "Testing for serial correlation in least-squares regression: II." *Biometrika*, **38**, 1-19.
- Durbin, J. and Watson, G. S. (1971). "Testing for serial correlation in least-squares regression: III." *Biometrika*, **58**, 409-428.
- Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. New York: Chapman and Hall.
- Engle, R. F. (1982). "Autoregressive conditional heteroscedasticity with estimates of the variance of U.K. inflation." *Econometrica*, **50**, 987-1007.
- Fay, G., Moulines, E., and Soulier, P. (2002). "Nonlinear functionals of the periodogram." *Journal of Time Series Analysis*, **23**, 5, 523-553.
- Fan, J. and Gijbels, I. (1996). *Local Polynomial Modeling and Its Applications*. London: Chapman and Hall.
- Fan, J., and Kreutzberger, E. (1998). "Automatic local smoothing for spectral density estimation." *Scandinavian Journal of Statistics*, **25**, 2, 359-369.
- Fuller, W. A. (1996). *Introduction to Statistical Time Series*, 2nd ed. New York: John Wiley & Sons.
- Furnival, G. M. and Wilson, Jr., R. W. (1974). "Regressions by leaps and bounds." *Technometrics*, **16**, 4, 499-511.
- Gardner, G., Harvey, A. C. and Phillips, G. D. A. (1980). Algorithm AS154. "An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering." *Applied Statistics*, **29**, 311-322.
- Gentleman, W. M. and Sande, G. (1966). "Fast Fourier transforms—for fun and profit." *Proc. American Federation of Information Processing Society*, **29**, 563-578.
- Geweke, J. and Terui, N. (1993). "Bayesian threshold autoregressive models for nonlinear time series." *Journal of Time Series Analysis*, **14**, 441-454.
- Goldberg, S. I. (1958). *Introduction to Difference Equations*. New York: Science Editions.
- Granger, C. W. J. and Teräsvirta, T. (1993). *Modelling Nonlinear Economic Relationships*. New York: Oxford University Press.
- Hannan, E. J. (1970). *Multiple Time Series*. New York: John Wiley & Sons.

- Hannan, E. J. (1973). "The asymptotic theory of linear time-series models." *Journal of Applied Probability*, **10**, 130–145.
- Hannan, E. J. and Rissanen, J. (1982). "Recursive estimation of mixed autoregressive-moving average order." *Biometrika*, **69**, 81–94.
- Harvey, A. C. (1981a). *The Econometric Analysis of Time Series*. Oxford: Phillip Allen.
- Harvey, A. C. (1981b). "Finite sample prediction and overdifferencing." *Journal of Time Series Analysis*, **2**, 221–232.
- Harvey, A. C. (1981c). *Time Series Models*. New York: Halsted Press.
- Harvey, A. C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.
- Harvey, A. C. (1990). *The Econometric Analysis of Time Series*, 2nd ed. Boston: MIT Press.
- Harvey, A. C. (1993). *Time Series Models*, 2nd ed. New York: Harvester Wheatsheaf.
- Harvey, A., Koopman, S. J., and Shephard, N. (2004). *State Space and Unobserved Component Models: Theory and Applications*. New York: Cambridge University Press.
- Hasza, D. P. (1980). The asymptotic distribution of the sample autocorrelation for an integrated ARMA process." *Journal of the American Statistical Association*, **75**, 349–352.
- Hurvich, C. M. and Tsai, C. L. (1989). "Regression and time series model selection in small samples." *Biometrika*, **76**, 2, 297–307.
- Jenkins, G. M. and Watts, D. G. (1968). *Spectral Analysis and Its Applications*. San Francisco: Holden-Day.
- Jiang, J. and Hui, Y. V. (2004). "Spectral density estimation with amplitude modulation and outlier detection." *Annals of the Institute of Statistical Mathematics*, **56**, 4, 611.
- Jones, R. H. (1980). "Maximum likelihood fitting of ARMA models to time series with missing observations." *Technometrics*, **20**, 389–395.
- Jost, C. and Ellner, S. P. (2000). "Testing for predator dependence in predator-prey dynamics: A non-parametric approach." *Proceedings of the Royal Society B: Biological Sciences*, **267**, 1453, 1611–1620.
- Kakizawa, Y. (2006). "Bernstein polynomial estimation of a spectral density". *Journal of Time Series Analysis*, **27**, 2, 253–287.
- Keenan, D. (1985). "A Tukey nonlinear type test for time series nonlinearities." *Biometrika*, **72**, 39–44.
- Kooperberg, C., Stone, C. J., and Truong, Y. K. (1995). "Logspline estimation of a possibly mixed spectral distribution." *Journal of Time Series Analysis*, **16**, 359–388.

- Lai, T. L. and Wei, C. Z. (1983). "Asymptotic properties of general autoregressive models and strong consistency of least squares estimates of their parameters." *Journal of Multivariate Analysis*, **13**, 1–13.
- Levinson, N. (1947). "The Weiner RMS error criterion in filter design and prediction." *Journal of Mathematical Physics*, **25**, 261–278.
- Li, W. K. (2004). *Diagnostic Checks in Time Series*. London: Chapman and Hall.
- Li, W. K. and Mak, T. K. (1994). "On the squared residual autocorrelations in non-linear time series with conditional heteroskedasticity." *Journal of Time Series Analysis*, **15**, 627–636.
- Ling, S. and McAleer, M. (2002). "Stationarity and the existence of moments of a family of GARCH processes." *Journal of Econometrics*, **106**, 109–117.
- Ljung, G. M. and Box, G. E. P. (1978). "On a measure of lack of fit in time series models." *Biometrika*, **65**, 553–564.
- Luukkonen, R., Saikkonen, P., and Teräsvirta, T. (1988). "Testing linearity against smooth transition autoregressive models." *Biometrika*, **75**, 491–499.
- MacLulich, D. A. (1937). *Fluctuations in the number of the varying hare (Lepus americanus)*. Toronto: University of Toronto Press.
- May, R. M. (1976). "Simple mathematical models with very complicated dynamics." *Nature*, **261**, 459–467.
- McLeod, A. I. (1978). "On the distribution of residual autocorrelations in Box-Jenkins models." *Journal of the Royal Statistical Society A*, **40**, 296–302.
- McLeod, A. I. and W. K. Li (1983). "Diagnostic checking ARMA time series models using squared residual autocorrelations." *Journal of Time Series Analysis*, **4**, 269–273.
- Montgomery, D. C. and Johnson, L. A. (1976). *Forecasting and Time Series Analysis*. New York: McGraw-Hill.
- Nadaraya, E. A. (1964). "On estimating regression." *Theory of Probability and Its Applications*, **9**, 141–142.
- Nelson, C. R. (1973). *Applied Time Series Analysis for Managerial Forecasting*. San Francisco: Holden-Day.
- Nelson, D. B. and Cao, C. Q. (1992). "Inequality constraints in the univariate GARCH model." *Journal of Business and Economic Statistics*, **10**, 229–235.
- Ong, C. S., Huang, J. J., and Tzeng, G. H. (2005). "Model Identification of ARIMA family using genetic algorithms." *Applied Mathematics and Computation*, **164**, 885–912.
- Parzen, E. (1982). "ARARMA models for time series analysis and forecasting." *Journal of Forecasting*, **1**, 67–82.

- Percival, D. B. and Walden, A. T. (1993). *Spectral Analysis for Physical Applications*. Cambridge: Cambridge University Press.
- Phillips, P. C. B. (1998). "New tools for understanding spurious regressions." *Econometrica*, **66**, 1299–1325.
- Phillips, P. C. B. and Xiao, Z. (1998). "A primer on unit root testing." *Journal of Economic Surveys*, **12**, 5, 423–469.
- Politis, D. N. (2003). "The impact of bootstrap methods on time series analysis." *Statistical Science*, **18**, 2, 219–230.
- Priestley, M. B. (1981). *Spectral Analysis and Time Series*, Volumes 1 and 2. New York: Academic Press.
- Quenouille, M. H. (1949). "Approximate tests of correlation in time series." *Journal of the Royal Statistical Society B*, **11**, 68–84.
- Roberts, H. V. (1991). *Data Analysis for Managers with Minitab*, second edition. Redwood City, CA, The Scientific Press.
- Roy, R. (1977). "On the asymptotic behaviour of the sample autocovariance function for an integrated moving average process." *Biometrika*, **64**, 419–421.
- Royston, P. (1982). "An extension of Shapiro and Wilk's *W* test for normality to large samples." *Applied Statistics*, **31**, 115–124.
- Said, S. E. and Dickey, D. A. (1984). "Testing for unit roots in autoregressive-moving average models of unknown order." *Biometrika*, **71**, 599–607.
- Samia, N. I., Chan, K. S., and Stenseth, N. C. (2007). "A generalised threshold mixed model for analysing nonnormal nonlinear time series; with application to plague in Kazakhstan." *Biometrika*, **94**, 101–118.
- Schuster, A. (1897). "On lunar and solar periodicities of earthquakes." *Proceedings of the Royal Society*, **61**, 455–465.
- Schuster, A. (1898). "On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena." *Terrestrial Magnetism*, **3**, 13–41.
- Shephard, N. (1996). "Statistical aspect of ARCH and stochastic volatility." In *Time Series Models: In Econometrics, Finance and Other Fields*, edited by D. R. Cox, D. V. Hinkley, and O. E. Barndorff-Nielsen. London: Chapman and Hall. 1–55.
- Shibata, R. (1976). "Selection of the order of an autoregressive model by Akaike's information criterion." *Biometrika*, **63**, 1, 117–126.
- Shin, K-I. and Kang, H-J. (2001). "A study on the effect of power transformation in the ARMA(p, q) model." *Journal of Applied Statistics*, **28**, 8, 1019–1028.
- Shumway, R. H. and Stoffer, D. S. (2006). *Time Series Analysis and Its Applications (with R Examples)*, 2nd ed. New York: Springer.

- Slutsky, E. (1927). "The summation of random causes as the source of cyclic processes" (in Russian). In *Problems of Economic Conditions*, English translation (1937) in *Econometrica*, **5**, 105–146.
- Stige, L. C., Stave, J., Chan, K-S, Ciannelli, L., Pettorelli, N., Glantz, M., Herren, H. R., and Stenseth, N. (2006). "The effect of climate variation on agro-pastoral production in Africa." *Proceedings of the National Academy of Science*, **103**, 9, 3049–3053
- Stenseth, N. C., Chan, K. S., Tavecchia, G., Coulson T., Mysterud, A., Clutton-Brock, T., and Grenfell, B. (2004). "Modelling non-additive and nonlinear signals from climatic noise in ecological time series: Soay sheep as an example." *Proceedings of the Royal Society of London Series B: Biological Sciences*, **271**, 1985–1993.
- Stenseth, N. C., Chan, K. S., Tong, H., Boonstra, R., Boutin, S., Krebs, C. J., Post, E., O'Donoghue, M., Yoccoz, N. G., Forchhammer, M. D., and Hurrell, J. W. (1999). "Common dynamic structure of Canada lynx populations within three climatic regions." *Science*, August 13, 1071–1073.
- Stenseth, N. C., Falck, W., Bjørnstad, O. N., and Krebs, C. J. (1997). "Population regulation in snowshoe hare and Canadian lynx: Asymmetric food web configurations between hare and lynx." *Proceedings of the National Academy of Science USA*, **94**, 5147–5152.
- Taylor, S. J. (1986). *Modeling Financial Time Series*. Chichester: John Wiley & Sons.
- The R Development Core Team (2006a). *R: A Language and Environment for Statistical Computing Reference Index*, Version 2.4.1 (2006-12-18).
- The R Development Core Team (2006b). *R Data Import/Export*, Version 2.4.1 (2006-12-18).
- Tong, H. (1978). "On a threshold model." In *Pattern Recognition and Signal Processing*, edited by C. H. Chen. Amsterdam: Sijthoff and Noordhoff.
- Tong, H. (1983). *Threshold Models In Non-linear Time Series Analysis*. New York: Springer-Verlag. 101–141.
- Tong, H. (1990). *Non-linear Time Series*. Oxford: Clarendon Press.
- Tong, H. (2007). "Birth of the threshold time series model." *Statistica Sinica*, **17**, 8–14.
- Tong, H. and Lim, K. S. (1980). "Threshold autoregression, limit cycles and cyclical data (with discussion)." *Journal of the Royal Statistical Society B*, **42**, 245–292.
- Tsai, H. and Chan, K. S. (2006). A note on the non-negativity of continuous-time ARMA and GARCH processes. Technical Report No. 359, Department of Statistics & Actuarial Science, The University of Iowa.
- Tsay, R. S. (1984). "Regression models with time series errors." *Journal of the American Statistical Association*, **79**, 385, 118–24.

- Tsay, R. S. (1986). "Nonlinearity tests for time series." *Biometrika*, **73**, 461–466.
- Tsay, R. S. (2005). *Analysis of Financial Time Series*, 2nd ed. New York: John Wiley & Sons.
- Tsay, R. S. and Tiao, G. (1984). "Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary ARMA Models." *Journal of the American Statistical Association*, **79**, 385, 84–96.
- Tsay, R. and Tiao, G. (1985). "Use of canonical analysis in time series model identification." *Biometrika*, **72**, 299–315.
- Tufte, E. (1983). *The Visual Display of Quantitative Information*. Cheshire, CT.: Graphics Press.
- Tukey, J. W. (1949). "One degree of freedom for non-additivity." *Biometrics*, **5**, 232–242.
- Veilleux, B. G. (1976). "The analysis of a predatory interaction between *Didinium* and *Paramecium*." MSc thesis, University of Alberta, Canada.
- Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*, 4th ed. New York: Springer.
- Venables, W. N., Smith, D. M., and the R Development Core Team (2006). *An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics*. Version 2.4.1 (2006-12-18).
- Watson, G. S. (1964). "Smooth Regression Analysis." *Sankhyā*, **26**, 359–372.
- Wei, W. W. S. (2005). *Time Series Analysis*, 2nd ed. Redwood City, CA: Addison-Wesley.
- Wichern, D. W. (1973). "The behavior of the sample autocorrelation function for an integrated moving average process." *Biometrika*, **60**, 235–239.
- Wiener, N. (1958). *Nonlinear Problems in Random Theory*. Cambridge, MA: MIT Press.
- White, H. (1989). "An additional hidden unit test for neglected nonlinearities in multi-layer feedforward networks." In *Proceedings of the International Joint Conference on Neural Networks*, New York: IEEE Press, 451–455.
- Whittaker, E. T. and Robinson, G., (1924). *The Calculus of Observations*. London: Blackie and Son.
- Wold, H. O. A. (1938). *A Study of the Analysis of Stationary Time Serie*. (2nd ed. 1954). Uppsala: Almqvist and Wiksell.
- Wold, H. O. A. (1948). "On prediction in stationary time series." *The Annals of Mathematical Statistics*, **19**, 558–567.

- Yeo, I-K and Johnson, R. A. (2000) “A new family of power transformations to improve normality or symmetry.” *Biometrika* **87**, 954–959.
- Yoshihide, K. (2006). “Bernstein polynomial estimation of a spectral density.” *Journal of Time Series Analysis*, **27**, 2, 253–287.
- Yule, G. U. (1926). “Why do we sometimes get nonsense-correlations between time-series? — A study in sampling and the nature of time-series.” *Journal of the Royal Statistical Society*, **89**, 1, 1–63.

INDEX

A

additive outlier (AO) 257
airline model 241
Akaike's
 information criterion (AIC) 130
 Markovian representation 223
aliased frequencies 326
amplitude 34, 319
AR characteristic polynomial 71
AR(1) 66
AR(2) 71
 ψ -coefficients 75
ARCH
 model 285
 order q 289
ARMA(1,1) model 77
attractor 407
autocorrelation function 11
 extended 115
 for ARMA(p,q) 85
 partial 113
 residuals 180
 sample or estimated 46
 sample properties 109
autocovariance function 11
 sample or estimated 329
autoregressive process
 general 66, 76
 order one 66
 order two 71
averaged sample spectral density 351

B

backshift operator 106
 lag operator 106
bandwidth 356
bandwidth guide 357
Bartlett lag window 377
Bayesian information criterion
 (BIC) 131
best linear unbiased estimates
 (BLUE) 40
Black-Scholes formula 307
blue spectrum 333
Bonferroni rule 258
bootstrap 445
bootstrapping ARIMA models 167

C

Canadian hare 4
causal filter 331
coefficient of determination 41
color property 3
comma-separated values 427
complementary function 221
complex conjugate 308
concentrated log-likelihood
 function 226
conditional
 expectation 218
 sum-of-squares function 154
 volatility 285
conditional variance process 277
confidence interval guide 357
constant terms
 ARIMA models 97
contemporaneous 260
convolution
 discrete time 331
correlation 26
correlogram 46
cosine
 bell 360
 split 360
 trend 34
 wave 18
covariance 25
 matrix 224
CREF
 bond fund 316
 stock fund 278
cross-correlation function 260
 sample 261
cyclical trend 32

D

damping factor 73
Daniell spectral window 352
 modified 353
De Moivre's theorem 350
delay 251
Denver public transportation usage 271
deterministic trend 27
diagnostics 8
Dickey-Fuller unit-root test 128

difference 90
 seasonal 233
 second 91
 Dirichlet spectral window 359
 modified 382
 tapering 381
 discrete Fourier transform 329
 discrete spectrum 328
 discrete time convolution 331
 distributed lag model 267
 distribution
 heavy-tailed 284
 light-tailed 284
 distribution function
 spectral 328
 dynamic regression model 267

E

equations of state 223
 equilibrium point 398
 ergodicity 398
 error variance
 forecast 192
 Euler's formulas 350
 euphonium 374, 381
 EWMA (exponentially weighted
 moving average)
 smoothing constant 209
 expectation
 conditional 218
 expected value 24
 exponentially weighted moving average
 (EWMA) 208
 extended autocorrelation function 115

F

filter 265, 331
 first difference 90
 first-order
 autoregressive process, AR(1) 66
 moving average process, MA(1) 57
 fitting 8
 forecast
 ARMA(p,q) 199
 error variance 192
 lead time 191
 MA(1) model 197
 nonstationary models 201
 origin 191
 random walk with drift 198
 unbiased 192
 with differencing 209
 with log transformations 210

forecast error 192
 one-step-ahead 195
 Fourier frequencies 38, 321
 Fourier transform
 discrete 329
 frequency 34, 73, 319
 frequency domain analysis 319

G

GARCH
 order (p, q) 289
 weak stationarity condition 296
 general autoregressive process
 AR(p) 76
 general linear process 55
 generalized autoregressive conditional
 heteroscedasticity 289
 generalized least squares (GLS) 40
 GJR model 311
 globally exponentially stable limit
 point, 398
 Google stock returns 317, 472

H

half-life 251
 hare 4
 heavy-tailed 284

I

IGARCH(1,1) 297
 innovation 257, 285
 innovative outlier (IO) 257
 integrated autoregressive moving
 average (ARIMA) 92
 intervention analysis 249
 invertibility 79
 iterated-expectation formula 288

J

Jarque-Bera test 284

K

Kalman
 filter equations 224
 filtering 222
 Kullback-Leibler divergence 130
 kurtosis 284

L

lag operator 106

lag window 377
Bartlett or triangular 377
rectangular 377
truncated 377
lead 266
lead time 191
leading indicator 266
leakage 325
least-squares estimation
autoregressive models 154
mixed models 157
moving average models 156
light-tailed 284
likelihood function
concentrated 226
line spectrum 328
linear filter 331
linear trend 30
Ljung-Box portmanteau test 183
logarithms 99
logical variable 444
log-likelihood function 160
concentrated 226
Los Angeles rainfall 1

M

MA characteristic polynomial 80
MA(1) process 57
MA(q) process 65
martingale differences 383
maximum likelihood estimation 158
McLeod-Li test 283
mean
function 11
sample 28
mean square error of prediction 218
median 211
method of moments 149
AR models 149
mixed models 151
moving average models 150
mixed autoregressive moving average model
ARMA(p,q) 77
model-building strategy
specification 8
modified
Daniell spectral window 353
Dirichlet kernel 382
moving average 14
order q 57, 65

multiplicative seasonal
ARIMA model 234
ARMA(p, q) $\times(P, Q)s$ model 231

N

negative frequencies 326
noise
white 17
nonsense correlation (spurious) 264
normality of residuals 178

O

observational equation 223
oil filter sales 6
one-step-ahead forecast error 195
orthogonal increments 328
orthogonality 349
outlier
additive (AO) 257
innovative (IO) 257
overdifferencing 126
overfitting 185

P

parameter redundancy 187
parsimony 8
partial autocorrelation 113
sample 115
particular solution 221
percentage changes 99
period 34
periodogram 322
phase 18, 34, 73, 319
plots of residuals 176
portmanteau test 183
generalized 304
power transfer function 332
power transformations 101
prediction error decomposition 225
prediction limits
ARIMA models 204
deterministic models 203
pre-intervention data 250
prewhitening 265
profile log-likelihood function 402
pulse function 251
purely deterministic process 383
purely discrete spectrum 328

Q

quadratic trend 30
quantile-quantile plot (QQ) 45
quasi-likelihood estimators 301

quasi-period 74

R

random cosine wave 18
 random walk 12
 with drift 22
 rational spectral density 339
 Rayleigh distribution 24
 rectangular lag window 377
 rectangular spectral window 352
 red spectrum 333
 regression methods 30
 representation 328
 residual 42
 autocorrelation 180
 plots 176
 residual analysis 42, 175
 residual standard deviation 41
 residuals
 normality 178
 returns 99, 278
 Riskmetrics software 297
 R-squared 41
 runs test 46

S

sample
 (auto)covariance function 329
 autocorrelation function 46
 cross-correlation function 261
 mean 28
 partial autocorrelation 115
 spectral density 329
 smoothed 351
 scaled spectral distribution
 function 329
 seasonal
 AR characteristic polynomial 230
 AR(P) model 230
 cosine trend 34
 difference 233
 MA characteristic polynomial 229
 MA(Q) model 229
 means 32
 period 229
 trend 32
 seasonality 6
 second-order
 autoregressive process, AR(2) 71
 moving average process, MA(2) 62
 semivariogram 23
 signal plus noise 323
 significant digits 446

skeleton 397
 skewness 284
 smoothed sample spectral density 351
 bias 355
 variance 355
 smoothing constant (in EWMA) 209
 spectral 328
 distribution function 328
 distribution interpretation 329
 scaled distribution function 329
 spectral density 330
 rational 339
 sample 329
 spectral window 351
 bandwidth 356
 Daniell 352
 Dirichlet 359
 modified Daniell 353
 rectangular 352
 spectrum
 blue 333
 red 333
 split cosine bell taper 360
 spurious correlation 264
 standard deviation 25
 residual 41
 standardized 25
 state space model 222
 stationarity conditions 16
 AR(1) 71
 AR(2) 71, 72, 84
 AR(p) 76
 ARMA(p,q) 78
 intrinsically 23
 second-order 17
 strict 16
 weak 17
 step function 250
 stochastic
 process 11
 seasonal models 228
 trend 27
 strange attractor 407

T

tapering 359
 TAR model 399
 temperatures
 Dubuque, Iowa 6
 Tempered Brass 472, 474
 tenor tuba 381
 threshold autoregressive model 399
 time domain analysis 319

time-invariant linear filter 331
causal 331
trading days 278
transfer-function model 267
trend
 cosine 34
 cyclical 32
 deterministic 27
 linear 30
 quadratic 30
 seasonal 32
 seasonal means 32
 stochastic 27
triangular lag window 377
trombone 374, 381
truncated
 lag window 377
 linear process 200, 221
TSA Library 468
tuba 381

U

unbiased forecast 192
unconditional
 least squares 160
 sum-of-squares function 159
unit-root test 128
unperturbed process 250

updating equation 207

V

Value at Risk (VaR) 277
variance 25
volatility 285
 clustering 279

W

weakly stationary 17
weight function 351
white light 332
white noise 17, 332
whitening 265
window closing 355
Wold decomposition 383
working directory 423
workspace 423

Y

ψ -coefficients
 AR(2) 75
Yule-Walker equations
 AR(2) 72
 AR(p) 76
Yule-Walker estimates 150

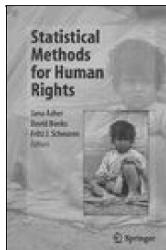


Time Series Analysis and its Applications with R Examples

Robert H. Shumway and David S. Stoffer

Time Series Analysis and Its Applications presents a balanced and comprehensive treatment of both time and frequency domain methods with accompanying theory. Numerous examples using non-trivial data illustrate solutions to problems such as evaluating pain perception experiments using magnetic resonance imaging or monitoring a nuclear test ban treaty. The book is designed to be useful as a text for graduate level students in the physical, biological and social sciences and as a graduate level text in statistics.

2nd ed., 2006, XIII, 575 pp. Hardcover ISBN 978-0-387-29317-2

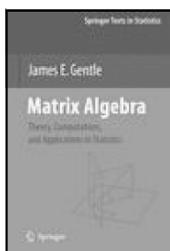


Statistical Methods for Human Rights

Jana Asher, David Banks and Fritz J. Scheuren (Eds.)

Human rights issues are shaping the modern world. They define the expectations by which nations are judged and affect the policy of governments, corporations, and foundations. This book describes the statistics that underlie the social science research in human rights. It includes case studies, methodology, and research papers that discuss the fundamental measurement issues. It is intended as an introduction to applied human rights research.

2007, Approx 410 pp. Softcover ISBN 978-0-387-72836-0



Matrix Algebra Theory, Computations, and Applications in Statistics

James E. Gentle

Matrix algebra is one of the most important areas of mathematics for data analysis and for statistical theory. The first part of this book presents the relevant aspects of the theory of matrix algebra for applications in statistics. Next begins a consideration of various types of matrices encountered in statistics and describes the special properties of those matrices. Additionally, this book covers numerical linear algebra.

2007, X,XII 528 pp. Hardcover ISBN 978-0-78702-0

Easy Ways to Order ►

Call: Toll-Free 1-800-SPRINGER • E-mail: orders-ny@springer.com • Write: Springer, Dept. S8113, PO Box 2485, Secaucus, NJ 07096-2485 • Visit: Your local scientific bookstore or urge your librarian to order.