Springer Texts in Statistics

Robert H. Shumway
David S. Stoffer

Time Series Analysis and Its Applications

With R Examples

Third Edition

Springer

Springer Texts in Statistics

Series Editors
G. Casella
S. Fienberg
I. Olkin

For other titles published in this series, go to
www.springer.com/series/417

Robert H. Shumway • David S. Stoffer

Time Series Analysis and Its Applications

With R Examples

Third edition

Springer

Prof. Robert H. Shumway
Department of Statistics
University of California
Davis, California
USA

Prof. David S. Stoffer
Department of Statistics
University of Pittsburgh
Pittsburgh, Pennsylvania
USA

ISSN 1431-875X
ISBN 978-1-4419-7864-6 e-ISBN 978-1-4419-7865-3
DOI 10.1007/978-1-4419-7865-3
Springer New York Dordrecht Heidelberg London

To my wife, Ruth, for her support and joie de vivre, and to the memory of my thesis adviser, Solomon Kullback.
R.H.S.

To my family and friends, who constantly remind me what is important.
D.S.S.

Preface to the Third Edition

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, and still maintain a commitment to theoretical integrity, as exemplified by the seminal works of Brillinger (1975) and Hannan (1970) and the texts by Brockwell and Davis (1991) and Fuller (1995). The advent of inexpensive powerful computing has provided both real data and new software that can take one considerably beyond the fitting of simple time domain models, such as have been elegantly described in the landmark work of Box and Jenkins (1970). This book is designed to be useful as a text for courses in time series on several different levels and as a reference work for practitioners facing the analysis of time-correlated data in the physical, biological, and social sciences.

We have used earlier versions of the text at both the undergraduate and graduate levels over the past decade. Our experience is that an undergraduate course can be accessible to students with a background in regression analysis and may include §1.1–§1.6, §2.1–§2.3, the results and numerical parts of §3.1–§3.9, and briefly the results and numerical parts of §4.1–§4.6. At the advanced undergraduate or master's level, where the students have some mathematical statistics background, more detailed coverage of the same sections, with the inclusion of §2.4 and extra topics from Chapter 5 or Chapter 6 can be used as a one-semester course. Often, the extra topics are chosen by the students according to their interests. Finally, a two-semester upper-level graduate course for mathematics, statistics, and engineering graduate students can be crafted by adding selected theoretical appendices. For the upper-level graduate course, we should mention that we are striving for a broader but less rigorous level of coverage than that which is attained by Brockwell and Davis (1991), the classic entry at this level.

The major difference between this third edition of the text and the second edition is that we provide R code for almost all of the numerical examples. In addition, we provide an R supplement for the text that contains the data and scripts in a compressed file called `tsa3.rda`; the supplement is available on the website for the third edition, `http://www.stat.pitt.edu/stoffer/tsa3/`,

or one of its mirrors. On the website, we also provide the code used in each example so that the reader may simply copy-and-paste code directly into R. Specific details are given in Appendix R and on the website for the text. Appendix R is new to this edition, and it includes a small R tutorial as well as providing a reference for the data sets and scripts included in `tsa3.rda`. So there is no misunderstanding, we emphasize the fact that this text is about time series analysis, not about R. R code is provided simply to enhance the exposition by making the numerical examples reproducible.

We have tried, where possible, to keep the problem sets in order so that an instructor may have an easy time moving from the second edition to the third edition. However, some of the old problems have been revised and there are some new problems. Also, some of the data sets have been updated. We added one section in Chapter 5 on unit roots and enhanced some of the presentations throughout the text. The exposition on state-space modeling, ARMAX models, and (multivariate) regression with autocorrelated errors in Chapter 6 have been expanded. In this edition, we use standard R functions as much as possible, but we use our own scripts (included in `tsa3.rda`) when we feel it is necessary to avoid problems with a particular R function; these problems are discussed in detail on the website for the text under R Issues.

We thank John Kimmel, Executive Editor, Springer Statistics, for his guidance in the preparation and production of this edition of the text. We are grateful to Don Percival, University of Washington, for numerous suggestions that led to substantial improvement to the presentation in the second edition, and consequently in this edition. We thank Doug Wiens, University of Alberta, for help with some of the R code in Chapters 4 and 7, and for his many suggestions for improvement of the exposition. We are grateful for the continued help and advice of Pierre Duchesne, University of Montreal, and Alexander Aue, University of California, Davis. We also thank the many students and other readers who took the time to mention typographical errors and other corrections to the first and second editions. Finally, work on the this edition was supported by the National Science Foundation while one of us (D.S.S.) was working at the Foundation under the Intergovernmental Personnel Act.

Davis, CA — *Robert H. Shumway*
Pittsburgh, PA — *David S. Stoffer*
September 2010

Contents

1

Characteristics of Time Series

1.1 Introduction

The analysis of experimental data that have been observed at different points in time leads to new and unique problems in statistical modeling and inference. The obvious correlation introduced by the sampling of adjacent points in time can severely restrict the applicability of the many conventional statistical methods traditionally dependent on the assumption that these adjacent observations are independent and identically distributed. The systematic approach by which one goes about answering the mathematical and statistical questions posed by these time correlations is commonly referred to as time series analysis.

The impact of time series analysis on scientific applications can be partially documented by producing an abbreviated listing of the diverse fields in which important time series problems may arise. For example, many familiar time series occur in the field of economics, where we are continually exposed to daily stock market quotations or monthly unemployment figures. Social scientists follow population series, such as birthrates or school enrollments. An epidemiologist might be interested in the number of influenza cases observed over some time period. In medicine, blood pressure measurements traced over time could be useful for evaluating drugs used in treating hypertension. Functional magnetic resonance imaging of brain-wave time series patterns might be used to study how the brain reacts to certain stimuli under various experimental conditions.

Many of the most intensive and sophisticated applications of time series methods have been to problems in the physical and environmental sciences. This fact accounts for the basic engineering flavor permeating the language of time series analysis. One of the earliest recorded series is the monthly sunspot numbers studied by Schuster (1906). More modern investigations may center on whether a warming is present in global temperature measurements

R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer Texts in Statistics, DOI 10.1007/978-1-4419-7865-3_1,

or whether levels of pollution may influence daily mortality in Los Angeles. The modeling of speech series is an important problem related to the efficient transmission of voice recordings. Common features in a time series characteristic known as the power spectrum are used to help computers recognize and translate speech. Geophysical time series such as those produced by yearly depositions of various kinds can provide long-range proxies for temperature and rainfall. Seismic recordings can aid in mapping fault lines or in distinguishing between earthquakes and nuclear explosions.

The above series are only examples of experimental databases that can be used to illustrate the process by which classical statistical methodology can be applied in the correlated time series framework. In our view, the first step in any time series investigation always involves careful scrutiny of the recorded data plotted over time. This scrutiny often suggests the method of analysis as well as statistics that will be of use in summarizing the information in the data. Before looking more closely at the particular statistical methods, it is appropriate to mention that two separate, but not necessarily mutually exclusive, approaches to time series analysis exist, commonly identified as the time domain approach and the frequency domain approach.

The time domain approach is generally motivated by the presumption that correlation between adjacent points in time is best explained in terms of a dependence of the current value on past values. The time domain approach focuses on modeling some future value of a time series as a parametric function of the current and past values. In this scenario, we begin with linear regressions of the present value of a time series on its own past values and on the past values of other series. This modeling leads one to use the results of the time domain approach as a forecasting tool and is particularly popular with economists for this reason.

One approach, advocated in the landmark work of Box and Jenkins (1970; see also Box et al., 1994), develops a systematic class of models called autoregressive integrated moving average (ARIMA) models to handle time-correlated modeling and forecasting. The approach includes a provision for treating more than one input series through multivariate ARIMA or through transfer function modeling. The defining feature of these models is that they are multiplicative models, meaning that the observed data are assumed to result from products of factors involving differential or difference equation operators responding to a white noise input.

A more recent approach to the same problem uses additive models more familiar to statisticians. In this approach, the observed data are assumed to result from sums of series, each with a specified time series structure; for example, in economics, assume a series is generated as the sum of trend, a seasonal effect, and error. The state-space model that results is then treated by making judicious use of the celebrated Kalman filters and smoothers, developed originally for estimation and control in space applications. Two relatively complete presentations from this point of view are in Harvey (1991) and Kitagawa and Gersch (1996). Time series regression is introduced in Chapter 2, and ARIMA

and related time domain models are studied in Chapter 3, with the emphasis on classical, statistical, univariate linear regression. Special topics on time domain analysis are covered in Chapter 5; these topics include modern treatments of, for example, time series with long memory and GARCH models for the analysis of volatility. The state-space model, Kalman filtering and smoothing, and related topics are developed in Chapter 6.

Conversely, the frequency domain approach assumes the primary characteristics of interest in time series analyses relate to periodic or systematic sinusoidal variations found naturally in most data. These periodic variations are often caused by biological, physical, or environmental phenomena of interest. A series of periodic shocks may influence certain areas of the brain; wind may affect vibrations on an airplane wing; sea surface temperatures caused by El Niño oscillations may affect the number of fish in the ocean. The study of periodicity extends to economics and social sciences, where one may be interested in yearly periodicities in such series as monthly unemployment or monthly birth rates.

In spectral analysis, the partition of the various kinds of periodic variation in a time series is accomplished by evaluating separately the variance associated with each periodicity of interest. This variance profile over frequency is called the power spectrum. In our view, no schism divides time domain and frequency domain methodology, although cliques are often formed that advocate primarily one or the other of the approaches to analyzing data. In many cases, the two approaches may produce similar answers for long series, but the comparative performance over short samples is better done in the time domain. In some cases, the frequency domain formulation simply provides a convenient means for carrying out what is conceptually a time domain calculation. Hopefully, this book will demonstrate that the best path to analyzing many data sets is to use the two approaches in a complementary fashion. Expositions emphasizing primarily the frequency domain approach can be found in Bloomfield (1976, 2000), Priestley (1981), or Jenkins and Watts (1968). On a more advanced level, Hannan (1970), Brillinger (1981, 2001), Brockwell and Davis (1991), and Fuller (1996) are available as theoretical sources. Our coverage of the frequency domain is given in Chapters 4 and 7.

The objective of this book is to provide a unified and reasonably complete exposition of statistical methods used in time series analysis, giving serious consideration to both the time and frequency domain approaches. Because a myriad of possible methods for analyzing any particular experimental series can exist, we have integrated real data from a number of subject fields into the exposition and have suggested methods for analyzing these data.

1.2 The Nature of Time Series Data

Some of the problems and questions of interest to the prospective time series analyst can best be exposed by considering real experimental data taken

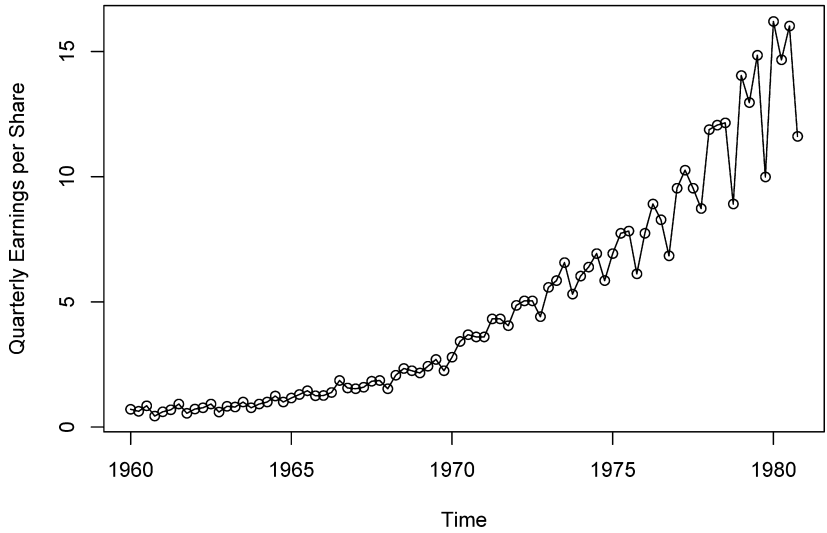

Fig. 1.1. Johnson & Johnson quarterly earnings per share, 84 quarters, 1960-I to 1980-IV.

from different subject areas. The following cases illustrate some of the common kinds of experimental time series data as well as some of the statistical questions that might be asked about such data.

Example 1.1 Johnson & Johnson Quarterly Earnings

Figure 1.1 shows quarterly earnings per share for the U.S. company Johnson & Johnson, furnished by Professor Paul Griffin (personal communication) of the Graduate School of Management, University of California, Davis. 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 gradually increasing underlying trend and the rather regular variation superimposed on the trend that seems to repeat over quarters. Methods for analyzing data such as these are explored in Chapter 2 (see Problem 2.1) using regression techniques and in Chapter 6, §6.5, using structural equation modeling.

To plot the data using the R statistical package, type the following:[1]

```
load("tsa3.rda")     # SEE THE FOOTNOTE
plot(jj, type="o", ylab="Quarterly Earnings per Share")
```

Example 1.2 Global Warming

Consider the global temperature series record shown in Figure 1.2. The data are the global mean land–ocean temperature index from 1880 to 2009, with

[1] We assume that `tsa3.rda` has been downloaded to a convenient directory. See Appendix R for further details.

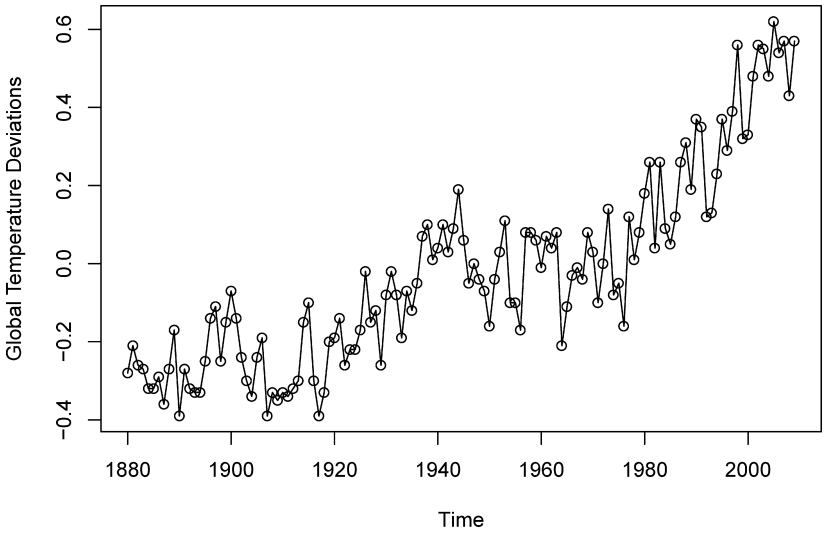

Fig. 1.2. Yearly average global temperature deviations (1880–2009) in degrees centigrade.

the base period 1951-1980. In particular, the data are deviations, measured in degrees centigrade, from the 1951-1980 average, and are an update of Hansen et al. (2006). We note an apparent upward trend in the series during the latter part of the twentieth century that has been used as an argument for the global warming hypothesis. Note also the leveling off at about 1935 and then another rather sharp upward trend at about 1970. The question of interest for global warming proponents and opponents is whether the overall trend is natural or whether it is caused by some human-induced interface. Problem 2.8 examines 634 years of glacial sediment data that might be taken as a long-term temperature proxy. Such percentage changes in temperature do not seem to be unusual over a time period of 100 years. Again, the question of trend is of more interest than particular periodicities.

The R code for this example is similar to the code in Example 1.1:

```
plot(gtemp, type="o", ylab="Global Temperature Deviations")
```

Example 1.3 Speech Data

More involved questions develop in applications to the physical sciences. Figure 1.3 shows a small .1 second (1000 point) sample of recorded speech for the phrase $aaa\cdots hhh$, and we note the repetitive nature of the signal and the rather regular periodicities. One current problem of great interest is computer recognition of speech, which would require converting this particular signal into the recorded phrase $aaa\cdots hhh$. Spectral analysis can be used in this context to produce a signature of this phrase that can be compared with signatures of various library syllables to look for a match.

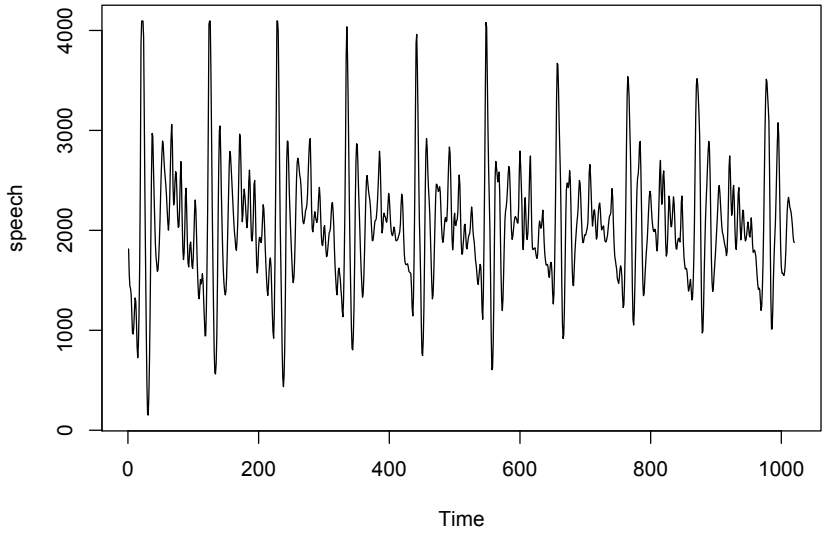

Fig. 1.3. Speech recording of the syllable $aaa \cdots hhh$ sampled at 10,000 points per second with $n = 1020$ points.

One can immediately notice the rather regular repetition of small wavelets. The separation between the packets is known as the pitch period and represents the response of the vocal tract filter to a periodic sequence of pulses stimulated by the opening and closing of the glottis.

In R, you can reproduce Figure 1.3 as follows:

```
plot(speech)
```

Example 1.4 New York Stock Exchange

As an example of financial time series data, Figure 1.4 shows the daily returns (or percent change) of the New York Stock Exchange (NYSE) from February 2, 1984 to December 31, 1991. It is easy to spot the crash of October 19, 1987 in the figure. The data shown in Figure 1.4 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 changes over time. In fact, the data show volatility clustering; that is, highly volatile periods tend to be clustered together. A problem in the analysis of these type of financial data is to forecast the volatility of future returns. Models such as ARCH and GARCH models (Engle, 1982; Bollerslev, 1986) and stochastic volatility models (Harvey, Ruiz and Shephard, 1994) have been developed to handle these problems. We will discuss these models and the analysis of financial data in Chapters 5 and 6. The R code for this example is similar to the previous examples:

```
plot(nyse, ylab="NYSE Returns")
```

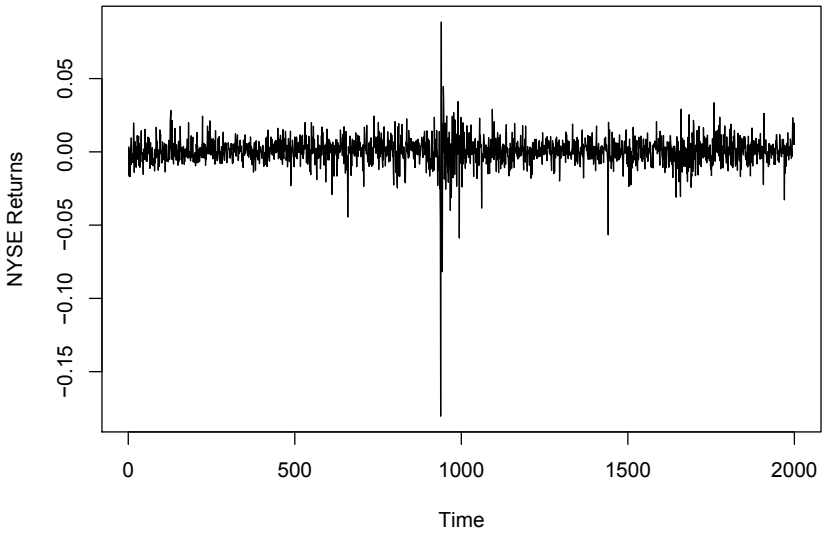

Fig. 1.4. Returns of the NYSE. The data are daily value weighted market returns from February 2, 1984 to December 31, 1991 (2000 trading days). The crash of October 19, 1987 occurs at $t = 938$.

Example 1.5 El Niño and Fish Population

We may also be interested in analyzing several time series at once. Figure 1.5 shows monthly values of an environmental series called the Southern Oscillation Index (SOI) and associated Recruitment (number of new fish) furnished by Dr. Roy Mendelssohn of the Pacific Environmental Fisheries Group (personal communication). Both series are for a period of 453 months ranging over the years 1950–1987. The 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 El Niño effect, which has been blamed, in particular, for the 1997 floods in the midwestern portions of the United States. Both series in Figure 1.5 tend to exhibit repetitive behavior, with regularly repeating cycles that are easily visible. This periodic behavior is of interest because underlying processes of interest may be regular and the rate or frequency of oscillation characterizing the behavior of the underlying series would help to identify them. One can also remark that the cycles of the SOI are repeating at a faster rate than those of the Recruitment series. The Recruitment series also shows several kinds of oscillations, a faster frequency that seems to repeat about every 12 months and a slower frequency that seems to repeat about every 50 months. The study of the kinds of cycles and their strengths is the subject of Chapter 4. The two series also tend to be somewhat related; it is easy to imagine that somehow the fish population is dependent on the SOI. Perhaps even a lagged relation exists, with the SOI signaling changes in the fish population. This possibility

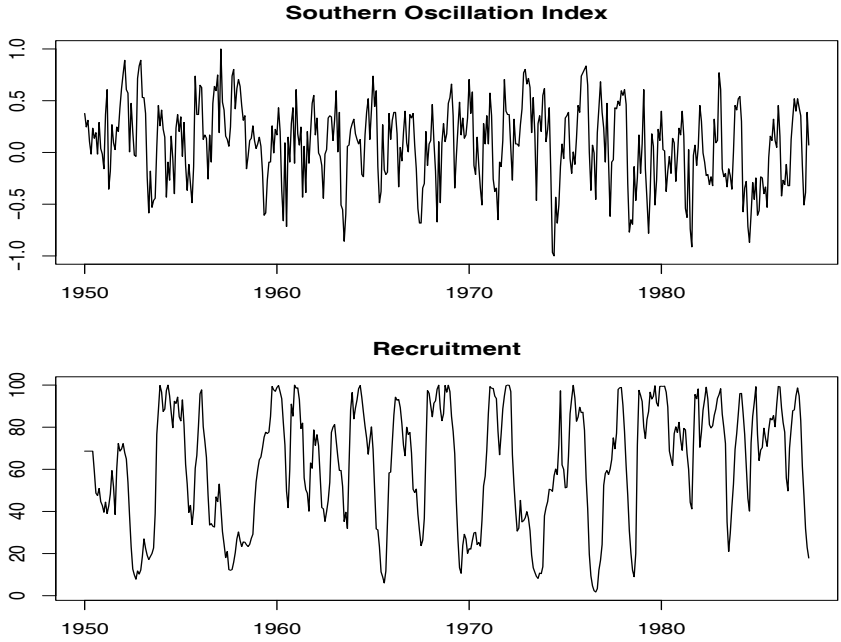

Fig. 1.5. Monthly SOI and Recruitment (estimated new fish), 1950-1987.

suggests trying some version of regression analysis as a procedure for relating the two series. Transfer function modeling, as considered in Chapter 5, can be applied in this case to obtain a model relating Recruitment to its own past and the past values of the SOI.

The following R code will reproduce Figure 1.5:

```
par(mfrow = c(2,1))  # set up the graphics
plot(soi, ylab="", xlab="", main="Southern Oscillation Index")
plot(rec, ylab="", xlab="", main="Recruitment")
```

Example 1.6 fMRI Imaging

A fundamental problem in classical statistics occurs when we are given a collection of independent series or vectors of series, generated under varying experimental conditions or treatment configurations. Such a set of series is shown in Figure 1.6, where we observe data collected from various locations in the brain via functional magnetic resonance imaging (fMRI). In this example, five subjects were given periodic brushing on the hand. The stimulus was applied for 32 seconds and then stopped for 32 seconds; thus, the signal period is 64 seconds. The sampling rate was one observation every 2 seconds for 256 seconds ($n = 128$). For this example, we averaged the results over subjects (these were evoked responses, and all subjects were in phase). The

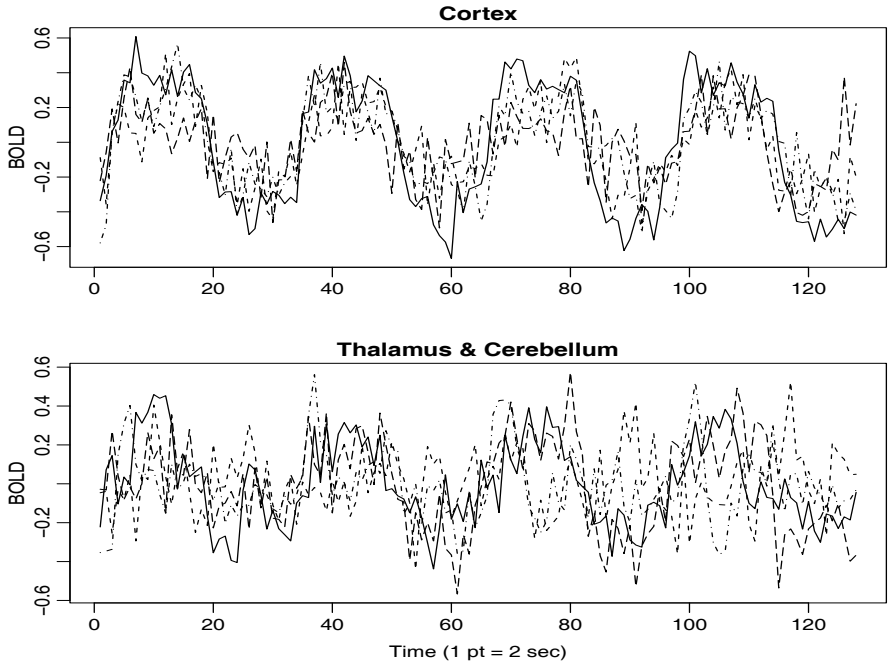

Fig. 1.6. fMRI data from various locations in the cortex, thalamus, and cerebellum; $n = 128$ points, one observation taken every 2 seconds.

series shown in Figure 1.6 are consecutive measures of blood oxygenation-level dependent (BOLD) signal intensity, which measures areas of activation in the brain. Notice that the periodicities appear strongly in the motor cortex series and less strongly in the thalamus and cerebellum. The fact that one has series from different areas of the brain suggests testing whether the areas are responding differently to the brush stimulus. Analysis of variance techniques accomplish this in classical statistics, and we show in Chapter 7 how these classical techniques extend to the time series case, leading to a spectral analysis of variance.

The following R commands were used to plot the data:

```
1 par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
2 ts.plot(fmri1[,2:5], lty=c(1,2,4,5), ylab="BOLD", xlab="",
      main="Cortex")
3 ts.plot(fmri1[,6:9], lty=c(1,2,4,5), ylab="BOLD", xlab="",
      main="Thalamus & Cerebellum")
4 mtext("Time (1 pt = 2 sec)", side=1, line=2)
```

Example 1.7 Earthquakes and Explosions

As a final example, the series in Figure 1.7 represent two phases or arrivals along the surface, denoted by P ($t = 1, \ldots, 1024$) and S ($t = 1025, \ldots, 2048$),

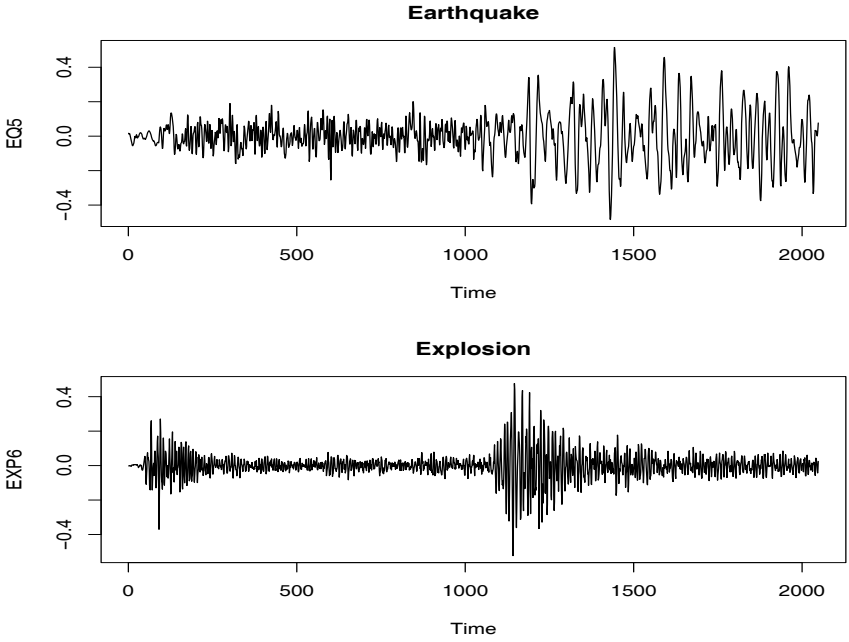

Fig. 1.7. Arrival phases from an earthquake (top) and explosion (bottom) at 40 points per second.

at a seismic recording station. The recording instruments in Scandinavia are observing earthquakes and mining explosions with one of each shown in Figure 1.7. The general problem of interest is in distinguishing or discriminating between waveforms generated by earthquakes and those generated by explosions. Features that may be important are the rough amplitude ratios of the first phase P to the second phase S, which tend to be smaller for earthquakes than for explosions. In the case of the two events in Figure 1.7, the ratio of maximum amplitudes appears to be somewhat less than .5 for the earthquake and about 1 for the explosion. Otherwise, note a subtle difference exists in the periodic nature of the S phase for the earthquake. We can again think about spectral analysis of variance for testing the equality of the periodic components of earthquakes and explosions. We would also like to be able to classify future P and S components from events of unknown origin, leading to the time series discriminant analysis developed in Chapter 7.

To plot the data as in this example, use the following commands in R:

```
par(mfrow=c(2,1))
plot(EQ5, main="Earthquake")
plot(EXP6, main="Explosion")
```

1.3 Time Series Statistical 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. In order to provide a statistical setting for describing the character of data that seemingly fluctuate in a random fashion over time, we assume a time series can be defined as a collection of random variables indexed according to the order they are obtained in time. For example, we may consider a time series as a sequence of random variables, $x_1, x_2, x_3, \ldots$, where the random variable x_1 denotes the value taken by the series at the first time point, the variable x_2 denotes the value for the second time period, x_3 denotes the value for the third time period, 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 $t = 0, \pm 1, \pm 2, ...$, or some subset of the integers. The observed values of a stochastic process are referred to as a realization of the stochastic process. Because it will be clear from the context of our discussions, we use the term time series whether we are referring generically to the process or to a particular realization and make no notational distinction between the two concepts.

It is conventional to display a sample time series graphically by plotting the values of the random variables on the vertical axis, or ordinate, with the time scale as the abscissa. It is usually convenient to connect the values at adjacent time periods to reconstruct visually some original hypothetical continuous time series that might have produced these values as a discrete sample. Many of the series discussed in the previous section, for example, could have been observed at any continuous point in time and are conceptually more properly treated as continuous time series. The approximation of these series by discrete time parameter series sampled at equally spaced points in time is simply an acknowledgment that sampled data will, for the most part, be discrete because of restrictions inherent in the method of collection. Furthermore, the analysis techniques are then feasible using computers, which are limited to digital computations. Theoretical developments also rest on the idea that a continuous parameter time series should be specified in terms of finite-dimensional distribution functions defined over a finite number of points in time. This is not to say that the selection of the sampling interval or rate is not an extremely important consideration. The appearance of data can be changed completely by adopting an insufficient sampling rate. We have all seen wagon wheels in movies appear to be turning backwards because of the insufficient number of frames sampled by the camera. This phenomenon leads to a distortion called aliasing (see §4.2).

The fundamental visual characteristic distinguishing the different series shown in Examples 1.1–1.7 is their differing degrees of smoothness. One possible explanation for this smoothness is that it is being induced by the supposition 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

model expresses a fundamental way in which we might think about generating realistic-looking time series. To begin to develop an approach to using collections of random variables to model time series, consider Example 1.8.

Example 1.8 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 and indicates that all possible periodic oscillations are present with equal strength.

We will, at times, also require the noise to be independent and identically distributed (iid) random variables with mean 0 and variance σ_w^2. We shall distinguish this case by saying white independent noise, or by writing $w_t \sim \text{iid}(0, \sigma_w^2)$. A particularly useful white noise series is Gaussian white noise, wherein the w_t are independent normal random variables, with mean 0 and variance σ_w^2; or more succinctly, $w_t \sim \text{iid N}(0, \sigma_w^2)$. Figure 1.8 shows in the upper panel a collection of 500 such random variables, with $\sigma_w^2 = 1$, plotted in the order in which they were drawn. The resulting series bears a slight resemblance to the explosion in Figure 1.7 but is not smooth enough to serve as a plausible model for any of the other experimental series. The plot tends to show visually a mixture of many different kinds of oscillations in the white noise series.

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 Examples 1.9 and 1.10.

Example 1.9 Moving Averages

We might replace the white noise series w_t by a moving average that smooths the series. For example, consider replacing w_t in Example 1.8 by an average of its current value and its immediate neighbors in the past and future. That is, let

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

which leads to the series shown in the lower panel of Figure 1.8. Inspecting the series shows a smoother version of the first series, reflecting the fact that the slower oscillations are more apparent and some of the faster oscillations are taken out. We begin to notice a similarity to the SOI in Figure 1.5, or perhaps, to some of the fMRI series in Figure 1.6.

To reproduce Figure 1.8 in R use the following commands. 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`.

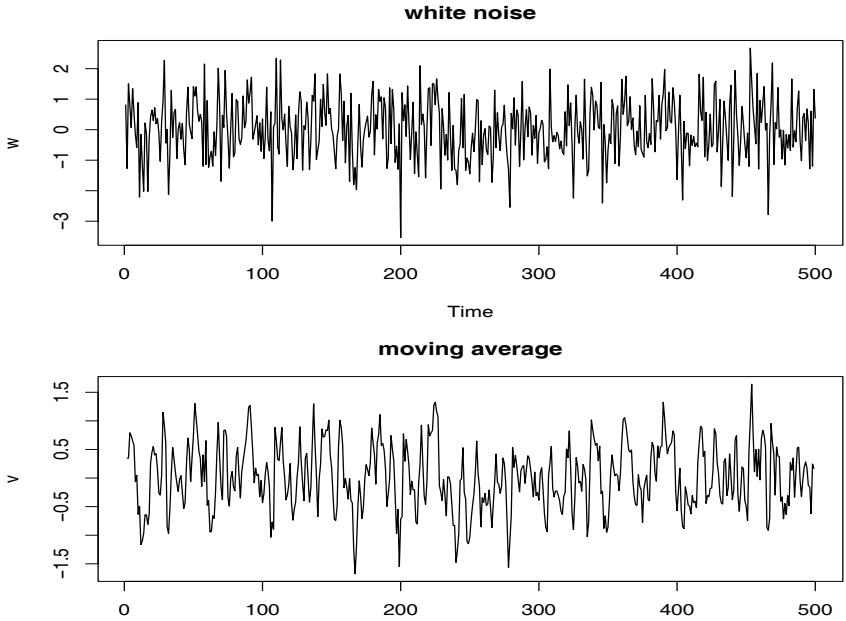

Fig. 1.8. Gaussian white noise series (top) and three-point moving average of the Gaussian white noise series (bottom).

```
w = rnorm(500,0,1)                  # 500 N(0,1) variates
v = filter(w, sides=2, rep(1/3,3))  # moving average
par(mfrow=c(2,1))
plot.ts(w, main="white noise")
plot.ts(v, main="moving average")
```

The speech series in Figure 1.3 and the Recruitment series in Figure 1.5, as well as some of the MRI series in Figure 1.6, differ from the moving average series because one particular kind of oscillatory behavior seems to predominate, producing a sinusoidal type of 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 3.

Example 1.10 Autoregressions

Suppose we consider the white noise series w_t of Example 1.8 as input and calculate the output using the second-order equation

$$x_t = x_{t-1} - .9x_{t-2} + w_t \tag{1.2}$$

successively for $t = 1, 2, \ldots, 500$. Equation (1.2) represents a regression or prediction of the current value x_t of a time series as a function of the past two values of the series, and, hence, the term autoregression is suggested

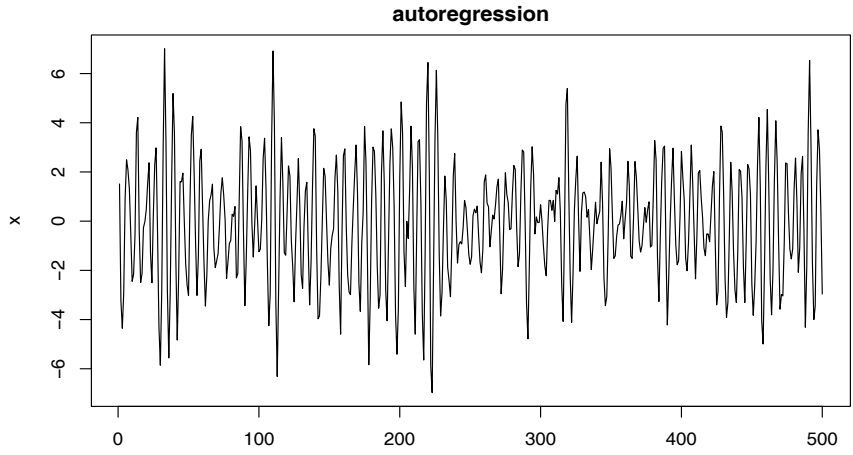

Fig. 1.9. Autoregressive series generated from model (1.2).

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 assume that we are given these values and generate the succeeding values by substituting into (1.2). The resulting output series is shown in Figure 1.9, and we note the periodic behavior of the series, which is similar to that displayed by the speech series in Figure 1.3. The autoregressive model above and its generalizations can be used as an underlying model for many observed series and will be studied in detail in Chapter 3.

One way to simulate and plot data from the model (1.2) in R is to use the following commands (another way is to use `arima.sim`).

```
w = rnorm(550,0,1)  # 50 extra to avoid startup problems
x = filter(w, filter=c(1,-.9), method="recursive")[-(1:50)]
plot.ts(x, main="autoregression")
```

Example 1.11 Random Walk with Drift

A model for analyzing trend such as seen in the global temperature data in Figure 1.2, is the random walk with drift model given by

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

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 $\delta = 0$, (1.3) is called simply a random walk. The term random walk comes from the fact that, when $\delta = 0$, 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,

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

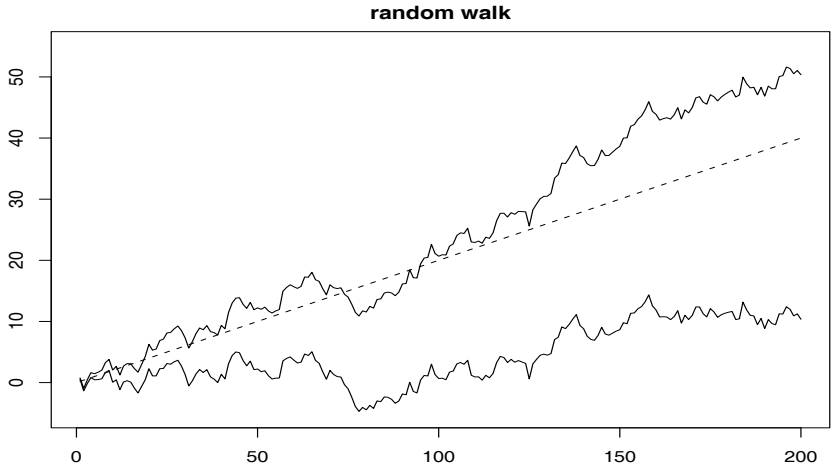

Fig. 1.10. Random walk, $\sigma_w = 1$, with drift $\delta = .2$ (upper jagged line), without drift, $\delta = 0$ (lower jagged line), and a straight line with slope .2 (dashed line).

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 .2, and with $\sigma_w = 1$. For comparison, we also superimposed the straight line $.2t$ 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(154)                 # so you can reproduce the results
w = rnorm(200,0,1);  x = cumsum(w)   # two commands in one line
wd = w +.2;   xd = cumsum(wd)
plot.ts(xd, ylim=c(-5,55), main="random walk")
lines(x); lines(.2*(1:200), lty="dashed")
```

Example 1.12 Signal in Noise

Many realistic models for generating time series assume an underlying signal with some consistent periodic variation, contaminated by adding a random noise. For example, it is easy to detect the regular cycle fMRI series displayed on the top of Figure 1.6. Consider the model

$$x_t = 2\cos(2\pi t/50 + .6\pi) + w_t \tag{1.5}$$

for $t = 1, 2, \ldots, 500$, where the first term is regarded as the signal, shown in the 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 $\phi = .6\pi$.

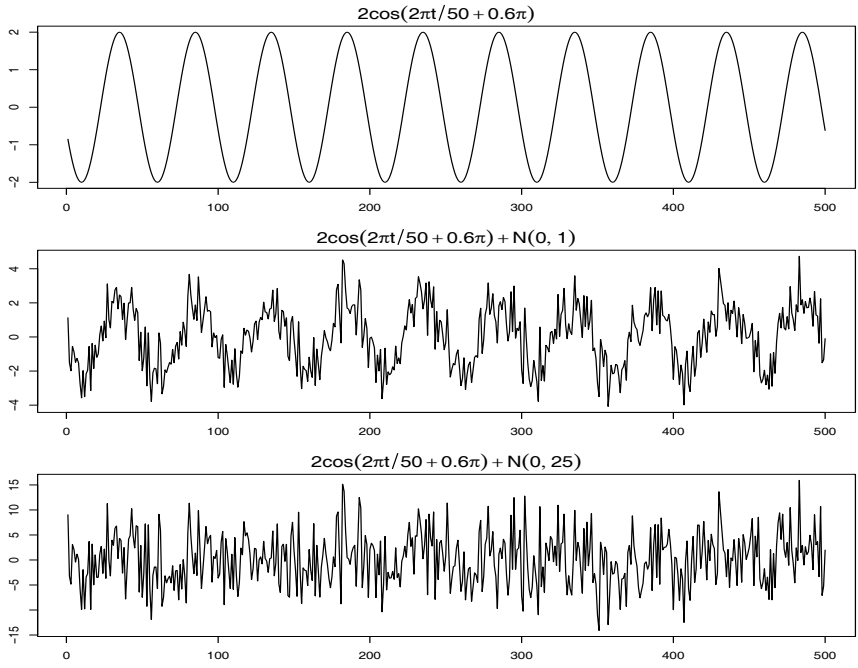

Fig. 1.11. Cosine wave with period 50 points (top panel) compared with the cosine wave contaminated with additive white Gaussian noise, $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel); see (1.5).

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. Of course, the degree to which the signal is obscured depends on the amplitude of the signal and 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 of Figure 1.11, 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:

```
cs = 2*cos(2*pi*1:500/50 + .6*pi)
w = rnorm(500,0,1)
par(mfrow=c(3,1), mar=c(3,2,2,1), cex.main=1.5)
plot.ts(cs, main=expression(2*cos(2*pi*t/50+.6*pi)))
plot.ts(cs+w, main=expression(2*cos(2*pi*t/50+.6*pi) + N(0,1)))
plot.ts(cs+5*w, main=expression(2*cos(2*pi*t/50+.6*pi) + N(0,25)))
```

In Chapter 4, we will study the use of spectral analysis as a possible technique for detecting regular or periodic signals, such as the one described

in Example 1.12. In general, we would emphasize the importance of simple additive models such as given above in the form

$$x_t = s_t + v_t, \tag{1.7}$$

where s_t denotes some unknown signal and v_t denotes a time series that may be white or correlated over time. The problems of detecting a signal and then in estimating or extracting the waveform of s_t are of great interest in many areas of engineering and the physical and biological sciences. In economics, the underlying signal may be a trend or it may be a seasonal component of a series. Models such as (1.7), where the signal has an autoregressive structure, form the motivation for the state-space model of Chapter 6.

In the above examples, we have tried to motivate the use of various combinations of random variables emulating real time series data. Smoothness characteristics of observed time series were introduced by combining the random variables in various ways. Averaging independent random variables over adjacent time points, as in Example 1.9, or looking at the output of difference equations that respond to white noise inputs, as in Example 1.10, are common ways of generating correlated data. In the next section, we introduce various theoretical measures used for describing how time series behave. As is usual in statistics, the complete description involves the multivariate distribution function of the jointly sampled values $x_1, x_2, \ldots, x_n$, whereas more economical descriptions can be had in terms of the mean and autocorrelation functions. Because correlation is an essential feature of time series analysis, the most useful descriptive measures are those expressed in terms of covariance and correlation functions.

1.4 Measures of Dependence: Autocorrelation and Cross-Correlation

A complete description of a time series, observed as a collection of n random variables at arbitrary integer time points $t_1, t_2, \ldots, t_n$, for any positive integer n, is provided by the joint distribution function, evaluated as the probability that the values of the series are jointly less than the n constants, $c_1, c_2, \ldots, c_n$; i.e.,

$$F(c_1, c_2, \ldots, c_n) = P\big(x_{t_1} \leq c_1, x_{t_2} \leq c_2, \ldots, x_{t_n} \leq c_n\big). \tag{1.8}$$

Unfortunately, the multidimensional distribution function cannot usually be written easily unless the random variables are jointly normal, in which case the joint density has the well-known form displayed in (1.31).

Although the joint distribution function describes the data completely, it is an unwieldy tool for displaying and analyzing time series data. The distribution function (1.8) must be evaluated as a function of n arguments, so any plotting of the corresponding multivariate density functions is virtually impossible. The marginal distribution functions

$$F_t(x) = P\{x_t \leq x\}$$

or the corresponding marginal density functions

$$f_t(x) = \frac{\partial F_t(x)}{\partial x},$$

when they exist, are often informative for examining the marginal behavior of a series.[2] Another informative marginal descriptive measure is the mean function.

Definition 1.1 *The* **mean function** *is defined as*

$$\mu_{xt} = E(x_t) = \int_{-\infty}^{\infty} x f_t(x)\, dx, \tag{1.9}$$

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 1.13 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 1.9 does not change the mean because we can write

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

Example 1.14 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.

[2] If x_t is Gaussian with mean μ_t and variance σ_t^2, abbreviated as $x_t \sim \mathrm{N}(\mu_t, \sigma_t^2)$, the marginal density is given by $f_t(x) = \dfrac{1}{\sigma_t\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma_t^2}(x-\mu_t)^2\right\}$.

Example 1.15 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

$$\begin{aligned}\mu_{xt} &= E(x_t) = E\big[2\cos(2\pi t/50 + .6\pi) + w_t\big] \\ &= 2\cos(2\pi t/50 + .6\pi) + E(w_t) \\ &= 2\cos(2\pi t/50 + .6\pi),\end{aligned}$$

and the mean function is just the cosine wave.

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 1.2 *The* **autocovariance function** *is defined as the second moment product*

$$\gamma_x(s,t) = \operatorname{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)], \tag{1.10}$$

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. Very smooth series exhibit autocovariance functions that stay large even when the t and s are far apart, whereas choppy series tend to have autocovariance functions that are nearly zero for large separations. The autocovariance (1.10) is the average cross-product relative to the joint distribution $F(x_s, x_t)$. Recall from classical statistics that if $\gamma_x(s,t) = 0$, 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

$$\gamma_x(t,t) = E[(x_t - \mu_t)^2] = \operatorname{var}(x_t). \tag{1.11}$$

Example 1.16 Autocovariance of White Noise

The white noise series w_t has $E(w_t) = 0$ and

$$\gamma_w(s,t) = \operatorname{cov}(w_s, w_t) = \begin{cases} \sigma_w^2 & s = t, \\ 0 & s \neq t. \end{cases} \tag{1.12}$$

A realization of white noise with $\sigma_w^2 = 1$ is shown in the top panel of Figure 1.8.

Example 1.17 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 1.9. In this case,

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

When $s = t$ we have[3]

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

When $s = t + 1$,

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

using (1.12). 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

$$\gamma_v(s,t) = \begin{cases} \frac{3}{9}\sigma_w^2 & s = t, \\ \frac{2}{9}\sigma_w^2 & |s-t| = 1, \\ \frac{1}{9}\sigma_w^2 & |s-t| = 2, \\ 0 & |s-t| > 2. \end{cases} \tag{1.13}$$

Example 1.17 shows clearly that the smoothing operation introduces a covariance function that decreases as the separation between the two time points increases and disappears completely when the time points are separated by three or more time points. This particular autocovariance is interesting because it only depends on the time separation or lag and not on the absolute location of the points along the series. We shall see later that this dependence suggests a mathematical model for the concept of weak stationarity.

Example 1.18 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. Note that, as opposed to the previous examples, the autocovariance function of a random walk

[3] If the random variables $U = \sum_{j=1}^{m} a_j X_j$ and $V = \sum_{k=1}^{r} b_k Y_k$ are linear combinations of random variables $\{X_j\}$ and $\{Y_k\}$, respectively, then $\text{cov}(U,V) = \sum_{j=1}^{m}\sum_{k=1}^{r} a_j b_k \text{cov}(X_j, Y_k)$. Furthermore, $\text{var}(U) = \text{cov}(U,U)$.

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 .2 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 1.3 *The* **autocorrelation function (ACF)** *is defined as*

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}}. \tag{1.14}$$

The ACF measures the linear predictability of the series at time t, say x_t, using only the value x_s. We can show easily that $-1 \le \rho(s,t) \le 1$ using the Cauchy–Schwarz inequality.[4] 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 1.4 *The* **cross-covariance function** *between two series, x_t and y_t, is*

$$\gamma_{xy}(s,t) = \text{cov}(x_s, y_t) = E[(x_s - \mu_{xs})(y_t - \mu_{yt})]. \tag{1.15}$$

There is also a scaled version of the cross-covariance function.

Definition 1.5 *The* **cross-correlation function (CCF)** *is given by*

$$\rho_{xy}(s,t) = \frac{\gamma_{xy}(s,t)}{\sqrt{\gamma_x(s,s)\gamma_y(t,t)}}. \tag{1.16}$$

We may easily extend the above ideas to the case of more than two series, say, $x_{t1}, x_{t2}, \ldots, x_{tr}$; that is, multivariate time series with r components. For example, the extension of (1.10) in this case is

$$\gamma_{jk}(s,t) = E[(x_{sj} - \mu_{sj})(x_{tk} - \mu_{tk})] \qquad j,k = 1,2,\ldots,r. \tag{1.17}$$

In the definitions above, the autocovariance and cross-covariance functions may change as one moves along the series because the values depend on both s

[4] The Cauchy–Schwarz inequality implies $|\gamma(s,t)|^2 \le \gamma(s,s)\gamma(t,t)$.

and t, the locations of the points in time. In Example 1.17, the autocovariance function depends on the separation of x_s and x_t, say, $h = |s - t|$, and not on where the points are located in time. As long as the points are separated by h units, the location of the two points does not matter. This notion, called weak stationarity, when the mean is constant, is fundamental in allowing us to analyze sample time series data when only a single series is available.

1.5 Stationary Time Series

The preceding definitions of the mean and autocovariance functions are completely general. Although we have not made any special assumptions about the behavior of the time series, many of the preceding examples have hinted that a sort of regularity may exist over time in the behavior of a time series. We introduce the notion of regularity using a concept called stationarity.

Definition 1.6 *A* **strictly stationary** *time series is one for which the probabilistic behavior of every collection of values*

$$\{x_{t_1}, x_{t_2}, \ldots, x_{t_k}\}$$

is identical to that of the time shifted set

$$\{x_{t_1+h}, x_{t_2+h}, \ldots, x_{t_k+h}\}.$$

That is,

$$P\{x_{t_1} \leq c_1, \ldots, x_{t_k} \leq c_k\} = P\{x_{t_1+h} \leq c_1, \ldots, x_{t_k+h} \leq c_k\} \tag{1.18}$$

for all $k = 1, 2, ...$, all time points $t_1, t_2, \ldots, t_k$, all numbers $c_1, c_2, \ldots, c_k$, and all time shifts $h = 0, \pm 1, \pm 2, ...$.

If a time series is strictly stationary, then all of the multivariate distribution functions for subsets of variables must agree with their counterparts in the shifted set for all values of the shift parameter h. For example, when $k = 1$, (1.18) implies that

$$P\{x_s \leq c\} = P\{x_t \leq c\} \tag{1.19}$$

for any time points s and t. This statement implies, for example, that the probability that the value of a time series sampled hourly is negative at 1 AM is the same as at 10 AM. In addition, if the mean function, μ_t, of the series x_t exists, (1.19) implies that $\mu_s = \mu_t$ for all s and t, and hence μ_t must be constant. Note, for example, that a random walk process with drift is *not* strictly stationary because its mean function changes with time; see Example 1.14 on page 18.

When $k = 2$, we can write (1.18) as

$$P\{x_s \leq c_1, x_t \leq c_2\} = P\{x_{s+h} \leq c_1, x_{t+h} \leq c_2\} \tag{1.20}$$

for any time points s and t and shift h. Thus, if the variance function of the process exists, (1.20) implies that the autocovariance function of the series x_t satisfies

$$\gamma(s,t) = \gamma(s+h, t+h)$$

for all s and t and h. We may interpret this result by saying the autocovariance function of the process depends only on the time difference between s and t, and not on the actual times.

The version of stationarity in Definition 1.6 is too strong for most applications. Moreover, it is difficult to assess strict stationarity from a single data set. Rather than imposing conditions on all possible distributions of a time series, we will use a milder version that imposes conditions only on the first two moments of the series. We now have the following definition.

Definition 1.7 *A* **weakly stationary** *time series, x_t, is a finite variance process such that*

(i) the mean value function, μ_t, defined in (1.9) is constant and does not depend on time t, and
(ii) the autocovariance function, $\gamma(s,t)$, defined in (1.10) depends on s and t only through their difference $|s-t|$.

Henceforth, we will use the term **stationary** *to mean weakly stationary; if a process is stationary in the strict sense, we will use the term strictly stationary.*

It should be clear from the discussion of strict stationarity following Definition 1.6 that a strictly stationary, finite variance, time series is also stationary. The converse is not true unless there are further conditions. One important case where stationarity implies strict stationarity is if the time series is Gaussian [meaning all finite distributions, (1.18), of the series are Gaussian]. We will make this concept more precise at the end of this section.

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

$$\mu_t = \mu. \tag{1.21}$$

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

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

because the time difference between times $t+h$ and t is the same as the time difference between times 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 $\gamma(h,0)$.

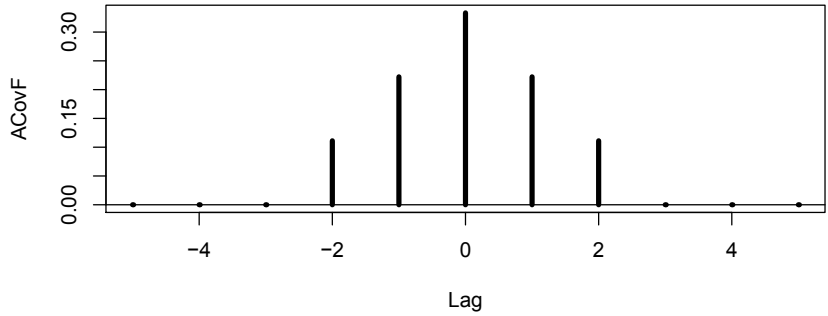

Fig. 1.12. Autocovariance function of a three-point moving average.

Definition 1.8 *The* **autocovariance function of a stationary time series** *will be written as*

$$\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) = E[(x_{t+h} - \mu)(x_t - \mu)]. \tag{1.22}$$

Definition 1.9 *The* **autocorrelation function (ACF) of a stationary time series** *will be written using (1.14) as*

$$\rho(h) = \frac{\gamma(t+h,t)}{\sqrt{\gamma(t+h,t+h)\gamma(t,t)}} = \frac{\gamma(h)}{\gamma(0)}. \tag{1.23}$$

The Cauchy–Schwarz inequality shows again that $-1 \leq \rho(h) \leq 1$ 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 1.19 Stationarity of White Noise

The mean and autocovariance functions of the white noise series discussed in Examples 1.8 and 1.16 are easily evaluated as $\mu_{wt} = 0$ and

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

Thus, white noise satisfies the conditions of Definition 1.7 and is weakly stationary or stationary. If the white noise variates are also normally distributed or Gaussian, the series is also strictly stationary, as can be seen by evaluating (1.18) using the fact that the noise would also be iid.

Example 1.20 Stationarity of a Moving Average

The three-point moving average process of Example 1.9 is stationary because, from Examples 1.13 and 1.17, 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 1.7. Figure 1.12 shows a plot of the autocovariance as a function of lag h with $\sigma_w^2 = 1$. Interestingly, the autocovariance function is symmetric about lag zero and decays as a function of lag.

The autocovariance function of a stationary process has several useful properties (also, see Problem 1.25). First, the value at $h = 0$, namely

$$\gamma(0) = E[(x_t - \mu)^2] \tag{1.24}$$

is the variance of the time series; note that the Cauchy–Schwarz inequality implies

$$|\gamma(h)| \le \gamma(0).$$

A final useful property, noted in the previous example, is that the autocovariance function of a stationary series is symmetric around the origin; that is,

$$\gamma(h) = \gamma(-h) \tag{1.25}$$

for all h. This property follows because shifting the series by h means that

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

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

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

Definition 1.10 *Two time series, say, x_t and y_t, are said to be* **jointly stationary** *if they are each stationary, and the cross-covariance function*

$$\gamma_{xy}(h) = \mathrm{cov}(x_{t+h}, y_t) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)] \tag{1.26}$$

is a function only of lag h.

Definition 1.11 *The* **cross-correlation function (CCF)** *of jointly stationary time series x_t and y_t is defined as*

$$\rho_{xy}(h) = \frac{\gamma_{xy}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}}. \tag{1.27}$$

Again, 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 [i.e., typically $\rho_{xy}(h) \neq \rho_{xy}(-h)$]; however, it is the case that

$$\rho_{xy}(h) = \rho_{yx}(-h), \tag{1.28}$$

which can be shown by manipulations similar to those used to show (1.25).

Example 1.21 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,

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

and

$$y_t = w_t - w_{t-1},$$

where w_t are independent random variables with zero means and variance σ_w^2. It is easy to show that $\gamma_x(0) = \gamma_y(0) = 2\sigma_w^2$ and $\gamma_x(1) = \gamma_x(-1) = \sigma_w^2, \gamma_y(1) = \gamma_y(-1) = -\sigma_w^2$. Also,

$$\gamma_{xy}(1) = \mathrm{cov}(x_{t+1}, y_t) = \mathrm{cov}(w_{t+1} + w_t, w_t - w_{t-1}) = \sigma_w^2$$

because only one term is nonzero (recall footnote 3 on page 20). Similarly, $\gamma_{xy}(0) = 0, \gamma_{xy}(-1) = -\sigma_w^2$. We obtain, using (1.27),

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

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

Example 1.22 Prediction Using Cross-Correlation

As a simple example of cross-correlation, consider the problem of determining possible leading or lagging relations between two series x_t and y_t. If 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 $\ell < 0$. Hence, the analysis of leading and lagging relations might be important in predicting the value of y_t from x_t. Assuming, for convenience, that x_t and y_t have zero means, and the noise w_t is uncorrelated with the x_t series, the cross-covariance function can be computed as

$$\gamma_{yx}(h) = \mathrm{cov}(y_{t+h}, x_t) = \mathrm{cov}(Ax_{t+h-\ell} + w_{t+h}, x_t)$$
$$= \mathrm{cov}(Ax_{t+h-\ell}, x_t) = A\gamma_x(h-\ell).$$

The cross-covariance function will look like the autocovariance of the input series x_t, with a peak on the positive side if x_t leads y_t and a peak on the negative side if x_t lags y_t.

The concept of weak stationarity forms the basis for much of the analysis performed with time series. The fundamental properties of the mean and autocovariance functions (1.21) and (1.22) are satisfied by many theoretical models that appear to generate plausible sample realizations. In Examples 1.9 and 1.10, two series were generated that produced stationary looking realizations, and in Example 1.20, we showed that the series in Example 1.9 was, in fact, weakly stationary. Both examples are special cases of the so-called linear process.

Definition 1.12 *A* **linear process**, x_t, *is defined to be a linear combination of white noise variates* w_t, *and is given by*

$$x_t = \mu + \sum_{j=-\infty}^{\infty} \psi_j w_{t-j}, \qquad \sum_{j=-\infty}^{\infty} |\psi_j| < \infty. \tag{1.29}$$

For the linear process (see Problem 1.11), we may show that the autocovariance function is given by

$$\gamma(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \psi_{j+h}\psi_j \tag{1.30}$$

for $h \geq 0$; recall that $\gamma(-h) = \gamma(h)$. This method exhibits the autocovariance function of the process in terms of the lagged products of the coefficients. Note that, for Example 1.9, we have $\psi_0 = \psi_{-1} = \psi_1 = 1/3$ and the result in Example 1.20 comes out immediately. The autoregressive series in Example 1.10 can also be put in this form, as can the general autoregressive moving average processes considered in Chapter 3.

Finally, as previously mentioned, an important case in which a weakly stationary series is also strictly stationary is the normal or Gaussian series.

Definition 1.13 *A process,* $\{x_t\}$, *is said to be a* **Gaussian process** *if the* n*-dimensional vectors* $\boldsymbol{x} = (x_{t_1}, x_{t_2}, \ldots, x_{t_n})'$, *for every collection of time points* $t_1, t_2, \ldots, t_n$, *and every positive integer* n, *have a multivariate normal distribution.*

Defining the $n \times 1$ mean vector $E(\boldsymbol{x}) \equiv \boldsymbol{\mu} = (\mu_{t_1}, \mu_{t_2}, \ldots, \mu_{t_n})'$ and the $n \times n$ covariance matrix as $\mathrm{var}(\boldsymbol{x}) \equiv \Gamma = \{\gamma(t_i, t_j);\ i, j = 1, \ldots, n\}$, which is

assumed to be positive definite, the multivariate normal density function can be written as

$$f(\boldsymbol{x}) = (2\pi)^{-n/2}|\Gamma|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Gamma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}, \tag{1.31}$$

where $|\cdot|$ denotes the determinant. This distribution forms the basis for solving problems involving statistical inference for time series. If a Gaussian time series, $\{x_t\}$, is weakly stationary, then $\mu_t = \mu$ and $\gamma(t_i, t_j) = \gamma(|t_i - t_j|)$, so that the vector $\boldsymbol{\mu}$ and the matrix Γ are independent of time. These facts imply that all the finite distributions, (1.31), of the series $\{x_t\}$ depend only on time lag and not on the actual times, and hence the series must be strictly stationary.

1.6 Estimation of Correlation

Although the theoretical autocorrelation and cross-correlation functions are useful for describing the properties of certain hypothesized models, most of the analyses must be performed using sampled data. This limitation means the sampled points $x_1, x_2, \ldots, x_n$ only are available for estimating the mean, autocovariance, and autocorrelation functions. From the point of view of classical statistics, this poses a problem because we will typically not have iid copies of x_t that are available for estimating the covariance and correlation functions. In the usual situation with only one realization, however, the assumption of stationarity becomes critical. Somehow, we must use averages over this single realization to estimate the population means and covariance functions.

Accordingly, if a time series is stationary, the mean function (1.21) $\mu_t = \mu$ is constant so that we can estimate it by the sample mean,

$$\bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t. \tag{1.32}$$

The standard error of the estimate is the square root of $\mathrm{var}(\bar{x})$, which can be computed using first principles (recall footnote 3 on page 20), and is given by

$$\begin{aligned}
\mathrm{var}(\bar{x}) &= \mathrm{var}\left(\frac{1}{n}\sum_{t=1}^{n} x_t\right) = \frac{1}{n^2}\mathrm{cov}\left(\sum_{t=1}^{n} x_t, \sum_{s=1}^{n} x_s\right) \\
&= \frac{1}{n^2}\Big(n\gamma_x(0) + (n-1)\gamma_x(1) + (n-2)\gamma_x(2) + \cdots + \gamma_x(n-1) \\
&\qquad + (n-1)\gamma_x(-1) + (n-2)\gamma_x(-2) + \cdots + \gamma_x(1-n)\Big) \\
&= \frac{1}{n}\sum_{h=-n}^{n}\Big(1 - \frac{|h|}{n}\Big)\gamma_x(h).
\end{aligned} \tag{1.33}$$

If the process is white noise, (1.33) 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 1.19)

The theoretical autocovariance function, (1.22), is estimated by the sample autocovariance function defined as follows.

Definition 1.14 *The* **sample autocovariance function** *is defined as*

$$\widehat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(x_t - \bar{x}), \tag{1.34}$$

with $\widehat{\gamma}(-h) = \widehat{\gamma}(h)$ *for* $h = 0, 1, \ldots, n-1$.

The sum in (1.34) runs over a restricted range because x_{t+h} is not available for $t + h > n$. The estimator in (1.34) is preferred to the one that would be obtained by dividing by $n-h$ because (1.34) is a non-negative definite function. The autocovariance function, $\gamma(h)$, of a stationary process is non-negative definite (see Problem 1.25) ensuring that variances of linear combinations of the variates x_t will never be negative. And, because $\text{var}(a_1 x_{t_1} + \cdots + a_n x_{t_n})$ is never negative, the estimate of that variance should also be non-negative. The estimator in (1.34) guarantees this result, but no such guarantee exists if we divide by $n-h$; this is explored further in Problem 1.25. Note that neither dividing by n nor $n-h$ in (1.34) yields an unbiased estimator of $\gamma(h)$.

Definition 1.15 *The* **sample autocorrelation function** *is defined, analogously to (1.23), as*

$$\widehat{\rho}(h) = \frac{\widehat{\gamma}(h)}{\widehat{\gamma}(0)}. \tag{1.35}$$

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 1.1 Large-Sample Distribution of the ACF

Under general conditions,[5] *if* x_t *is white noise, then for* n *large, the sample ACF,* $\widehat{\rho}_x(h)$*, for* $h = 1, 2, \ldots, H$*, where* H *is fixed but arbitrary, is approximately normally distributed with zero mean and standard deviation given by*

$$\sigma_{\widehat{\rho}_x(h)} = \frac{1}{\sqrt{n}}. \tag{1.36}$$

[5] The general conditions are that x_t is iid with finite fourth moment. A sufficient condition for this to hold is that x_t is white Gaussian noise. Precise details are given in Theorem A.7 in Appendix A.

Based on the previous result, we obtain a rough method of assessing whether peaks in $\widehat{\rho}(h)$ are significant by determining whether the observed peak is outside the interval $\pm 2/\sqrt{n}$ (or plus/minus two standard errors); for a white noise sequence, approximately 95% of the sample ACFs should be within these limits. The applications of this property develop because many statistical modeling procedures depend on reducing a time series to a white noise series using various kinds of transformations. After such a procedure is applied, the plotted ACFs of the residuals should then lie roughly within the limits given above.

Definition 1.16 *The estimators for the cross-covariance function, $\gamma_{xy}(h)$, as given in (1.26) and the cross-correlation, $\rho_{xy}(h)$, in (1.27) are given, respectively, by the* **sample cross-covariance function**

$$\widehat{\gamma}_{xy}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(y_t - \bar{y}), \tag{1.37}$$

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

$$\widehat{\rho}_{xy}(h) = \frac{\widehat{\gamma}_{xy}(h)}{\sqrt{\widehat{\gamma}_x(0)\widehat{\gamma}_y(0)}}. \tag{1.38}$$

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 1.22 for the theoretical cross-covariance function. Because $-1 \leq \widehat{\rho}_{xy}(h) \leq 1$, the practical importance of peaks can be assessed by comparing their magnitudes with their theoretical maximum values. Furthermore, for x_t and y_t independent linear processes of the form (1.29), we have the following property.

Property 1.2 Large-Sample Distribution of Cross-Correlation Under Independence

The large sample distribution of $\widehat{\rho}_{xy}(h)$ is normal with mean zero and

$$\sigma_{\widehat{\rho}_{xy}} = \frac{1}{\sqrt{n}} \tag{1.39}$$

if at least one of the processes is independent white noise (see Theorem A.8 in Appendix A).

Example 1.23 A Simulated Time Series

To give an example of the procedure for calculating numerically the auto-covariance and cross-covariance functions, consider a contrived set of data

Table 1.1. Sample Realization of the Contrived Series y_t

t	1	2	3	4	5	6	7	8	9	10
Coin	H	H	T	H	T	T	T	H	T	H
x_t	1	1	−1	1	−1	−1	−1	1	−1	1
y_t	6.7	5.3	3.3	6.7	3.3	4.7	4.7	6.7	3.3	6.7
$y_t - \bar{y}$	1.56	.16	−1.84	1.56	−1.84	−.44	−.44	1.56	−1.84	1.56

generated by tossing a fair coin, letting $x_t = 1$ when a head is obtained and $x_t = -1$ when a tail is obtained. Construct y_t as

$$y_t = 5 + x_t - .7x_{t-1}. \tag{1.40}$$

Table 1.1 shows sample realizations of the appropriate processes with $x_0 = -1$ and $n = 10$.

The sample autocorrelation for the series y_t can be calculated using (1.34) and (1.35) for $h = 0, 1, 2, \ldots$. It is not necessary to calculate for negative values because of the symmetry. For example, for $h = 3$, the autocorrelation becomes the ratio of

$$\begin{aligned}\widehat{\gamma}_y(3) &= \tfrac{1}{10}\sum_{t=1}^{7}(y_{t+3} - \bar{y})(y_t - \bar{y}) \\ &= \tfrac{1}{10}\Big[(1.56)(1.56) + (-1.84)(.16) + (-.44)(-1.84) + (-.44)(1.56) \\ &\quad + (1.56)(-1.84) + (-1.84)(-.44) + (1.56)(-.44)\Big] = -.048\end{aligned}$$

to

$$\widehat{\gamma}_y(0) = \tfrac{1}{10}[(1.56)^2 + (.16)^2 + \cdots + (1.56)^2] = 2.030$$

so that

$$\widehat{\rho}_y(3) = \frac{-.048}{2.030} = -.024.$$

The theoretical ACF can be obtained from the model (1.40) using the fact that the mean of x_t is zero and the variance of x_t is one. It can be shown that

$$\rho_y(1) = \frac{-.7}{1 + .7^2} = -.47$$

and $\rho_y(h) = 0$ for $|h| > 1$ (Problem 1.24). Table 1.2 compares the theoretical ACF with sample ACFs for a realization where $n = 10$ and another realization where $n = 100$; we note the increased variability in the smaller size sample.

Table 1.2. Theoretical and Sample ACFs for $n = 10$ and $n = 100$

h	$\rho_y(h)$	$n = 10$ $\widehat{\rho}_y(h)$	$n = 100$ $\widehat{\rho}_y(h)$
0	1.00	1.00	1.00
±1	−.47	−.55	−.45
±2	.00	.17	−.12
±3	.00	−.02	.14
±4	.00	.15	.01
±5	.00	−.46	−.01

Example 1.24 ACF of a Speech Signal

Computing the sample ACF as in the previous example can be thought of as matching the time series h units in the future, say, x_{t+h} against itself, x_t. Figure 1.13 shows the ACF of the speech series of Figure 1.3. The original series appears to contain a sequence of repeating short signals. The ACF confirms this behavior, showing repeating peaks spaced at about 106-109 points. Autocorrelation functions of the short signals appear, spaced at the intervals mentioned above. The distance between the repeating signals is known as the pitch period and is a fundamental parameter of interest in systems that encode and decipher speech. Because the series is sampled at 10,000 points per second, the pitch period appears to be between .0106 and .0109 seconds.

To put the data into `speech` as a time series object (if it is not there already from Example 1.3) and compute the sample ACF in R, use

```
1 acf(speech, 250)
```

Example 1.25 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 1.5 (see Figure 1.5), we have considered simultaneous monthly readings of the SOI and the number of new fish (Recruitment) computed from a model. Figure 1.14 shows the 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. 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

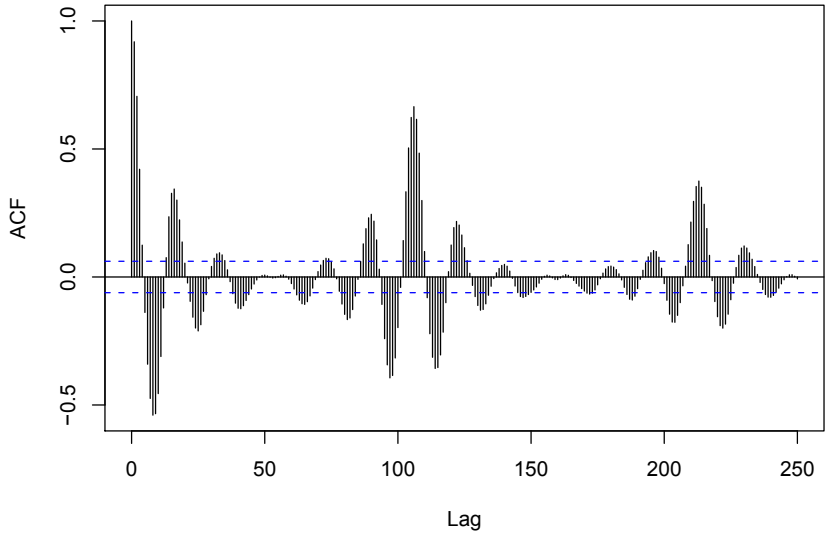

Fig. 1.13. ACF of the speech series.

six months. The sign of the ACF 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 [see (1.36) and (1.39)].

To reproduce Figure 1.14 in R, use the following commands:

```
par(mfrow=c(3,1))
acf(soi, 48, main="Southern Oscillation Index")
acf(rec, 48, main="Recruitment")
ccf(soi, rec, 48, main="SOI vs Recruitment", ylab="CCF")
```

1.7 Vector-Valued and Multidimensional Series

We frequently encounter situations in which the relationships between a number of jointly measured time series are of interest. For example, in the previous sections, we considered discovering the relationships between the SOI and Recruitment series. Hence, it will be useful to consider the notion of a vector time series $\boldsymbol{x}_t = (x_{t1}, x_{t2}, \ldots, x_{tp})'$, which contains as its components p univariate time series. We denote the $p \times 1$ column vector of the observed series as $\boldsymbol{x}_t$. The row vector $\boldsymbol{x}_t'$ is its transpose. For the stationary case, the $p \times 1$ mean vector

$$\boldsymbol{\mu} = E(\boldsymbol{x}_t) \tag{1.41}$$

of the form $\boldsymbol{\mu} = (\mu_{t1}, \mu_{t2}, \ldots, \mu_{tp})'$ and the $p \times p$ autocovariance matrix

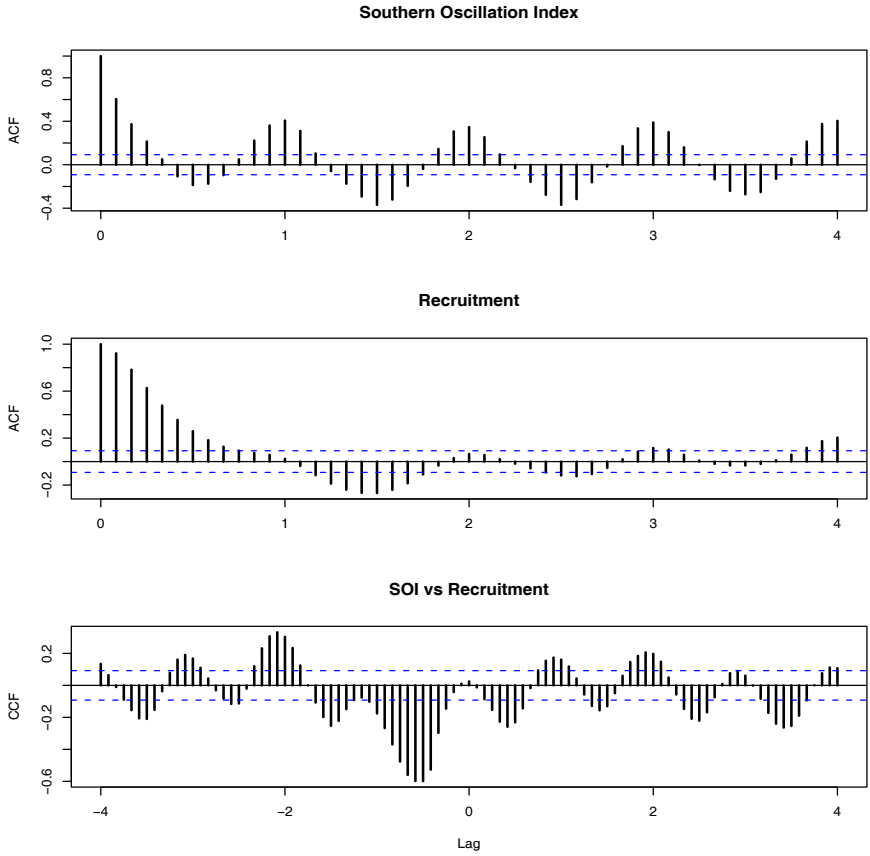

Fig. 1.14. Sample ACFs of the SOI series (top) and of the Recruitment series (middle), and the sample CCF of the two series (bottom); negative lags indicate SOI leads Recruitment. The lag axes are in terms of seasons (12 months).

$$\Gamma(h) = E[(\boldsymbol{x}_{t+h} - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})'] \tag{1.42}$$

can be defined, where the elements of the matrix $\Gamma(h)$ are the cross-covariance functions

$$\gamma_{ij}(h) = E[(x_{t+h,i} - \mu_i)(x_{tj} - \mu_j)] \tag{1.43}$$

for $i, j = 1, \ldots, p$. Because $\gamma_{ij}(h) = \gamma_{ji}(-h)$, it follows that

$$\Gamma(-h) = \Gamma'(h). \tag{1.44}$$

Now, the sample autocovariance matrix of the vector series $\boldsymbol{x}_t$ is the $p \times p$ matrix of sample cross-covariances, defined as

$$\widehat{\Gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (\boldsymbol{x}_{t+h} - \bar{\boldsymbol{x}})(\boldsymbol{x}_t - \bar{\boldsymbol{x}})', \tag{1.45}$$

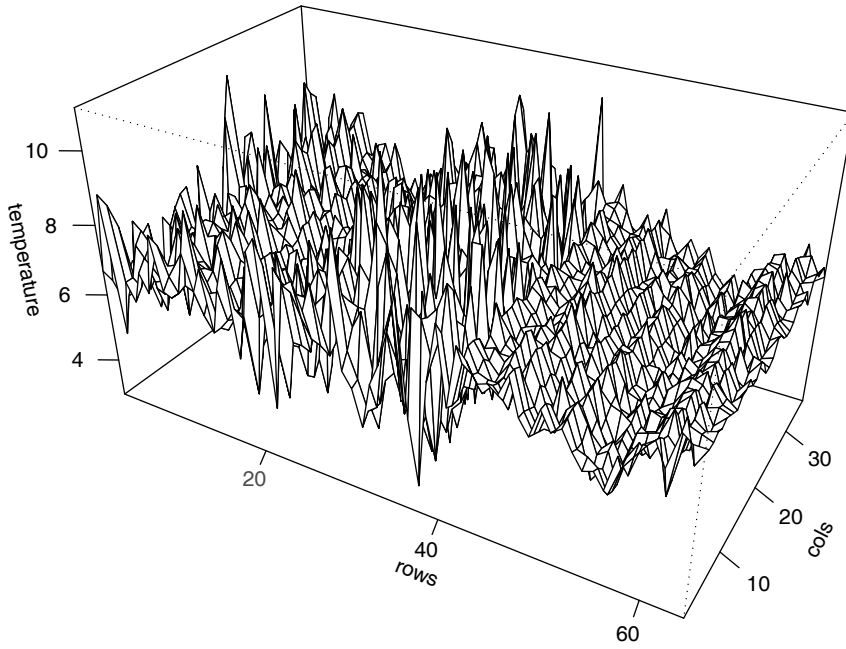

Fig. 1.15. Two-dimensional time series of temperature measurements taken on a rectangular field (64×36 with 17-foot spacing). Data are from Bazza et al. (1988).

where

$$\bar{\boldsymbol{x}} = n^{-1} \sum_{t=1}^{n} \boldsymbol{x}_t \tag{1.46}$$

denotes the $p \times 1$ sample mean vector. The symmetry property of the theoretical autocovariance (1.44) extends to the sample autocovariance (1.45), which is defined for negative values by taking

$$\widehat{\Gamma}(-h) = \widehat{\Gamma}(h)'. \tag{1.47}$$

In many applied problems, an observed series may be indexed by more than time alone. For example, the position in space of an experimental unit might be described by two coordinates, say, s_1 and s_2. We may proceed in these cases by defining a multidimensional process $x_{\boldsymbol{s}}$ as a function of the $r \times 1$ vector $\boldsymbol{s} = (s_1, s_2, \ldots, s_r)'$, where s_i denotes the coordinate of the ith index.

Example 1.26 Soil Surface Temperatures

As an example, the two-dimensional ($r = 2$) temperature series x_{s_1,s_2} in Figure 1.15 is indexed by a row number s_1 and a column number s_2 that

represent positions on a 64×36 spatial grid set out on an agricultural field. The value of the temperature measured at row s_1 and column s_2, is denoted by $x_{\boldsymbol{s}} = x_{s1,s2}$. We can note from the two-dimensional plot that a distinct change occurs in the character of the two-dimensional surface starting at about row 40, where the oscillations along the row axis become fairly stable and periodic. For example, averaging over the 36 columns, we may compute an average value for each s_1 as in Figure 1.16. It is clear that the noise present in the first part of the two-dimensional series is nicely averaged out, and we see a clear and consistent temperature signal.

To generate Figures 1.15 and 1.16 in R, use the following commands:

```
persp(1:64, 1:36, soiltemp, phi=30, theta=30, scale=FALSE, expand=4,
    ticktype="detailed", xlab="rows", ylab="cols",
    zlab="temperature")
plot.ts(rowMeans(soiltemp), xlab="row", ylab="Average Temperature")
```

The autocovariance function of a stationary multidimensional process, $x_{\boldsymbol{s}}$, can be defined as a function of the multidimensional lag vector, say, $\boldsymbol{h} = (h_1, h_2, \ldots, h_r)'$, as

$$\gamma(\boldsymbol{h}) = E[(x_{\boldsymbol{s}+\boldsymbol{h}} - \mu)(x_{\boldsymbol{s}} - \mu)], \tag{1.48}$$

where

$$\mu = E(x_{\boldsymbol{s}}) \tag{1.49}$$

does not depend on the spatial coordinate $\boldsymbol{s}$. For the two dimensional temperature process, (1.48) becomes

$$\gamma(h_1, h_2) = E[(x_{s_1+h_1,s_2+h_2} - \mu)(x_{s_1,s_2} - \mu)], \tag{1.50}$$

which is a function of lag, both in the row (h_1) and column (h_2) directions.

The multidimensional sample autocovariance function is defined as

$$\widehat{\gamma}(\boldsymbol{h}) = (S_1 S_2 \cdots S_r)^{-1} \sum_{s_1} \sum_{s_2} \cdots \sum_{s_r} (x_{\boldsymbol{s}+\boldsymbol{h}} - \bar{x})(x_{\boldsymbol{s}} - \bar{x}), \tag{1.51}$$

where $\boldsymbol{s} = (s_1, s_2, \ldots, s_r)'$ and the range of summation for each argument is $1 \le s_i \le S_i - h_i$, for $i = 1, \ldots, r$. The mean is computed over the r-dimensional array, that is,

$$\bar{x} = (S_1 S_2 \cdots S_r)^{-1} \sum_{s_1} \sum_{s_2} \cdots \sum_{s_r} x_{s_1,s_2,\cdots,s_r}, \tag{1.52}$$

where the arguments s_i are summed over $1 \le s_i \le S_i$. The multidimensional sample autocorrelation function follows, as usual, by taking the scaled ratio

$$\widehat{\rho}(\boldsymbol{h}) = \frac{\widehat{\gamma}(\boldsymbol{h})}{\widehat{\gamma}(\boldsymbol{0})}. \tag{1.53}$$

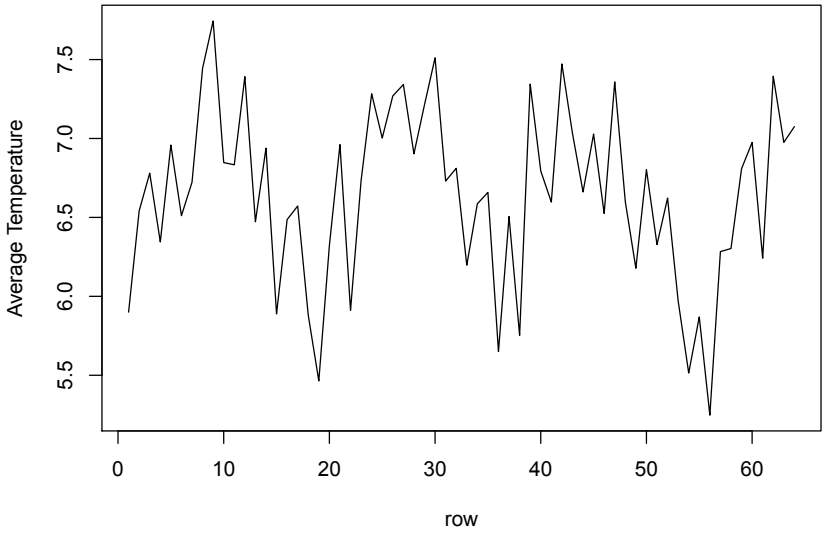

Fig. 1.16. Row averages of the two-dimensional soil temperature profile. $\bar{x}_{s_1} = \sum_{s_2} x_{s_1,s_2}/36$.

Example 1.27 Sample ACF of the Soil Temperature Series

The autocorrelation function of the two-dimensional (2d) temperature process can be written in the form

$$\widehat{\rho}(h_1, h_2) = \frac{\widehat{\gamma}(h_1, h_2)}{\widehat{\gamma}(0,0)},$$

where

$$\widehat{\gamma}(h_1, h_2) = (S_1 S_2)^{-1} \sum_{s_1} \sum_{s_2} (x_{s_1+h_1, s_2+h_2} - \bar{x})(x_{s_1,s_2} - \bar{x})$$

Figure 1.17 shows the autocorrelation function for the temperature data, and we note the systematic periodic variation that appears along the rows. The autocovariance over columns seems to be strongest for $h_1 = 0$, implying columns may form replicates of some underlying process that has a periodicity over the rows. This idea can be investigated by examining the mean series over columns as shown in Figure 1.16.

The easiest way (that we know of) to calculate a 2d ACF in R is by using the fast Fourier transform (FFT) as shown below. Unfortunately, the material needed to understand this approach is given in Chapter 4, §4.4. The 2d autocovariance function is obtained in two steps and is contained in `cs` below; $\widehat{\gamma}(0,0)$ is the (1,1) element so that $\widehat{\rho}(h_1, h_2)$ is obtained by dividing each element by that value. The 2d ACF is contained in `rs` below, and the rest of the code is simply to arrange the results to yield a nice display.

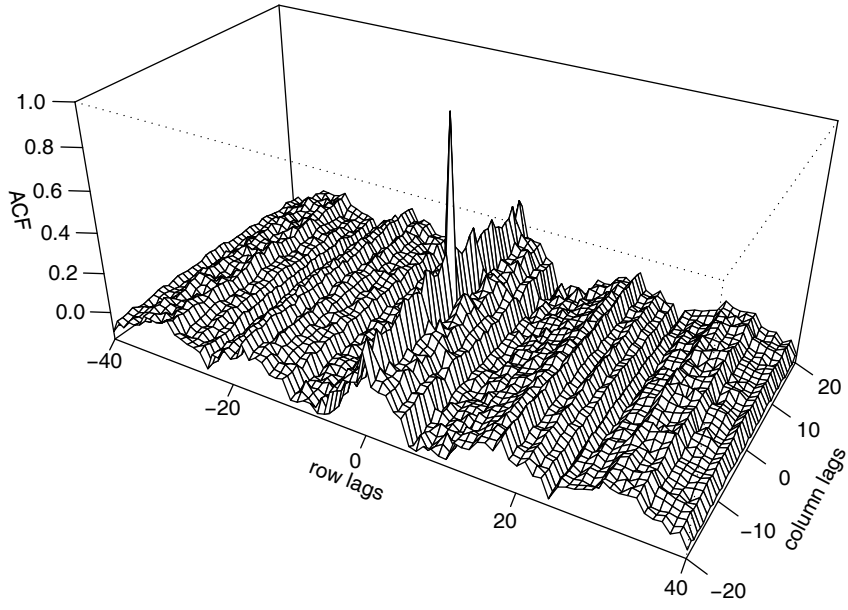

Fig. 1.17. Two-dimensional autocorrelation function for the soil temperature data.

```
fs = abs(fft(soiltemp-mean(soiltemp)))^2/(64*36)
cs = Re(fft(fs, inverse=TRUE)/sqrt(64*36))  # ACovF
rs = cs/cs[1,1]  # ACF
rs2 = cbind(rs[1:41,21:2], rs[1:41,1:21])
rs3 = rbind(rs2[41:2,], rs2)
par(mar = c(1,2.5,0,0)+.1)
persp(-40:40, -20:20, rs3, phi=30, theta=30, expand=30,
      scale="FALSE", ticktype="detailed", xlab="row lags",
      ylab="column lags", zlab="ACF")
```

The sampling requirements for multidimensional processes are rather severe because values must be available over some uniform grid in order to compute the ACF. In some areas of application, such as in soil science, we may prefer to sample a limited number of rows or *transects* and hope these are essentially replicates of the basic underlying phenomenon of interest. One-dimensional methods can then be applied. When observations are irregular in time space, modifications to the estimators need to be made. Systematic approaches to the problems introduced by irregularly spaced observations have been developed by Journel and Huijbregts (1978) or Cressie (1993). We shall not pursue such methods in detail here, but it is worth noting that the introduction of the variogram

$$2V_x(\boldsymbol{h}) = \mathrm{var}\{x_{\boldsymbol{s}+\boldsymbol{h}} - x_{\boldsymbol{s}}\} \tag{1.54}$$

and its sample estimator

$$2\widehat{V}_x(\boldsymbol{h}) = \frac{1}{N(\boldsymbol{h})}\sum_{\boldsymbol{s}}(x_{\boldsymbol{s}+\boldsymbol{h}} - x_{\boldsymbol{s}})^2 \tag{1.55}$$

play key roles, where $N(\boldsymbol{h})$ denotes both the number of points located within $\boldsymbol{h}$, and the sum runs over the points in the neighborhood. Clearly, substantial indexing difficulties will develop from estimators of the kind, and often it will be difficult to find non-negative definite estimators for the covariance function. Problem 1.27 investigates the relation between the variogram and the autocovariance function in the stationary case.

Problems

Section 1.2

1.1 To compare the earthquake and explosion signals, plot the data displayed in Figure 1.7 on the same graph using different colors or different line types and comment on the results. (The R code in Example 1.11 may be of help on how to add lines to existing plots.)

1.2 Consider a signal-plus-noise model of the general form $x_t = s_t + w_t$, where w_t is Gaussian white noise with $\sigma_w^2 = 1$. Simulate and plot $n = 200$ observations from each of the following two models (*Save the data or your code for use in Problem 1.22*):

(a) $x_t = s_t + w_t$, for $t = 1, ..., 200$, where

$$s_t = \begin{cases} 0, & t = 1, \ldots, 100 \\ 10\exp\{-\frac{(t-100)}{20}\}\cos(2\pi t/4), & t = 101, \ldots, 200. \end{cases}$$

Hint:

```
s = c(rep(0,100), 10*exp(-(1:100)/20)*cos(2*pi*1:100/4))
x = ts(s + rnorm(200, 0, 1))
plot(x)
```

(b) $x_t = s_t + w_t$, for $t = 1, \ldots, 200$, where

$$s_t = \begin{cases} 0, & t = 1, \ldots, 100 \\ 10\exp\{-\frac{(t-100)}{200}\}\cos(2\pi t/4), & t = 101, \ldots, 200. \end{cases}$$

(c) Compare the general appearance of the series (a) and (b) with the earthquake series and the explosion series shown in Figure 1.7. In addition, plot (or sketch) and compare the signal modulators (a) $\exp\{-t/20\}$ and (b) $\exp\{-t/200\}$, for $t = 1, 2, \ldots, 100$.

Section 1.3

1.3 (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 1.10, page 13. 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. Comment on the behavior of x_t and how applying the moving average filter changes that behavior. [Hints: Use `v = filter(x, rep(1/4, 4), sides = 1)` for the filter and note that the R code in Example 1.11 may be of help on how to add lines to existing plots.]

(b) Repeat (a) but with

$$x_t = \cos(2\pi t/4).$$

(c) Repeat (b) but with added N(0, 1) noise,

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

(d) Compare and contrast (a)–(c).

Section 1.4

1.4 Show that the autocovariance function can be written as

$$\gamma(s,t) = E[(x_s - \mu_s)(x_t - \mu_t)] = E(x_s x_t) - \mu_s \mu_t,$$

where $E[x_t] = \mu_t$.

1.5 For the two series, x_t, in Problem 1.2 (a) and (b):

(a) Compute and plot the mean functions $\mu_x(t)$, for $t = 1, \ldots, 200$.
(b) Calculate the autocovariance functions, $\gamma_x(s,t)$, for $s, t = 1, \ldots, 200$.

Section 1.5

1.6 Consider the time series

$$x_t = \beta_1 + \beta_2 t + w_t,$$

where β_1 and β_2 are known constants 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 moving average

$$v_t = \frac{1}{2q+1} \sum_{j=-q}^{q} x_{t-j}$$

is $\beta_1 + \beta_2 t$, and give a simplified expression for the autocovariance function.

1.7 For a moving average process of the form

$$x_t = w_{t-1} + 2w_t + w_{t+1},$$

where w_t are independent with zero means and variance σ_w^2, determine the autocovariance and autocorrelation functions as a function of lag $h = s - t$ and plot the ACF as a function of h.

1.8 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. (Hint: See Problem 1.6b.)

1.9 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

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

1.10 Suppose we would like to predict a single stationary series x_t with zero mean and autocorrelation function $\gamma(h)$ at some time in the future, say, $t + \ell$, for $\ell > 0$.

(a) If we predict using only x_t and some scale multiplier A, show that the mean-square prediction error

$$MSE(A) = E[(x_{t+\ell} - Ax_t)^2]$$

is minimized by the value

$$A = \rho(\ell).$$

(b) Show that the minimum mean-square prediction error is

$$MSE(A) = \gamma(0)[1 - \rho^2(\ell)].$$

(c) Show that if $x_{t+\ell} = Ax_t$, then $\rho(\ell) = 1$ if $A > 0$, and $\rho(\ell) = -1$ if $A < 0$.

1.11 Consider the linear process defined in (1.29).

(a) Verify that the autocovariance function of the process is given by (1.30). Use the result to verify your answer to Problem 1.7.
(b) Show that x_t exists as a limit in mean square (see Appendix A).

1.12 For two weakly stationary series x_t and y_t, verify (1.28).

1.13 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.

1.14 Let x_t be a stationary normal process with mean μ_x and autocovariance function $\gamma(h)$. Define the nonlinear time series

$$y_t = \exp\{x_t\}.$$

(a) Express the mean function $E(y_t)$ in terms of μ_x and $\gamma(0)$. The moment generating function of a normal random variable x with mean μ and variance σ^2 is

$$M_x(\lambda) = E[\exp\{\lambda x\}] = \exp\left\{\mu\lambda + \frac{1}{2}\sigma^2\lambda^2\right\}.$$

(b) Determine the autocovariance function of y_t. The sum of the two normal random variables $x_{t+h} + x_t$ is still a normal random variable.

1.15 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.

1.16 Consider the series

$$x_t = \sin(2\pi U t),$$

$t = 1, 2, \ldots$, where U has a uniform distribution on the interval $(0, 1)$.

(a) Prove x_t is weakly stationary.
(b) Prove x_t is not strictly stationary. [Hint: consider the joint bivariate cdf (1.18) at the points $t = 1, s = 2$ with $h = 1$, and find values of c_t, c_s where strict stationarity does not hold.]

1.17 Suppose we have the linear process x_t generated by

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

$t = 0, 1, 2, \ldots$, where $\{w_t\}$ is independent and identically distributed with characteristic function $\phi_w(\cdot)$, and θ is a fixed constant. [Replace "characteristic function" with "moment generating function" if instructed to do so.]

(a) Express the joint characteristic function of $x_1, x_2, \ldots, x_n$, say,

$$\phi_{x_1, x_2, \ldots, x_n}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

in terms of $\phi_w(\cdot)$.
(b) Deduce from (a) that x_t is strictly stationary.

1.18 Suppose that x_t is a linear process of the form (1.29). Prove

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty.$$

Section 1.6

1.19 Suppose $x_1, \ldots, x_n$ is a sample from the process $x_t = \mu + w_t - .8w_{t-1}$, where $w_t \sim wn(0, \sigma_w^2)$.

(a) Show that mean function is $E(x_t) = \mu$.
(b) Use (1.33) to calculate the standard error of $\bar{x}$ for estimating μ.
(c) Compare (b) to the case where x_t is white noise and show that (b) is smaller. Explain the result.

1.20 (a) Simulate a series of $n = 500$ Gaussian white noise 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 1.19.]
(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

1.21 (a) Simulate a series of $n = 500$ moving average observations as in Example 1.9 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 1.20.]
(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

1.22 Although the model in Problem 1.2(a) is not stationary (Why?), the sample ACF can be informative. For the data you generated in that problem, calculate and plot the sample ACF, and then comment.

1.23 Simulate a series of $n = 500$ observations from the signal-plus-noise model presented in Example 1.12 with $\sigma_w^2 = 1$. Compute the sample ACF to lag 100 of the data you generated and comment.

1.24 For the time series y_t described in Example 1.23, verify the stated result that $\rho_y(1) = -.47$ and $\rho_y(h) = 0$ for $h > 1$.

1.25 A real-valued function $g(t)$, defined on the integers, is non-negative definite if and only if

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i g(t_i - t_j) a_j \geq 0$$

for all positive integers n and for all vectors $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)'$ and $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)'$. For the matrix $G = \{g(t_i - t_j);\, i, j = 1, 2, \ldots, n\}$, this implies that $\boldsymbol{a}'G\boldsymbol{a} \geq 0$ for all vectors $\boldsymbol{a}$. It is called positive definite if we can replace '$\geq$' with '$>$' for all $\boldsymbol{a} \neq \boldsymbol{0}$, the zero vector.

(a) Prove that $\gamma(h)$, the autocovariance function of a stationary process, is a non-negative definite function.
(b) Verify that the sample autocovariance $\widehat{\gamma}(h)$ is a non-negative definite function.

Section 1.7

1.26 Consider a collection of time series $x_{1t}, x_{2t}, \ldots, x_{Nt}$ that are observing some common signal μ_t observed in noise processes $e_{1t}, e_{2t}, \ldots, e_{Nt}$, with a model for the j-th observed series given by

$$x_{jt} = \mu_t + e_{jt}.$$

Suppose the noise series have zero means and are uncorrelated for different j. The common autocovariance functions of all series are given by $\gamma_e(s,t)$. Define the sample mean

$$\bar{x}_t = \frac{1}{N}\sum_{j=1}^{N} x_{jt}.$$

(a) Show that $E[\bar{x}_t] = \mu_t$.
(b) Show that $E[(\bar{x}_t - \mu)^2)] = N^{-1}\gamma_e(t,t)$.
(c) How can we use the results in estimating the common signal?

1.27 A concept used in geostatistics, see Journel and Huijbregts (1978) or Cressie (1993), is that of the variogram, defined for a spatial process $x_{\boldsymbol{s}}$, $\boldsymbol{s} = (s_1, s_2)$, for $s_1, s_2 = 0, \pm 1, \pm 2, ...$, as

$$V_x(\boldsymbol{h}) = \frac{1}{2} E[(x_{\boldsymbol{s}+\boldsymbol{h}} - x_{\boldsymbol{s}})^2],$$

where $\boldsymbol{h} = (h_1, h_2)$, for $h_1, h_2 = 0, \pm 1, \pm 2, ...$ Show that, for a stationary process, the variogram and autocovariance functions can be related through

$$V_x(\boldsymbol{h}) = \gamma(\boldsymbol{0}) - \gamma(\boldsymbol{h}),$$

where $\gamma(\boldsymbol{h})$ is the usual lag $\boldsymbol{h}$ covariance function and $\boldsymbol{0} = (0, 0)$. Note the easy extension to any spatial dimension.

The following problems require the material given in Appendix A

1.28 Suppose $x_t = \beta_0 + \beta_1 t$, where β_0 and β_1 are constants. Prove as $n \to \infty$, $\widehat{\rho}_x(h) \to 1$ for fixed h, where $\widehat{\rho}_x(h)$ is the ACF (1.35).

1.29 (a) Suppose x_t is a weakly stationary time series with mean zero and with absolutely summable autocovariance function, $\gamma(h)$, such that

$$\sum_{h=-\infty}^{\infty} \gamma(h) = 0.$$

Prove that $\sqrt{n}\, \bar{x} \xrightarrow{p} 0$, where $\bar{x}$ is the sample mean (1.32).

(b) Give an example of a process that satisfies the conditions of part (a). What is special about this process?

1.30 Let x_t be a linear process of the form (A.43)–(A.44). If we define

$$\tilde{\gamma}(h) = n^{-1} \sum_{t=1}^{n} (x_{t+h} - \mu_x)(x_t - \mu_x),$$

show that

$$n^{1/2} \big(\tilde{\gamma}(h) - \widehat{\gamma}(h) \big) = o_p(1).$$

Hint: The Markov Inequality

$$P\{|x| \geq \epsilon\} < \frac{E|x|}{\epsilon}$$

can be helpful for the cross-product terms.

1.31 For a linear process of the form

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

where $\{w_t\}$ satisfies the conditions of Theorem A.7 and $|\phi| < 1$, show that

$$\sqrt{n}\frac{(\widehat{\rho}_x(1) - \rho_x(1))}{\sqrt{1 - \rho_x^2(1)}} \xrightarrow{d} N(0,1),$$

and construct a 95% confidence interval for ϕ when $\widehat{\rho}_x(1) = .64$ and $n = 100$.

1.32 Let $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ be iid$(0, \sigma^2)$.

(a) For $h \geq 1$ and $k \geq 1$, show that $x_t x_{t+h}$ and $x_s x_{s+k}$ are uncorrelated for all $s \neq t$.

(b) For fixed $h \geq 1$, show that the $h \times 1$ vector

$$\sigma^{-2} n^{-1/2} \sum_{t=1}^{n} (x_t x_{t+1}, \ldots, x_t x_{t+h})' \xrightarrow{d} (z_1, \ldots, z_h)'$$

where $z_1, \ldots, z_h$ are iid N$(0,1)$ random variables. [Note: the sequence $\{x_t x_{t+h};\ t = 1, 2, \ldots\}$ is h-dependent and white noise $(0, \sigma^4)$. Also, recall the Cramér-Wold device.]

(c) Show, for each $h \geq 1$,

$$n^{-1/2}\left[\sum_{t=1}^{n} x_t x_{t+h} - \sum_{t=1}^{n-h} (x_t - \bar{x})(x_{t+h} - \bar{x})\right] \xrightarrow{p} 0 \qquad \text{as } n \to \infty$$

where $\bar{x} = n^{-1} \sum_{t=1}^{n} x_t$.

(d) Noting that $n^{-1} \sum_{t=1}^{n} x_t^2 \xrightarrow{p} \sigma^2$, conclude that

$$n^{1/2} \left[\widehat{\rho}(1), \ldots, \widehat{\rho}(h)\right]' \xrightarrow{d} (z_1, \ldots, z_h)'$$

where $\widehat{\rho}(h)$ is the sample ACF of the data $x_1, \ldots, x_n$.

2

Time Series Regression and Exploratory Data Analysis

2.1 Introduction

The linear model and its applications are at least as dominant in the time series context as in classical statistics. Regression models are important for time domain models discussed in Chapters 3, 5, and 6, and in the frequency domain models considered in Chapters 4 and 7. The primary ideas depend on being able to express a response series, say x_t, as a linear combination of inputs, say $z_{t1}, z_{t2}, \ldots, z_{tq}$. Estimating the coefficients $\beta_1, \beta_2, \ldots, \beta_q$ in the linear combinations by least squares provides a method for modeling x_t in terms of the inputs.

In the time domain applications of Chapter 3, for example, we will express x_t as a linear combination of previous values $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$, of the currently observed series. The outputs x_t may also depend on lagged values of another series, say $y_{t-1}, y_{t-2}, \ldots, y_{t-q}$, that have influence. It is easy to see that forecasting becomes an option when prediction models can be formulated in this form. Time series smoothing and filtering can be expressed in terms of local regression models. Polynomials and regression splines also provide important techniques for smoothing.

If one admits sines and cosines as inputs, the frequency domain ideas that lead to the periodogram and spectrum of Chapter 4 follow from a regression model. Extensions to filters of infinite extent can be handled using regression in the frequency domain. In particular, many regression problems in the frequency domain can be carried out as a function of the periodic components of the input and output series, providing useful scientific intuition into fields like acoustics, oceanographics, engineering, biomedicine, and geophysics.

The above considerations motivate us to include a separate chapter on regression and some of its applications that is written on an elementary level and is formulated in terms of time series. The assumption of linearity, stationarity, and homogeneity of variances over time is critical in the regression

R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer Texts in Statistics, DOI 10.1007/978-1-4419-7865-3_2,

context, and therefore we include some material on transformations and other techniques useful in exploratory data analysis.

2.2 Classical Regression in the Time Series Context

We begin our discussion of linear regression in the time series context by assuming some output or dependent time series, say, x_t, for $t = 1, \ldots, n$, is being influenced by a collection of possible inputs or independent series, say, $z_{t1}, z_{t2}, \ldots, z_{tq}$, where we first regard the inputs as fixed and known. This assumption, necessary for applying conventional linear regression, will be relaxed later on. We express this relation through the linear regression model

$$x_t = \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq} + w_t, \tag{2.1}$$

where $\beta_1, \beta_2, \ldots, \beta_q$ are unknown fixed regression coefficients, and $\{w_t\}$ is a random error or noise process consisting of independent and identically distributed (iid) normal variables with mean zero and variance σ_w^2; we will relax the iid assumption later. A more general setting within which to embed mean square estimation and linear regression is given in Appendix B, where we introduce Hilbert spaces and the Projection Theorem.

Example 2.1 Estimating a Linear Trend

Consider the global temperature data, say x_t, shown in Figures 1.2 and 2.1. As discussed in Example 1.2, there is an apparent upward trend in the series that has been used to argue the global warming hypothesis. We might use simple linear regression to estimate that trend by fitting the model

$$x_t = \beta_1 + \beta_2 t + w_t, \quad t = 1880, 1857, \ldots, 2009.$$

This is in the form of the regression model (2.1) when we make the identification $q = 2$, $z_{t1} = 1$ and $z_{t2} = t$. Note that we are making the assumption that the errors, w_t, are an iid normal sequence, which may not be true. We will address this problem further in §2.3; the problem of autocorrelated errors is discussed in detail in §5.5. Also note that we could have used, for example, $t = 1, \ldots, 130$, without affecting the interpretation of the slope coefficient, β_2; only the intercept, β_1, would be affected.

Using simple linear regression, we obtained the estimated coefficients $\widehat{\beta}_1 = -11.2$, and $\widehat{\beta}_2 = .006$ (with a standard error of .0003) yielding a highly significant estimated increase of .6 degrees centigrade per 100 years. We discuss the precise way in which the solution was accomplished after the example. Finally, Figure 2.1 shows the global temperature data, say x_t, with the estimated trend, say $\widehat{x}_t = -11.2 + .006t$, superimposed. It is apparent that the estimated trend line obtained via simple linear regression does not quite capture the trend of the data and better models will be needed.

To perform this analysis in R, use the following commands:

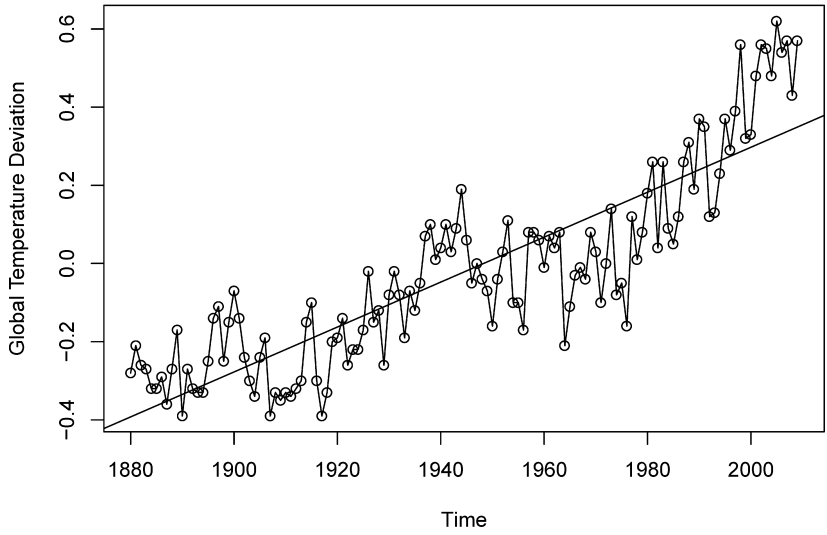

Fig. 2.1. Global temperature deviations shown in Figure 1.2 with fitted linear trend line.

```
summary(fit <- lm(gtemp~time(gtemp)))  # regress gtemp on time
plot(gtemp, type="o", ylab="Global Temperature Deviation")
abline(fit)  # add regression line to the plot
```

The linear model described by (2.1) above can be conveniently written in a more general notation by defining the column vectors $\boldsymbol{z}_t = (z_{t1}, z_{t2}, \ldots, z_{tq})'$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_q)'$, where $'$ denotes transpose, so (2.1) can be written in the alternate form

$$x_t = \boldsymbol{\beta}' \boldsymbol{z}_t + w_t. \tag{2.2}$$

where $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$. It is natural to consider estimating the unknown coefficient vector $\boldsymbol{\beta}$ by minimizing the error sum of squares

$$Q = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} (x_t - \boldsymbol{\beta}' \boldsymbol{z}_t)^2, \tag{2.3}$$

with respect to $\beta_1, \beta_2, \ldots, \beta_q$. Minimizing Q yields the ordinary least squares estimator of $\boldsymbol{\beta}$. This minimization can be accomplished by differentiating (2.3) with respect to the vector $\boldsymbol{\beta}$ or by using the properties of projections. In the notation above, this procedure gives the normal equations

$$\left(\sum_{t=1}^{n} \boldsymbol{z}_t \boldsymbol{z}_t' \right) \widehat{\boldsymbol{\beta}} = \sum_{t=1}^{n} \boldsymbol{z}_t x_t. \tag{2.4}$$

The notation can be simplified by defining $Z = [\boldsymbol{z}_1 \,|\, \boldsymbol{z}_2 \,|\, \cdots \,|\, \boldsymbol{z}_n]'$ as the $n \times q$ matrix composed of the n samples of the input variables, the observed $n \times 1$ vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)'$ and the $n \times 1$ vector of errors

$\boldsymbol{w} = (w_1, w_2, \ldots, w_n)'$. In this case, model (2.2) may be written as

$$\boldsymbol{x} = Z\boldsymbol{\beta} + \boldsymbol{w}. \tag{2.5}$$

The normal equations, (2.4), can now be written as

$$(Z'Z)\,\widehat{\boldsymbol{\beta}} = Z'\boldsymbol{x} \tag{2.6}$$

and the solution

$$\widehat{\boldsymbol{\beta}} = (Z'Z)^{-1}Z'\boldsymbol{x} \tag{2.7}$$

when the matrix $Z'Z$ is nonsingular. The minimized error sum of squares (2.3), denoted SSE, can be written as

$$\begin{aligned} SSE &= \sum_{t=1}^{n}(x_t - \widehat{\boldsymbol{\beta}}'\boldsymbol{z}_t)^2 \\ &= (\boldsymbol{x} - Z\widehat{\boldsymbol{\beta}})'(\boldsymbol{x} - Z\widehat{\boldsymbol{\beta}}) \\ &= \boldsymbol{x}'\boldsymbol{x} - \widehat{\boldsymbol{\beta}}'Z'\boldsymbol{x} \\ &= \boldsymbol{x}'\boldsymbol{x} - \boldsymbol{x}'Z(Z'Z)^{-1}Z'\boldsymbol{x}, \end{aligned} \tag{2.8}$$

to give some useful versions for later reference. The ordinary least squares estimators are unbiased, i.e., $E(\widehat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$, and have the smallest variance within the class of linear unbiased estimators.

If the errors w_t are normally distributed, $\widehat{\boldsymbol{\beta}}$ is also the maximum likelihood estimator for $\boldsymbol{\beta}$ and is normally distributed with

$$\mathrm{cov}(\widehat{\boldsymbol{\beta}}) = \sigma_w^2\left(\sum_{t=1}^{n}\boldsymbol{z}_t\boldsymbol{z}_t'\right)^{-1} = \sigma_w^2(Z'Z)^{-1} = \sigma_w^2 C, \tag{2.9}$$

where

$$C = (Z'Z)^{-1} \tag{2.10}$$

is a convenient notation for later equations. An unbiased estimator for the variance σ_w^2 is

$$s_w^2 = MSE = \frac{SSE}{n-q}, \tag{2.11}$$

where MSE denotes the *mean squared error*, which is contrasted with the maximum likelihood estimator $\widehat{\sigma}_w^2 = SSE/n$. Under the normal assumption, s_w^2 is distributed proportionally to a chi-squared random variable with $n-q$ degrees of freedom, denoted by χ^2_{n-q}, and independently of $\widehat{\beta}$. It follows that

$$t_{n-q} = \frac{(\widehat{\beta}_i - \beta_i)}{s_w\sqrt{c_{ii}}} \tag{2.12}$$

has the t-distribution with $n-q$ degrees of freedom; c_{ii} denotes the i-th diagonal element of C, as defined in (2.10).

Table 2.1. Analysis of Variance for Regression

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

Various competing models are 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, $\boldsymbol{z}_{t:r} = (z_{t1}, z_{t2}, \ldots, z_{tr})'$ is influencing the dependent variable x_t. The reduced model is

$$\boldsymbol{x} = Z_r \boldsymbol{\beta}_r + \boldsymbol{w} \tag{2.13}$$

where $\boldsymbol{\beta}_r = (\beta_1, \beta_2, \ldots, \beta_r)'$ is a subset of coefficients of the original q variables and $Z_r = [\boldsymbol{z}_{1:r} \mid \cdots \mid \boldsymbol{z}_{n:r}]'$ is the $n \times r$ matrix of inputs. The null hypothesis in this case is H_0: $\beta_{r+1} = \cdots = \beta_q = 0$. We can test the reduced model (2.13) against the full model (2.2) by comparing the error sums of squares under the two models using the F-statistic

$$F_{q-r,n-q} = \frac{(SSE_r - SSE)/(q-r)}{SSE/(n-q)}, \tag{2.14}$$

which has the central F-distribution with $q-r$ and $n-q$ degrees of freedom when (2.13) is the correct model. Note that SSE_r is the error sum of squares under the reduced model (2.13) and it can be computed by replacing Z with Z_r in (2.8). The statistic, which follows from applying the likelihood ratio criterion, has the improvement per number of parameters added in the numerator compared with the error sum of squares under the full model in the denominator. The information involved in the test procedure is often summarized in an Analysis of Variance (ANOVA) table as given in Table 2.1 for this particular case. The difference in the numerator is often called the regression sum of squares

In terms of Table 2.1, it is conventional to write the F-statistic (2.14) as the ratio of the two mean squares, obtaining

$$F_{q-r,n-q} = \frac{MSR}{MSE}, \tag{2.15}$$

where MSR, the *mean squared regression*, is the numerator of (2.14). A special case of interest is $r = 1$ and $z_{t1} \equiv 1$, when the model in (2.13) becomes

$$x_t = \beta_1 + w_t,$$

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

$$R^2 = \frac{SSE_1 - SSE}{SSE_1}, \tag{2.16}$$

where the residual sum of squares under the reduced model

$$SSE_1 = \sum_{t=1}^{n} (x_t - \bar{x})^2, \tag{2.17}$$

in this case is just the sum of squared deviations from the mean $\bar{x}$. The measure R^2 is also the *squared multiple correlation* between x_t and the variables $z_{t2}, z_{t3}, \ldots, z_{tq}$.

The techniques discussed in the previous paragraph can be used to test various models against one another using the F test given in (2.14), (2.15), and the ANOVA table. These tests have been used in the past in a stepwise manner, where variables are added or deleted when the values from the F-test either exceed or fail to exceed some predetermined levels. The procedure, called stepwise multiple regression, is useful in arriving at a set of useful variables. An alternative is to focus on a procedure for model selection that does not proceed sequentially, but simply evaluates each model on its own merits. Suppose we consider a normal regression model with k coefficients and denote the maximum likelihood estimator for the variance as

$$\widehat{\sigma}_k^2 = \frac{SSE_k}{n}, \tag{2.18}$$

where SSE_k denotes the residual sum of squares under the model with k regression coefficients. Then, Akaike (1969, 1973, 1974) suggested measuring the goodness of fit for this particular model by balancing the error of the fit against the number of parameters in the model; we define the following.[1]

Definition 2.1 Akaike's Information Criterion (AIC)

$$\text{AIC} = \log\ \widehat{\sigma}_k^2 + \frac{n + 2k}{n}, \tag{2.19}$$

where $\widehat{\sigma}_k^2$ is given by (2.18) and k is the number of parameters in the model.

The value of k yielding the minimum AIC specifies the best model. The idea is roughly that minimizing $\widehat{\sigma}_k^2$ would be a reasonable objective, except that it decreases monotonically as k increases. Therefore, we ought to penalize the error variance by a term proportional to the number of parameters. The choice for the penalty term given by (2.19) is not the only one, and a considerable literature is available advocating different penalty terms. A corrected

[1] Formally, AIC is defined as $-2 \log L_k + 2k$ where L_k is the maximized log-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 (2.19). AIC is an estimate of the Kullback-Leibler discrepency between a true model and a candidate model; see Problems 2.4 and 2.5 for further details.

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 (details are provided in Problems 2.4 and 2.5). The corrected form is defined as follows.

Definition 2.2 AIC, Bias Corrected (AICc)

$$\text{AICc} = \log\ \widehat{\sigma}_k^2 + \frac{n+k}{n-k-2}, \tag{2.20}$$

where $\widehat{\sigma}_k^2$ is given by (2.18), k is the number of parameters in the model, and n is the sample size.

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

Definition 2.3 Bayesian Information Criterion (BIC)

$$\text{BIC} = \log\ \widehat{\sigma}_k^2 + \frac{k \log n}{n}, \tag{2.21}$$

using the same notation as in Definition 2.2.

BIC is also called the Schwarz Information Criterion (SIC); see also Rissanen (1978) for an approach yielding the same statistic based on a minimum description length argument. 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. In fitting regression models, two measures that have been used in the past are adjusted R-squared, which is essentially s_w^2, and Mallows C_p, Mallows (1973), which we do not consider in this context.

Example 2.2 Pollution, Temperature and Mortality

The data shown in Figure 2.2 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.

A scatterplot matrix, shown in Figure 2.3, indicates a possible linear relation between mortality and the pollutant particulates and a possible relation to temperature. Note the curvilinear shape of the temperature mortality curve, indicating that higher temperatures as well as lower temperatures are associated with increases in cardiovascular mortality.

Based on the scatterplot matrix, we entertain, tentatively, four models where M_t denotes cardiovascular mortality, T_t denotes temperature and P_t denotes the particulate levels. They are

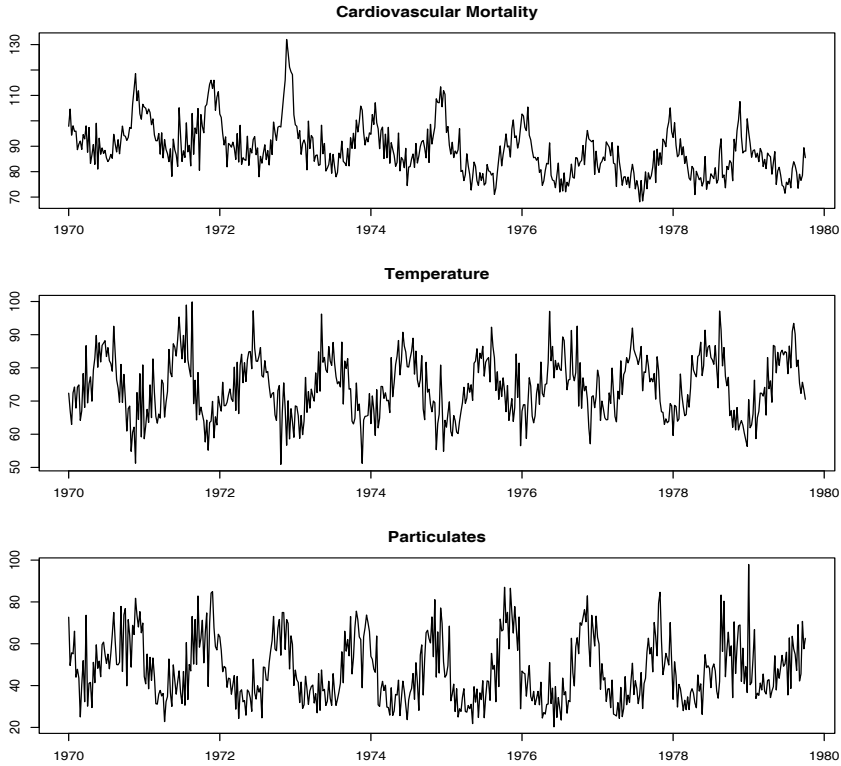

Fig. 2.2. Average weekly cardiovascular mortality (top), temperature (middle) and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.

$$M_t = \beta_1 + \beta_2 t + w_t \tag{2.22}$$
$$M_t = \beta_1 + \beta_2 t + \beta_3 (T_t - T_\cdot) + w_t \tag{2.23}$$
$$M_t = \beta_1 + \beta_2 t + \beta_3 (T_t - T_\cdot) + \beta_4 (T_t - T_\cdot)^2 + w_t \tag{2.24}$$
$$M_t = \beta_1 + \beta_2 t + \beta_3 (T_t - T_\cdot) + \beta_4 (T_t - T_\cdot)^2 + \beta_5 P_t + w_t \tag{2.25}$$

where we adjust temperature for its mean, $T_\cdot = 74.6$, to avoid scaling problems. It is clear that (2.22) is a trend only model, (2.23) is linear temperature, (2.24) is curvilinear temperature and (2.25) is curvilinear temperature and pollution. We summarize some of the statistics given for this particular case in Table 2.2. The values of R^2 were computed by noting that $SSE_1 = 50,687$ using (2.17).

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

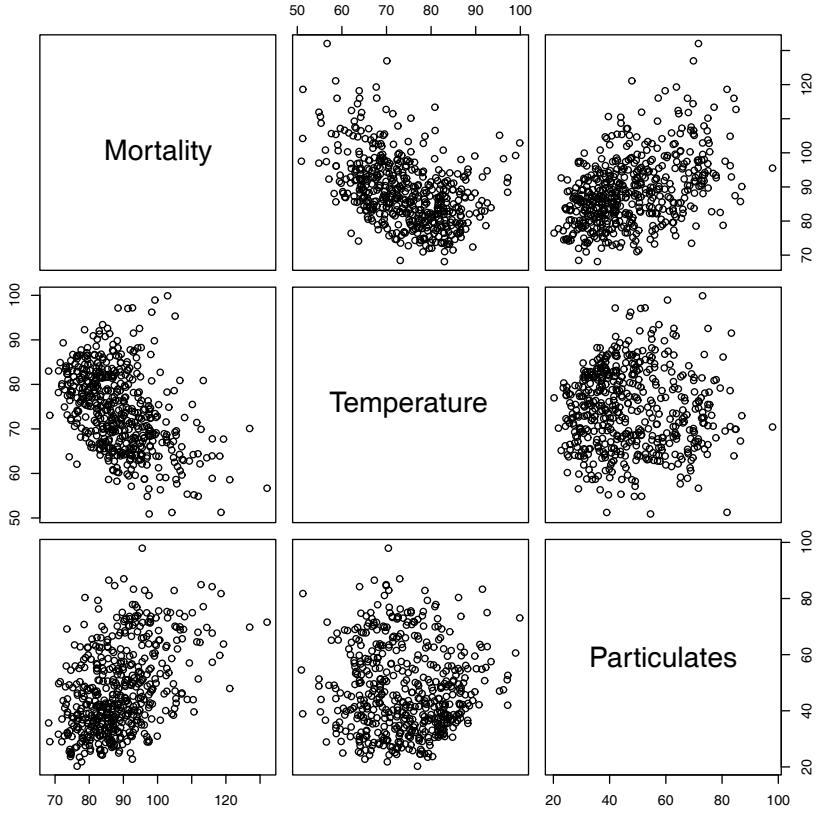

Fig. 2.3. Scatterplot matrix showing plausible relations between mortality, temperature, and pollution.

Table 2.2. Summary Statistics for Mortality Models

Model	k	SSE	df	MSE	R^2	AIC	BIC
(2.22)	2	40,020	506	79.0	.21	5.38	5.40
(2.23)	3	31,413	505	62.2	.38	5.14	5.17
(2.24)	4	27,985	504	55.5	.45	5.03	5.07
(2.25)	5	20,508	503	40.8	.60	4.72	4.77

and AICc are nearly the same). Note that one can compare any two models using the residual sums of squares and (2.14). Hence, a model with only trend could be compared to the full model using $q = 5, r = 2, n = 508$, 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,

$$\widehat{M}_t = 81.59 - .027_{(.002)}t - .473_{(.032)}(T_t - 74.6) + .023_{(.003)}(T_t - 74.6)^2 + .255_{(.019)}P_t,$$

for mortality, where the standard errors, computed from (2.9)-(2.11), are given in parentheses. As expected, a negative trend is present in time as well as a negative coefficient for adjusted temperature. The quadratic effect of temperature can clearly be seen in the scatterplots of Figure 2.3. 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 $\widehat{w}_t = M_t - \widehat{M}_t$ for autocorrelation (of which there is a substantial amount), but we defer this question to to §5.6 when we discuss regression with correlated errors.

Below is the R code to plot the series, display the scatterplot matrix, fit the final regression model (2.25), and compute the corresponding values of AIC, AICc and BIC.[2] Finally, the use of `na.action` in `lm()` is to retain the time series attributes for the residuals and fitted values.

```
par(mfrow=c(3,1))
plot(cmort, main="Cardiovascular Mortality", xlab="", ylab="")
plot(tempr, main="Temperature", xlab="", ylab="")
plot(part, main="Particulates", xlab="", ylab="")
dev.new()  # open a new graphic device for the scatterplot matrix
pairs(cbind(Mortality=cmort, Temperature=tempr, Particulates=part))
temp = tempr-mean(tempr)  # center temperature
temp2 = temp^2
trend = time(cmort)  # time
fit = lm(cmort~ trend + temp + temp2 + part, na.action=NULL)
summary(fit)  # regression results
summary(aov(fit))  # ANOVA table   (compare to next line)
summary(aov(lm(cmort~cbind(trend, temp, temp2, part)))) # Table 2.1
num = length(cmort)  # sample size
AIC(fit)/num - log(2*pi)  # AIC
AIC(fit, k=log(num))/num - log(2*pi)  # BIC
(AICc = log(sum(resid(fit)^2)/num) + (num+5)/(num-5-2)) # AICc
```

As previously mentioned, it is possible to include lagged variables in time series regression models and we will continue to discuss this type of problem throughout the text. This concept is explored further in Problems 2.2 and 2.11. The following is a simple example of lagged regression.

[2] The easiest way to extract AIC and BIC from an `lm()` run in R is to use the command `AIC()`. Our definitions differ from R by terms that do not change from model to model. In the example, we show how to obtain (2.19) and (2.21) from the R output. It is more difficult to obtain AICc.

Example 2.3 Regression With Lagged Variables

In Example 1.25, 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 evidence that the relationship is not linear (this is discussed further in Example 2.7), we may consider the following regression,

$$R_t = \beta_1 + \beta_2 S_{t-6} + w_t, \tag{2.26}$$

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

$$\widehat{R}_t = 65.79 - 44.28_{(2.78)} S_{t-6} \tag{2.27}$$

with $\widehat{\sigma}_w = 22.5$ on 445 degrees of freedom. This result indicates the strong predictive ability of SOI for Recruitment six months in advance. Of course, it is still essential to check the the model assumptions, but again we defer this until later.

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 a data frame that we call `fish` using `ts.intersect`, which aligns the lagged series.

```
1 fish = ts.intersect(rec, soiL6=lag(soi,-6), dframe=TRUE)
2 summary(lm(rec~soiL6, data=fish, na.action=NULL))
```

2.3 Exploratory Data Analysis

In general, it is necessary for time series data to be stationary, so averaging lagged products over time, as in the previous section, will be a sensible thing to do. With time series data, 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 that dependence if the dependence structure is not regular or is changing at every time point. Hence, to achieve any meaningful statistical analysis of time series data, it will be crucial that, if nothing else, the mean and the autocovariance functions satisfy the conditions of stationarity (for at least some reasonable stretch of time) stated in Definition 1.7. Often, this is not the case, and we will mention some methods in this section for playing down the effects of nonstationarity so the stationary properties of the series may be studied.

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 contains some

evidence of a trend over time; human-induced global warming advocates seize on this as empirical evidence to advance their hypothesis that temperatures are increasing.

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

$$x_t = \mu_t + y_t \tag{2.28}$$

where x_t are the observations, μ_t denotes the trend, and y_t is a stationary process. Quite often, strong trend, μ_t, 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

$$\widehat{y}_t = x_t - \widehat{\mu}_t. \tag{2.29}$$

Consider the following example.

Example 2.4 Detrending Global Temperature

Here we suppose the model is of the form of (2.28),

$$x_t = \mu_t + y_t,$$

where, as we suggested in the analysis of the global temperature data presented in Example 2.1, a straight line might be a reasonable model for the trend, i.e.,

$$\mu_t = \beta_1 + \beta_2\, t.$$

In that example, we estimated the trend using ordinary least squares[3] and found

$$\widehat{\mu}_t = -11.2 + .006\, t.$$

Figure 2.1 shows the data with the estimated trend line superimposed. To obtain the detrended series we simply subtract $\widehat{\mu}_t$ from the observations, x_t, to obtain the detrended series

$$\widehat{y}_t = x_t + 11.2 - .006\, t.$$

The top graph of Figure 2.4 shows the detrended series. Figure 2.5 shows the ACF of the original data (top panel) as well as the ACF of the detrended data (middle panel).

[3] Because the error term, y_t, is not assumed to be iid, 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 (1957, Ch 7), however, is that under mild conditions on y_t, for polynomial regression or periodic regression, asymptotically, ordinary least squares is equivalent to weighted least squares.

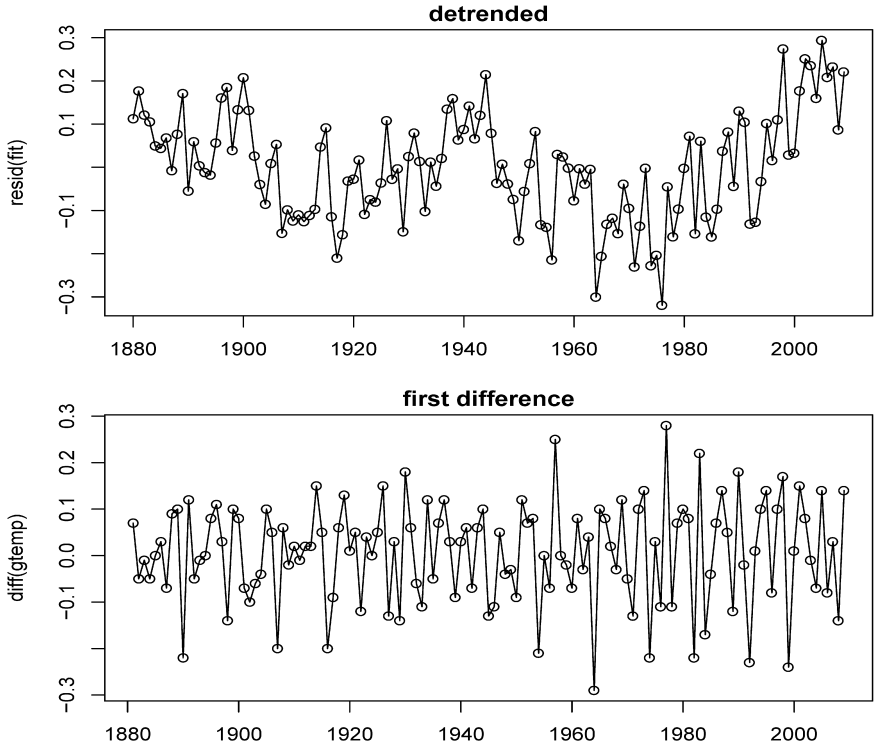

Fig. 2.4. Detrended (top) and differenced (bottom) global temperature series. The original data are shown in Figures 1.2 and 2.1.

To detrend in the series in R, use the following commands. We also show how to difference and plot the differenced data; we discuss differencing after this example. In addition, we show how to generate the sample ACFs displayed in Figure 2.5.

```
fit = lm(gtemp~time(gtemp), na.action=NULL) # regress gtemp on time
par(mfrow=c(2,1))
plot(resid(fit), type="o", main="detrended")
plot(diff(gtemp), type="o", main="first difference")
par(mfrow=c(3,1))  # plot ACFs
acf(gtemp, 48, main="gtemp")
acf(resid(fit), 48, main="detrended")
acf(diff(gtemp), 48, main="first difference")
```

In Example 1.11 and the corresponding Figure 1.10 we saw that a random walk might also be a good model for trend. That is, rather than modeling trend as fixed (as in Example 2.4), we might model trend as a stochastic component using the random walk with drift model,

$$\mu_t = \delta + \mu_{t-1} + w_t, \tag{2.30}$$

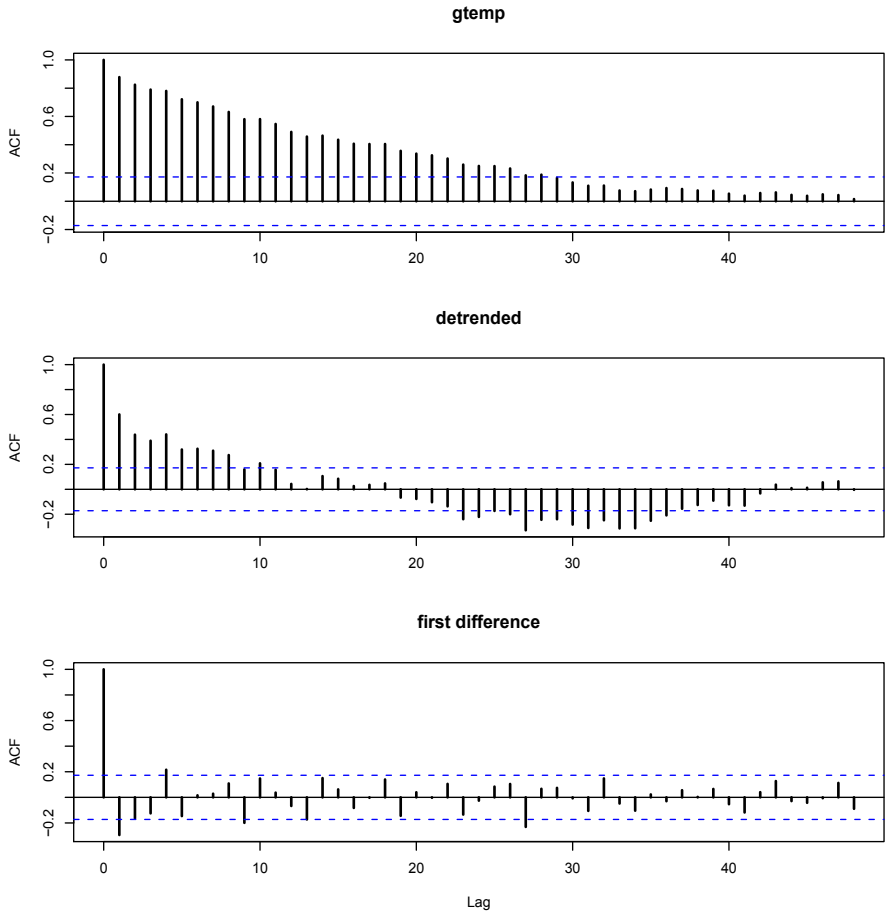

Fig. 2.5. Sample ACFs of the global temperature (top), and of the detrended (middle) and the differenced (bottom) series.

where w_t is white noise and is independent of y_t. If the appropriate model is (2.28), then differencing the data, x_t, yields a stationary process; that is,

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + w_t + y_t - y_{t-1}. \end{aligned} \tag{2.31}$$

It is easy to show $z_t = y_t - y_{t-1}$ is stationary using footnote 3 of Chapter 1 on page 20. That is, because y_t is stationary,

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

is independent of time; we leave it as an exercise (Problem 2.7) to show that $x_t - x_{t-1}$ in (2.31) 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 (2.31). If an estimate of y_t is essential, then detrending may be more appropriate. 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 2.4. That is, e.g., if $\mu_t = \beta_1 + \beta_2 t$ in the model (2.28), differencing the data produces stationarity (see Problem 2.6):

$$x_t - x_{t-1} = (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) = \beta_2 + y_t - y_{t-1}.$$

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

$$\nabla x_t = x_t - x_{t-1}. \tag{2.32}$$

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of (2.32), 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 3.

Definition 2.4 *We define the* **backshift operator** *by*

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

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

$$B^k x_t = x_{t-k}. \tag{2.33}$$

It is clear that we may then rewrite (2.32) as

$$\nabla x_t = (1 - B)x_t, \tag{2.34}$$

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

$$\begin{aligned} \nabla^2 x_t &= (1 - B)^2 x_t = (1 - 2B + B^2)x_t \\ &= x_t - 2x_{t-1} + x_{t-2} \end{aligned}$$

by the linearity of the operator. To check, just take the difference of the first difference $\nabla(\nabla x_t) = \nabla(x_t - x_{t-1}) = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2})$.

Definition 2.5 Differences of order d *are defined as*

$$\nabla^d = (1 - B)^d, \tag{2.35}$$

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 (2.32) 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 3. The differencing technique is an important component of the ARIMA model of Box and Jenkins (1970) (see also Box et al., 1994), to be discussed in Chapter 3.

Example 2.5 Differencing Global Temperature

The first difference of the global temperature series, also shown in Figure 2.4, produces different results than removing trend by detrending via regression. For example, the differenced series does not contain the long middle cycle we observe in the detrended series. The ACF of this series is also shown in Figure 2.5. In this case it appears that the differenced process shows minimal autocorrelation, 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 .0066 (but with a large standard error):

```
mean(diff(gtemp))    # = 0.00659 (drift)
sd(diff(gtemp))/sqrt(length(diff(gtemp))) # = 0.00966 (SE)
```

An alternative to differencing is a less-severe operation that still assumes stationarity of the underlying time series. This alternative, called fractional differencing, extends the notion of the difference operator (2.35) to fractional powers $-.5 < d < .5$, which still define stationary processes. Granger and Joyeux (1980) and Hosking (1981) introduced long memory time series, which corresponds to the case when $0 < d < .5$. This model is often used for environmental time series arising in hydrology. We will discuss long memory processes in more detail in §5.2.

Often, obvious aberrations are present that can contribute nonstationary as well as nonlinear behavior in observed time series. In such cases, transformations may be useful to equalize the variability over the length of a single series. A particularly useful transformation is

$$y_t = \log x_t, \tag{2.36}$$

which tends to suppress larger fluctuations that occur over portions of the series where the underlying values are larger. Other possibilities are power transformations in the Box–Cox family of the form

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log x_t & \lambda = 0. \end{cases} \tag{2.37}$$

Methods for choosing the power λ are available (see Johnson and Wichern, 1992, §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.

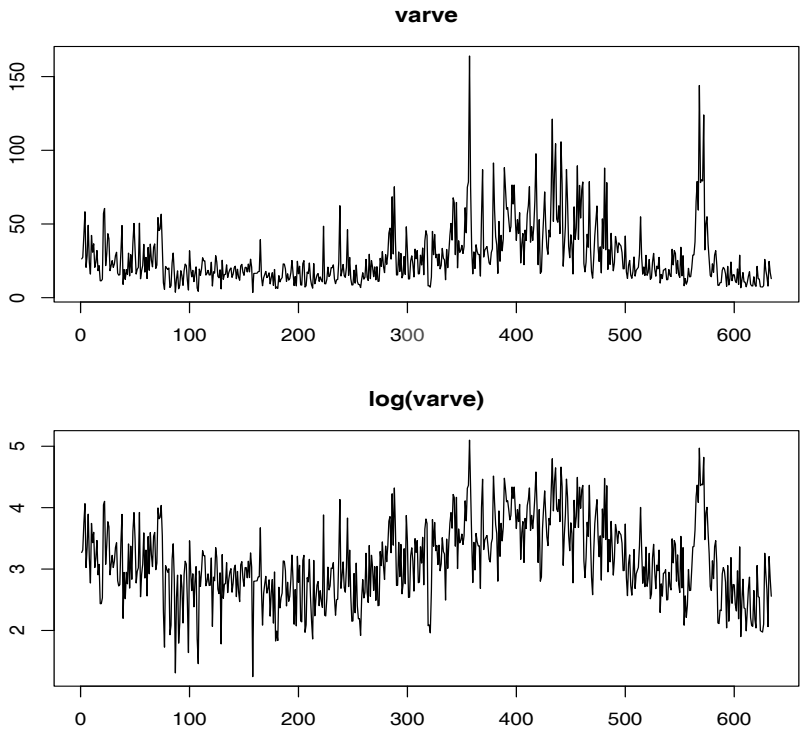

Fig. 2.6. Glacial varve thicknesses (top) from Massachusetts for $n = 634$ years compared with log transformed thicknesses (bottom).

Example 2.6 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. Figure 2.6 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 2.6 shows the original and transformed varves, and it is clear that this improvement has occurred. We may also plot the histogram of the original and transformed data, as in Problem 2.8, to argue that the approximation to normality is improved. The ordinary first differences (2.34) are also computed in Problem 2.8, and we note that the first differences have a significant negative correlation at

lag $h = 1$. Later, in Chapter 5, we will show that perhaps the varve series has long memory and will propose using fractional differencing.

Figure 2.6 was generated in R as follows:

```
par(mfrow=c(2,1))
plot(varve, main="varve", ylab="")
plot(log(varve), main="log(varve)", ylab="" )
```

Next, we consider another preliminary data processing technique that is used for the purpose of visualizing the relations between series at different lags, namely, scatterplot matrices. In the definition of the ACF, we are essentially interested in relations between x_t and x_{t-h}; the autocorrelation function tells us whether a substantial linear relation exists between the series and its own lagged values. The ACF gives a profile of the linear correlation at all possible lags and shows which values of h lead to the best predictability. The restriction of this idea to linear predictability, however, may mask a possible nonlinear relation between current values, x_t, and past values, x_{t-h}. This idea extends to two series where one may be interested in examining scatterplots of y_t versus x_{t-h}

Example 2.7 Scatterplot Matrices, SOI and Recruitment

To check for nonlinear relations of this form, it is convenient to display a lagged scatterplot matrix, as in Figure 2.7, that displays values 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 scatterplots 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 robust method for fitting nonlinear regression.

In Figure 2.7, we notice that the lowess 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 $h = 6, 7$. These results match up well with peaks noticed in the ACF in Figure 1.14.

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 predict the Recruitment series, R_t, from current or past values of the SOI series, S_{t-h}, for $h = 0, 1, 2, ...$ it would be worthwhile to examine the scatterplot matrix. Figure 2.8 shows the lagged scatterplot 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 2.8 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

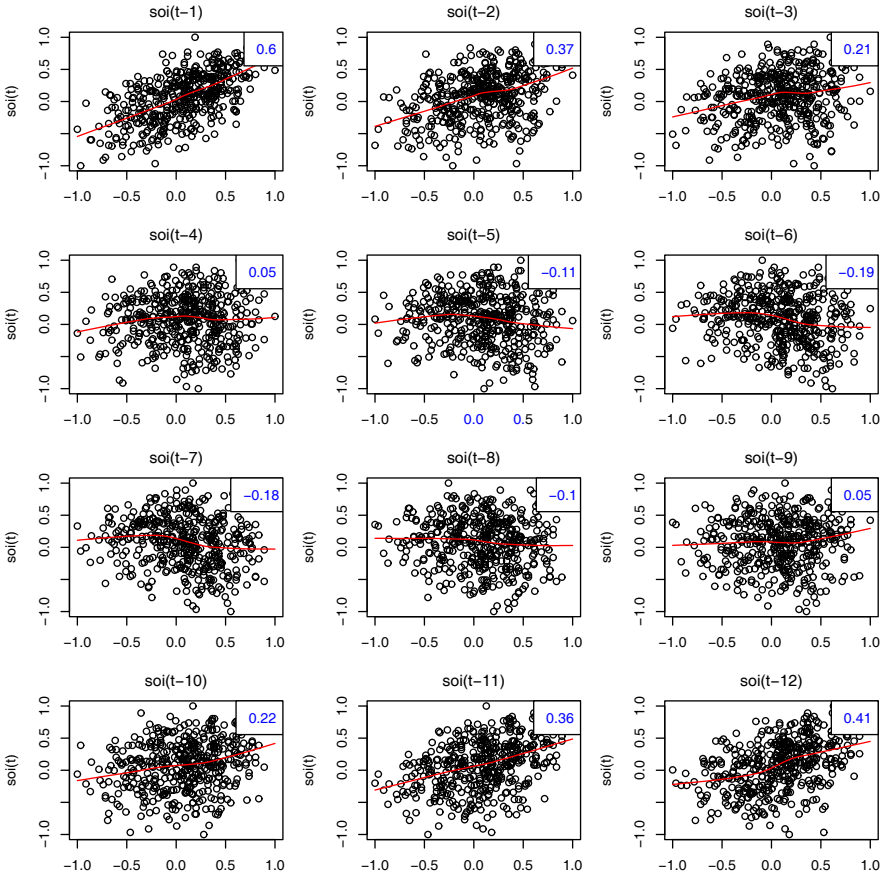

Fig. 2.7. Scatterplot matrix relating current SOI values, S_t, to past SOI values, S_{t-h}, at lags $h = 1, 2, ..., 12$. The values in the upper right corner are the sample autocorrelations and the lines are a lowess fit.

nonlinearity observed in the scatterplots (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.

Simple scatterplot matrices for one series can be obtained in R using the `lag.plot` command. Figures 2.7 and 2.8 may be reproduced using the following scripts provided with the text (see Appendix R for detials):

```
1 lag.plot1(soi, 12)      # Fig 2.7
2 lag.plot2(soi, rec, 8)  # Fig 2.8
```

As a final exploratory tool, we discuss assessing periodic behavior in time series data using regression analysis and the periodogram; this material may be thought of as an introduction to spectral analysis, which we discuss in

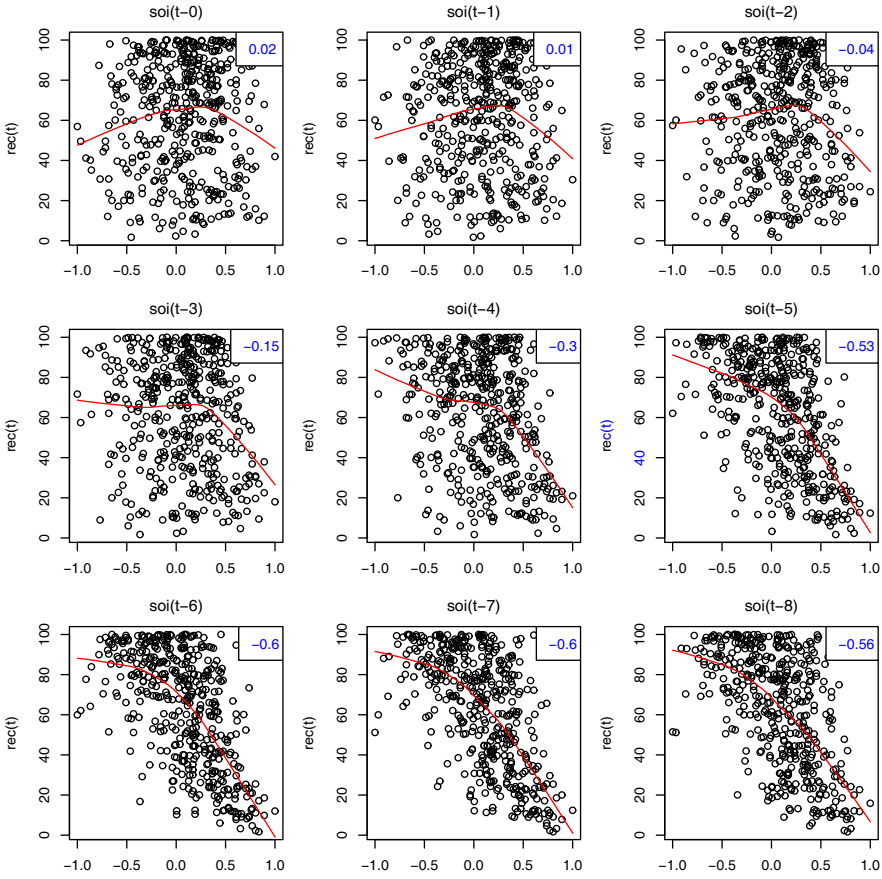

Fig. 2.8. Scatterplot matrix of the Recruitment series, R_t, on the vertical axis plotted against the SOI series, S_{t-h}, on the horizontal axis at lags $h = 0, 1, \ldots, 8$. The values in the upper right corner are the sample cross-correlations and the lines are a lowess fit.

detail in Chapter 4. In Example 1.12, 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 2.2 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 1.6 show strong yearly cycles, but hidden in the series are clues to the El Niño cycle.

Example 2.8 Using Regression to Discover a Signal in Noise

In Example 1.12, we generated $n = 500$ observations from the model

$$x_t = A\cos(2\pi\omega t + \phi) + w_t, \tag{2.38}$$

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 on page 16. 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 (2.38) in a nonlinear way, so we use a trigonometric identity[4] 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 (2.38) can be written in the usual linear regression form given by (no intercept term is needed here)

$$x_t = \beta_1\cos(2\pi t/50) + \beta_2\sin(2\pi t/50) + w_t. \tag{2.39}$$

Using linear regression on the generated data, the fitted model is

$$\widehat{x}_t = -.71_{(.30)}\cos(2\pi t/50) - 2.55_{(.30)}\sin(2\pi t/50) \tag{2.40}$$

with $\widehat{\sigma}_w = 4.68$, where 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 $\beta_2 = -2\sin(.6\pi) = -1.90$. Because the parameter estimates are significant and close to the actual values, it is clear that we are able to detect the signal in the noise using regression, even though the signal appears to be obscured by the noise in the bottom panel of Figure 1.11. Figure 2.9 shows data generated by (2.38) with the fitted line, (2.40), superimposed.

To reproduce the analysis and Figure 2.9 in R, use the following commands:

```
set.seed(1000)  # 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
plot.ts(x, lty="dashed")
lines(fitted(fit), lwd=2)
```

Example 2.9 Using the Periodogram to Discover a Signal in Noise

The analysis in Example 2.8 may seem like cheating because we assumed we knew the value of the frequency parameter ω. If we do not know ω, we could try to fit the model (2.38) using nonlinear regression with ω as a parameter. Another method is to try various values of ω in a systematic way. Using the

[4] $\cos(\alpha \pm \beta) = \cos(\alpha)\cos(\beta) \mp \sin(\alpha)\sin(\beta)$.

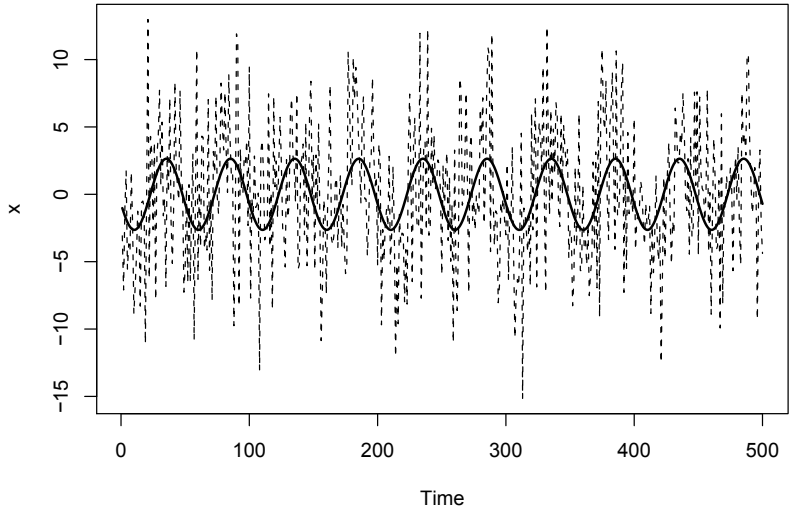

Fig. 2.9. Data generated by (2.38) [dashed line] with the fitted [solid] line, (2.40), superimposed.

regression results of §2.2, we can show the estimated regression coefficients in Example 2.8 take on the special form given by

$$\widehat{\beta}_1 = \frac{\sum_{t=1}^{n} x_t \cos(2\pi t/50)}{\sum_{t=1}^{n} \cos^2(2\pi t/50)} = \frac{2}{n}\sum_{t=1}^{n} x_t \cos(2\pi t/50); \tag{2.41}$$

$$\widehat{\beta}_2 = \frac{\sum_{t=1}^{n} x_t \sin(2\pi t/50)}{\sum_{t=1}^{n} \sin^2(2\pi t/50)} = \frac{2}{n}\sum_{t=1}^{n} x_t \sin(2\pi t/50). \tag{2.42}$$

This suggests looking at all possible regression parameter estimates,[5] say

$$\widehat{\beta}_1(j/n) = \frac{2}{n}\sum_{t=1}^{n} x_t \cos(2\pi t\, j/n); \tag{2.43}$$

$$\widehat{\beta}_2(j/n) = \frac{2}{n}\sum_{t=1}^{n} x_t \sin(2\pi t\, j/n), \tag{2.44}$$

where, $n = 500$ and $j = 1, \ldots, \frac{n}{2} - 1$, and inspecting the results for large values. For the endpoints, $j = 0$ and $j = n/2$, we have $\widehat{\beta}_1(0) = n^{-1}\sum_{t=1}^{n} x_t$ and $\widehat{\beta}_1(\frac{1}{2}) = n^{-1}\sum_{t=1}^{n}(-1)^t x_t$, and $\widehat{\beta}_2(0) = \widehat{\beta}_2(\frac{1}{2}) = 0$.

For this particular example, the values calculated in (2.41) and (2.42) are $\widehat{\beta}_1(10/500)$ and $\widehat{\beta}_2(10/500)$. By doing this, we have regressed a series, x_t, of

[5] In the notation of §2.2, the estimates are of the form $\sum_{t=1}^{n} x_t z_t \,/\, \sum_{t=1}^{n} z_t^2$ where $z_t = \cos(2\pi tj/n)$ or $z_t = \sin(2\pi tj/n)$. In this setup, unless $j = 0$ or $j = n/2$ if n is even, $\sum_{t=1}^{n} z_t^2 = n/2$; see Problem 2.10.

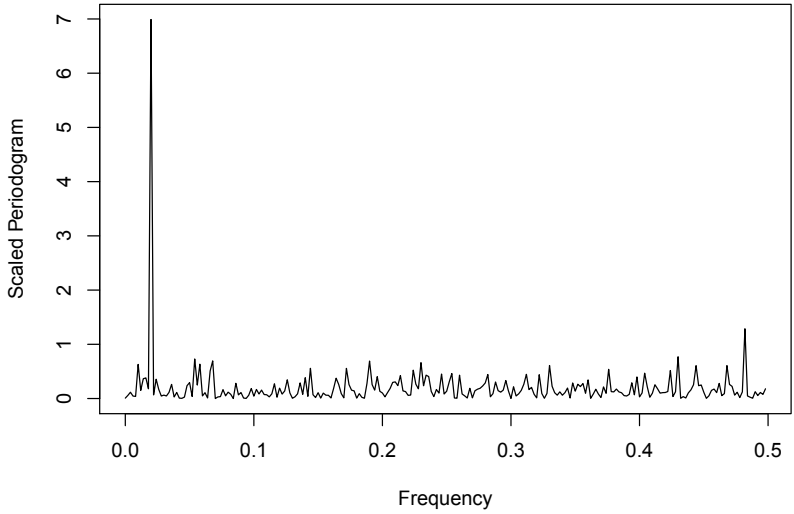

Fig. 2.10. The scaled periodogram, (2.45), of the 500 observations generated by (2.38); the data are displayed in Figures 1.11 and 2.9.

length n using n regression parameters, so that we will have a perfect fit. The point, however, is that if the data contain any cyclic behavior we are likely to catch it by performing these saturated regressions.

Next, note that the regression coefficients $\widehat{\beta}_1(j/n)$ and $\widehat{\beta}_2(j/n)$, for each j, are essentially measuring the correlation of the data with a sinusoid oscillating at j cycles in n time points.[6] Hence, an appropriate measure of the presence of a frequency of oscillation of j cycles in n time points in the data would be

$$P(j/n) = \widehat{\beta}_1^2(j/n) + \widehat{\beta}_2^2(j/n), \tag{2.45}$$

which is basically a measure of squared correlation. The quantity (2.45) is sometimes called the periodogram, but we will call $P(j/n)$ the scaled periodogram and we will investigate its properties in Chapter 4. Figure 2.10 shows the scaled periodogram for the data generated by (2.38), and it easily discovers the periodic component with frequency $\omega = .02 = 10/500$ even though it is difficult to visually notice that component in Figure 1.11 due to the noise.

Finally, we mention that it is not necessary to run a large regression

$$x_t = \sum_{j=0}^{n/2} \beta_1(j/n)\cos(2\pi t j/n) + \beta_2(j/n)\sin(2\pi t j/n) \tag{2.46}$$

to obtain the values of $\beta_1(j/n)$ and $\beta_2(j/n)$ [with $\beta_2(0) = \beta_2(1/2) = 0$] because they can be computed quickly if n (assumed even here) is a highly

[6] Sample correlations are of the form $\sum_t x_t z_t / \left(\sum_t x_t^2 \sum_t z_t^2\right)^{1/2}$.

composite integer. There is no error in (2.46) because there are n observations and n parameters; the regression fit will be perfect. The discrete Fourier transform (DFT) is a complex-valued weighted average of the data given by

$$\begin{aligned} d(j/n) &= n^{-1/2}\sum_{t=1}^{n} x_t \exp(-2\pi i t j/n) \\ &= n^{-1/2}\left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n) - i\sum_{t=1}^{n} x_t \sin(2\pi t j/n)\right) \end{aligned} \tag{2.47}$$

where the frequencies j/n are called the Fourier or fundamental frequencies. Because of a large number of redundancies in the calculation, (2.47) may be computed quickly using the fast Fourier transform (FFT)[7], which is available in many computing packages such as Matlab®, S-PLUS® and R. Note that[8]

$$|d(j/n)|^2 = \frac{1}{n}\left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n)\right)^2 + \frac{1}{n}\left(\sum_{t=1}^{n} x_t \sin(2\pi t j/n)\right)^2 \tag{2.48}$$

and it is this quantity that is called the periodogram; we will write

$$I(j/n) = |d(j/n)|^2.$$

We may calculate the scaled periodogram, (2.45), using the periodogram as

$$P(j/n) = \frac{4}{n} I(j/n). \tag{2.49}$$

We will discuss this approach in more detail and provide examples with data in Chapter 4.

Figure 2.10 can be created in R using the following commands (and the data already generated in `x`):

```
I = abs(fft(x))^2/500  # the periodogram
P = (4/500)*I[1:250]   # the scaled periodogram
f = 0:249/500          # frequencies
plot(f, P, type="l", xlab="Frequency", ylab="Scaled Periodogram")
```

2.4 Smoothing in the Time Series Context

In §1.4, we introduced the concept of smoothing a time series, and in Example 1.9, we discussed using a moving average to smooth white noise. This method is useful in discovering certain traits in a time series, such as long-term

[7] Different packages scale the FFT differently; consult the documentation. R calculates (2.47) without scaling by $n^{-1/2}$.

[8] If $z = a - ib$ is complex, then $|z|^2 = z\bar{z} = (a - ib)(a + ib) = a^2 + b^2$.

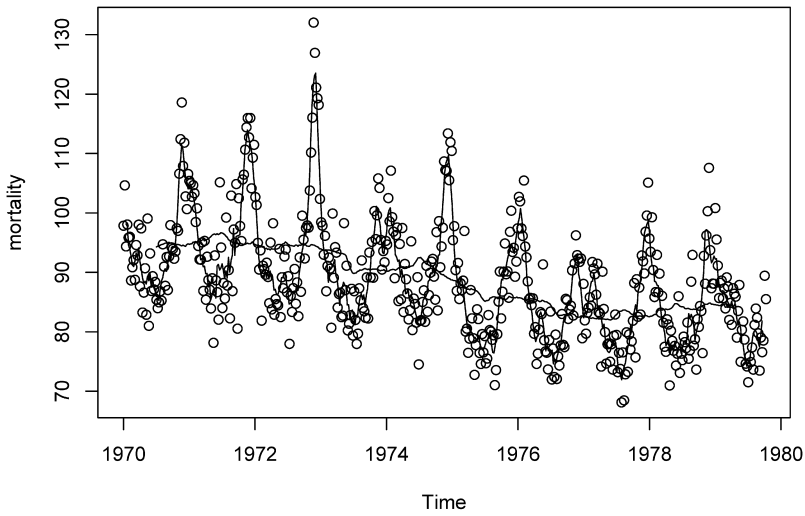

Fig. 2.11. The weekly cardiovascular mortality series discussed in Example 2.2 smoothed using a five-week moving average and a 53-week moving average.

trend and seasonal components. In particular, if x_t represents the observations, then

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j}, \tag{2.50}$$

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

Example 2.10 Moving Average Smoother

For example, Figure 2.11 shows the weekly mortality series discussed in Example 2.2, a five-point moving average (which is essentially a monthly average with $k = 2$) that helps bring out the seasonal component and a 53-point moving average (which is essentially a yearly average with $k = 26$) that helps bring out the (negative) trend in cardiovascular mortality. In both cases, the weights, $a_{-k}, \ldots, a_0, \ldots, a_k$, we used were all the same, and equal to $1/(2k + 1)$.[9]

To reproduce Figure 2.11 in R:

```
ma5 = filter(cmort, sides=2, rep(1,5)/5)
ma53 = filter(cmort, sides=2, rep(1,53)/53)
plot(cmort, type="p", ylab="mortality")
lines(ma5); lines(ma53)
```

[9] Sometimes, the end weights, a_{-k} and a_k are set equal to half the value of the other weights.

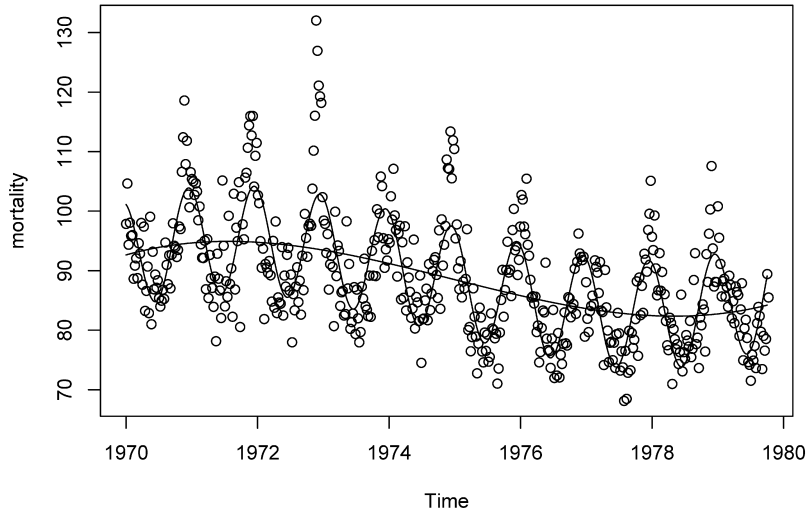

Fig. 2.12. The weekly cardiovascular mortality series with a cubic trend and cubic trend plus periodic regression.

Many other techniques are available for smoothing time series data based on methods from scatterplot smoothers. The general setup for a time plot is

$$x_t = f_t + y_t, \tag{2.51}$$

where f_t is some smooth function of time, and y_t is a stationary process. We may think of the moving average smoother m_t, given in (2.50), as an estimator of f_t. An obvious choice for f_t in (2.51) is polynomial regression

$$f_t = \beta_0 + \beta_1 t + \cdots + \beta_p t^p. \tag{2.52}$$

We have seen the results of a linear fit on the global temperature data in Example 2.1. For periodic data, one might employ periodic regression

$$\begin{aligned} f_t = {} & \alpha_0 + \alpha_1 \cos(2\pi\omega_1 t) + \beta_1 \sin(2\pi\omega_1 t) \\ & + \cdots + \alpha_p \cos(2\pi\omega_p t) + \beta_p \sin(2\pi\omega_p t), \end{aligned} \tag{2.53}$$

where $\omega_1, \ldots, \omega_p$ are distinct, specified frequencies. In addition, one might consider combining (2.52) and (2.53). These smoothers can be applied using classical linear regression.

Example 2.11 Polynomial and Periodic Regression Smoothers
Figure 2.12 shows the weekly mortality series with an estimated (via ordinary least squares) cubic smoother

$$\widehat{f}_t = \widehat{\beta}_0 + \widehat{\beta}_1 t + \widehat{\beta}_2 t^2 + \widehat{\beta}_3 t^3$$

superimposed to emphasize the trend, and an estimated (via ordinary least squares) cubic smoother plus a periodic regression

$$\widehat{f}_t = \widehat{\beta}_0 + \widehat{\beta}_1 t + \widehat{\beta}_2 t^2 + \widehat{\beta}_3 t^3 + \widehat{\alpha}_1 \cos(2\pi t/52) + \widehat{\alpha}_2 \sin(2\pi t/52)$$

superimposed to emphasize trend and seasonality.

The R commands for this example are as follows (we note that the sampling rate is $1/52$, so that `wk` below is essentially $t/52$).

```
wk = time(cmort) - mean(time(cmort))
wk2 = wk^2; wk3 = wk^3
cs = cos(2*pi*wk);  sn = sin(2*pi*wk)
reg1 = lm(cmort~wk + wk2 + wk3, na.action=NULL)
reg2 = lm(cmort~wk + wk2 + wk3 + cs + sn, na.action=NULL)
plot(cmort, type="p", ylab="mortality")
lines(fitted(reg1));  lines(fitted(reg2))
```

Modern regression techniques can be used to fit general smoothers to the pairs of points (t, x_t) where the estimate of f_t is smooth. Many of the techniques can easily be applied to time series data using the R or S-PLUS statistical packages; see Venables and Ripley (1994, Chapter 10) for details on applying these methods in S-PLUS (R is similar). A problem with the techniques used in Example 2.11 is that they assume f_t is the same function over the range of time, t; we might say that the technique is global. The moving average smoothers in Example 2.10 fit the data better because the technique is local; that is, moving average smoothers allow for the possibility that f_t is a different function over time. We describe some other local methods in the following examples.

Example 2.12 Kernel Smoothing

Kernel smoothing is a moving average smoother that uses a weight function, or kernel, to average the observations. Figure 2.13 shows kernel smoothing of the mortality series, where f_t in (2.51) is estimated by

$$\widehat{f}_t = \sum_{i=1}^{n} w_i(t) x_i, \tag{2.54}$$

where

$$w_i(t) = K\left(\tfrac{t-i}{b}\right) \Big/ \sum_{j=1}^{n} K\left(\tfrac{t-j}{b}\right). \tag{2.55}$$

are the weights and $K(\cdot)$ is a kernel function. This estimator, which was originally explored by Parzen (1962) and Rosenblatt (1956b), is often called the Nadaraya–Watson estimator (Watson, 1966); typically, the normal kernel, $K(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$, is used. To implement this in R, use the `ksmooth` function. The wider the bandwidth, b, the smoother the result. In Figure 2.13, the values of b for this example were $b = 5/52$ (roughly

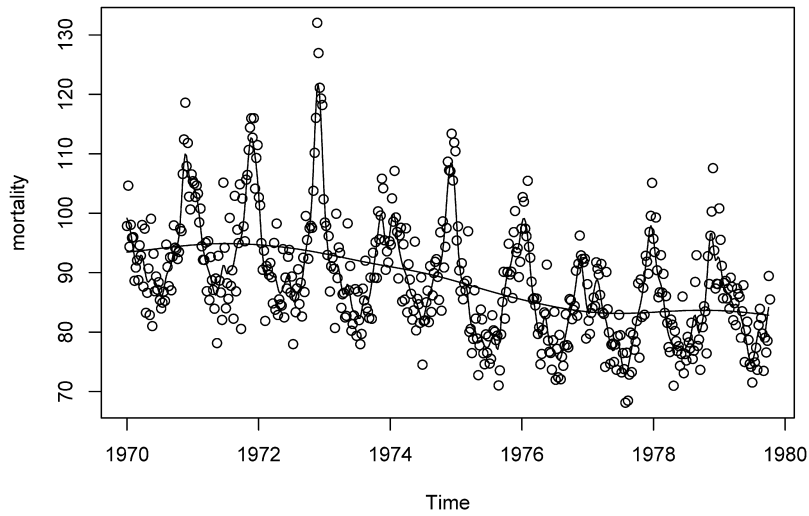

Fig. 2.13. Kernel smoothers of the mortality data.

weighted two to three week averages because $b/2$ is the inner quartile range of the kernel) for the seasonal component, and $b = 104/52 = 2$ (roughly weighted yearly averages) for the trend component.

Figure 2.13 can be reproduced in R (or S-PLUS) as follows.

```
plot(cmort, type="p", ylab="mortality")
lines(ksmooth(time(cmort), cmort, "normal", bandwidth=5/52))
lines(ksmooth(time(cmort), cmort, "normal", bandwidth=2))
```

Example 2.13 Lowess and Nearest Neighbor Regression

Another approach to smoothing a time plot is nearest neighbor regression. The technique is based on k-nearest neighbors linear regression, wherein one uses the data $\{x_{t-k/2}, \ldots, x_t, \ldots, x_{t+k/2}\}$ to predict x_t using linear regression; the result is $\widehat{f}_t$. For example, Figure 2.14 shows cardiovascular mortality and the nearest neighbor method using the R (or S-PLUS) smoother `supsmu`. We used $k = n/2$ to estimate the trend and $k = n/100$ to estimate the seasonal component. In general, `supsmu` uses a variable window for smoothing (see Friedman, 1984), but it can be used for correlated data by fixing the smoothing window, as was done here.

Lowess is a method of smoothing that is rather complex, but the basic idea is close to nearest neighbor regression. Figure 2.14 shows smoothing of mortality using the R or S-PLUS 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 the smoothed estimate of f_t. The larger the fraction of nearest neighbors included, the smoother the estimate

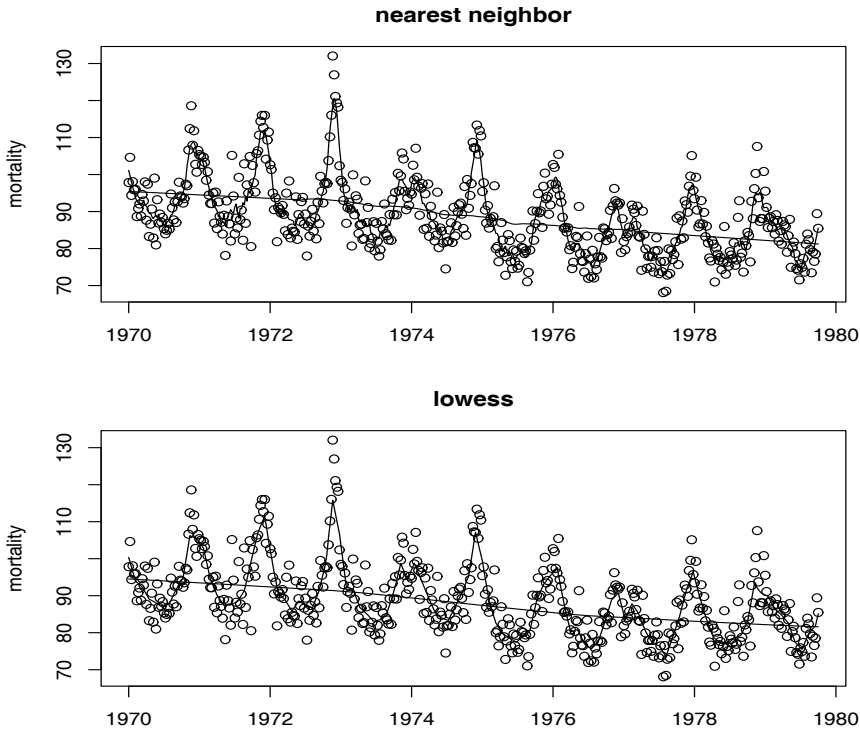

Fig. 2.14. Nearest neighbor (`supsmu`) and locally weighted regression (`lowess`) smoothers of the mortality data.

$\widehat{f_t}$ will be. In Figure 2.14, the smoother uses about two-thirds of the data to obtain an estimate of the trend component, and the seasonal component uses 2% of the data.

Figure 2.14 can be reproduced in R or S-PLUS as follows.

```
par(mfrow=c(2,1))
plot(cmort, type="p", ylab="mortality", main="nearest neighbor")
lines(supsmu(time(cmort), cmort, span=.5))
lines(supsmu(time(cmort), cmort, span=.01))
plot(cmort, type="p", ylab="mortality", main="lowess")
lines(lowess(cmort, f=.02));  lines(lowess(cmort, f=2/3))
```

Example 2.14 Smoothing Splines

An extension of polynomial regression is to first divide time $t = 1, \ldots, n$, into k intervals, $[t_0 = 1, t_1]$, $[t_1 + 1, t_2]$, $\ldots$, $[t_{k-1} + 1, t_k = n]$. The values $t_0, t_1, \ldots, t_k$ are called *knots*. Then, in each interval, one fits a regression of the form (2.52); typically, $p = 3$, and this is called cubic splines.

A related method is smoothing splines, which minimizes a compromise between the fit and the degree of smoothness given by

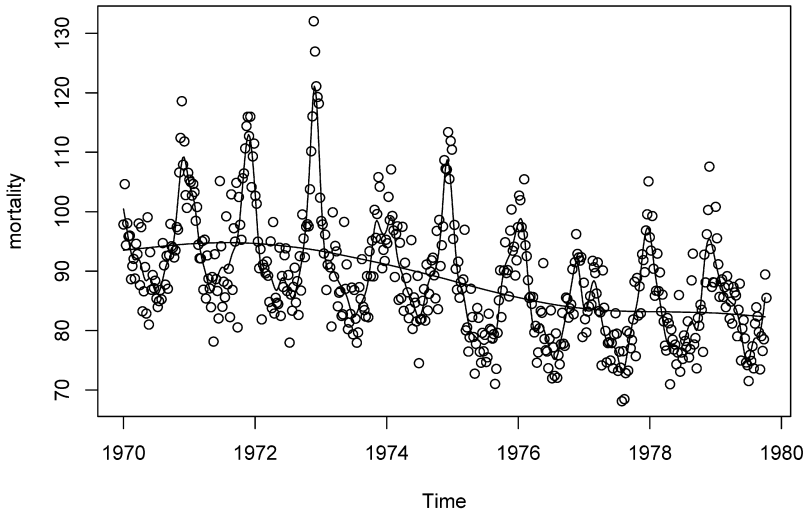

Fig. 2.15. Smoothing splines fit to the mortality data.

$$\sum_{t=1}^{n} [x_t - f_t]^2 + \lambda \int \left(f_t''\right)^2 dt, \tag{2.56}$$

where f_t is a cubic spline with a knot at each t. The degree of smoothness is controlled by $\lambda > 0$. There is a relationship between smoothing splines and state space models, which is investigated in Problem 6.7.

In R, the smoothing parameter is called `spar` and it is monotonically related to λ; type `?smooth.spline` to view the help file for details. Figure 2.15 shows smoothing spline fits on the mortality data using generalized cross-validation, which uses the data to "optimally" assess the smoothing parameter, for the seasonal component, and `spar=1` for the trend. The figure can be reproduced in R as follows.

```
plot(cmort, type="p", ylab="mortality")
lines(smooth.spline(time(cmort), cmort))
lines(smooth.spline(time(cmort), cmort, spar=1))
```

Example 2.15 Smoothing One Series as a Function of Another

In addition to smoothing time plots, smoothing techniques can be applied to smoothing a time series as a function of another time series. In this example, we smooth the scatterplot of two contemporaneously measured time series, mortality as a function of temperature. In Example 2.2, we discovered a nonlinear relationship between mortality and temperature. Continuing along these lines, Figure 2.16 shows scatterplots of mortality, M_t, and temperature, T_t, along with M_t smoothed as a function of T_t using lowess and using smoothing splines. In both cases, mortality increases at extreme

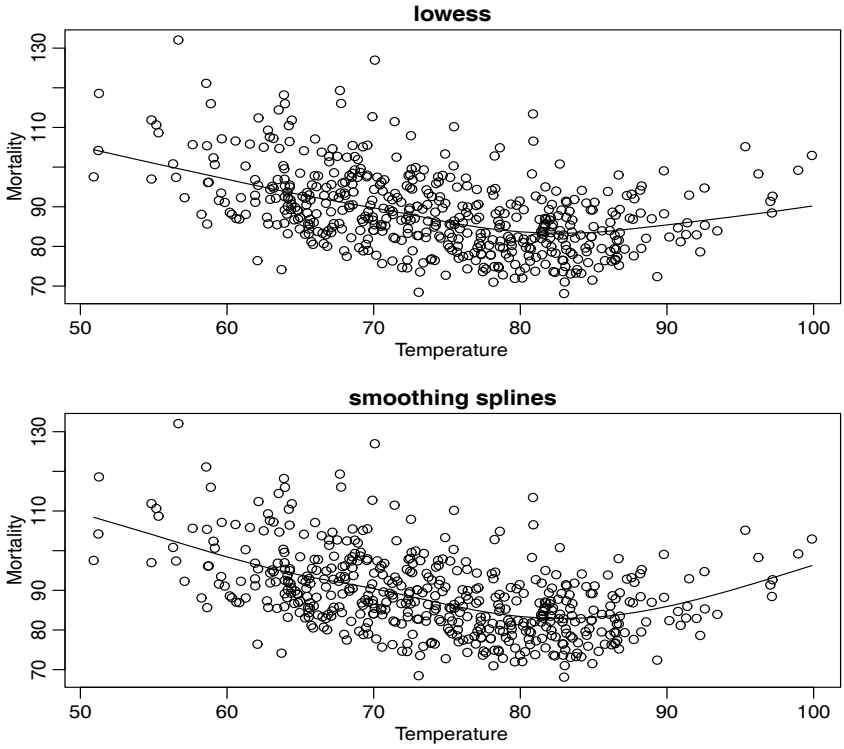

Fig. 2.16. Smoothers of mortality as a function of temperature using lowess and smoothing splines.

temperatures, but in an asymmetric way; mortality is higher at colder temperatures than at hotter temperatures. The minimum mortality rate seems to occur at approximately 80° F.

Figure 2.16 can be reproduced in R as follows.

```
1 par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
2 plot(tempr, cmort, main="lowess", xlab="Temperature",
      ylab="Mortality")
3 lines(lowess(tempr,cmort))
4 plot(tempr, cmort, main="smoothing splines", xlab="Temperature",
      ylab="Mortality")
5 lines(smooth.spline(tempr, cmort))
```

As a final word of caution, the methods mentioned in this section may not take into account the fact that the data are serially correlated, and most of the techniques have been designed for independent observations. That is, for example, the smoothers shown in Figure 2.16 are calculated under the false assumption that the pairs (M_t, T_t), are iid pairs of observations. In addition,

the degree of smoothness used in the previous examples were chosen arbitrarily to bring out what might be considered obvious features in the data set.

Problems

Section 2.2

2.1 For the Johnson & Johnson data, say y_t, shown in Figure 1.1, let $x_t = \log(y_t)$.

(a) Fit the 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 $Q_i(t) = 1$ 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. What is the interpretation of the parameters β, α_1, α_2, α_3, and α_4? (*Detailed code is given in Appendix R on page 574.*)

(b) What happens if you include an intercept term in the model in (a)?

(c) 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)?

2.2 For the mortality data examined in Example 2.2:

(a) Add another component to the regression in (2.25) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression in (2.25). State your conclusion.

(b) Draw a scatterplot matrix of M_t, T_t, P_t and P_{t-4} and then calculate the pairwise correlations between the series. Compare the relationship between M_t and P_t versus M_t and P_{t-4}.

2.3 Repeat the following exercise six times and then discuss the results. Generate a random walk with drift, (1.4), of length $n = 100$ with $\delta = .01$ and $\sigma_w = 1$. Call the data x_t for $t = 1, \ldots, 100$. Fit the regression $x_t = \beta t + w_t$ using least squares. Plot the data, the mean function (i.e., $\mu_t = .01\, t$) and the fitted line, $\widehat{x}_t = \widehat{\beta}\, t$, on the same graph. Discuss your results.

The following R code may be useful:

```
1 par(mfcol = c(3,2))  # set up graphics
2 for (i in 1:6){
3  x = ts(cumsum(rnorm(100,.01,1)))   # the data
4  reg = lm(x~0+time(x), na.action=NULL)  # the regression
5  plot(x) # plot data
6  lines(.01*time(x), col="red", lty="dashed") # plot mean
7  abline(reg, col="blue") }  # plot regression line
```

2.4 *Kullback-Leibler Information.* Given the random vector $\boldsymbol{y}$, we define the information for discriminating between two densities in the same family, indexed by a parameter $\boldsymbol{\theta}$, say $f(\boldsymbol{y};\boldsymbol{\theta}_1)$ and $f(\boldsymbol{y};\boldsymbol{\theta}_2)$, as

$$I(\boldsymbol{\theta}_1;\boldsymbol{\theta}_2) = \frac{1}{n} E_1 \log \frac{f(\boldsymbol{y};\boldsymbol{\theta}_1)}{f(\boldsymbol{y};\boldsymbol{\theta}_2)}, \tag{2.57}$$

where E_1 denotes expectation with respect to the density determined by $\boldsymbol{\theta}_1$. For the Gaussian regression model, the parameters are $\boldsymbol{\theta} = (\boldsymbol{\beta}', \sigma^2)'$. Show that we obtain

$$I(\boldsymbol{\theta}_1;\boldsymbol{\theta}_2) = \frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2} - \log\frac{\sigma_1^2}{\sigma_2^2} - 1\right) + \frac{1}{2}\frac{(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)' Z' Z (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)}{n\sigma_2^2} \tag{2.58}$$

in that case.

2.5 *Model Selection.* Both selection criteria (2.19) and (2.20) are derived from information theoretic arguments, based on the well-known Kullback-Leibler discrimination information numbers (see Kullback and Leibler, 1951, Kullback, 1958). We give an argument due to Hurvich and Tsai (1989). We think of the measure (2.58) as measuring the discrepancy between the two densities, characterized by the parameter values $\boldsymbol{\theta}_1' = (\boldsymbol{\beta}_1', \sigma_1^2)'$ and $\boldsymbol{\theta}_2' = (\boldsymbol{\beta}_2', \sigma_2^2)'$. Now, if the true value of the parameter vector is $\boldsymbol{\theta}_1$, we argue that the best model would be one that minimizes the discrepancy between the theoretical value and the sample, say $I(\boldsymbol{\theta}_1;\widehat{\boldsymbol{\theta}})$. Because $\boldsymbol{\theta}_1$ will not be known, Hurvich and Tsai (1989) considered finding an unbiased estimator for $E_1[I(\boldsymbol{\beta}_1, \sigma_1^2; \hat{\boldsymbol{\beta}}, \hat{\sigma}^2)]$, where

$$I(\boldsymbol{\beta}_1, \sigma_1^2; \hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = \frac{1}{2}\left(\frac{\sigma_1^2}{\hat{\sigma}^2} - \log\frac{\sigma_1^2}{\hat{\sigma}^2} - 1\right) + \frac{1}{2}\frac{(\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}})' Z' Z (\boldsymbol{\beta}_1 - \hat{\boldsymbol{\beta}})}{n\hat{\sigma}^2}$$

and $\boldsymbol{\beta}$ is a $k \times 1$ regression vector. Show that

$$E_1[I(\boldsymbol{\beta}_1, \sigma_1^2; \hat{\boldsymbol{\beta}}, \hat{\sigma}^2)] = \frac{1}{2}\left(-\log\sigma_1^2 + E_1 \log\widehat{\sigma}^2 + \frac{n+k}{n-k-2} - 1\right), \tag{2.59}$$

using the distributional properties of the regression coefficients and error variance. An unbiased estimator for $E_1 \log\widehat{\sigma}^2$ is $\log\widehat{\sigma}^2$. Hence, we have shown that the expectation of the above discrimination information is as claimed. As models with differing dimensions k are considered, only the second and third terms in (2.59) will vary and we only need unbiased estimators for those two terms. This gives the form of AICc quoted in (2.20) in the chapter. You will need the two distributional results

$$\frac{n\widehat{\sigma}^2}{\sigma_1^2} \sim \chi^2_{n-k} \quad \text{and} \quad \frac{(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_1)' Z' Z (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_1)}{\sigma_1^2} \sim \chi^2_k$$

The two quantities are distributed independently as chi-squared distributions with the indicated degrees of freedom. If $x \sim \chi^2_n$, $E(1/x) = 1/(n-2)$.

Section 2.3

2.6 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 $\gamma_y(h)$.

2.7 Show (2.31) is stationary.

2.8 The glacial varve record plotted in Figure 2.6 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? Hint: For $|p|$ close to zero, $\log(1+p) \approx p$; let $p = (y_t - y_{t-1})/y_{t-1}$.
(e) Based on the sample ACF of the differenced transformed series computed in (c), argue that a generalization of the model given by Example 1.23 might be reasonable. Assume

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

is stationary when the inputs w_t are assumed independent with mean 0 and variance σ_w^2. Show that

$$\gamma_u(h) = \begin{cases} \sigma_w^2(1+\theta^2) & \text{if } h = 0, \\ -\theta\ \sigma_w^2 & \text{if } h = \pm 1, \\ 0 & \text{if } |h| > 1. \end{cases}$$

(f) Based on part (e), use $\widehat{\rho}_u(1)$ and the estimate of the variance of u_t, $\widehat{\gamma}_u(0)$, to derive estimates of θ and σ_w^2. This is an application of the method of moments from classical statistics, where estimators of the parameters are derived by equating sample moments to theoretical moments.

2.9 In this problem, we will explore the periodic nature of S_t, the SOI series displayed in Figure 1.5.

(a) Detrend the series by fitting a regression of S_t on time t. Is there a significant trend in the sea surface temperature? Comment.
(b) Calculate the periodogram for the detrended series obtained in part (a). Identify the frequencies of the two main peaks (with an obvious one at the frequency of one cycle every 12 months). What is the probable El Niño cycle indicated by the minor peak?

2.10 Consider the model (2.46) used in Example 2.9,

$$x_t = \sum_{j=0}^{n} \beta_1(j/n)\cos(2\pi tj/n) + \beta_2(j/n)\sin(2\pi tj/n).$$

(a) Display the model design matrix Z [see (2.5)] for $n = 4$.
(b) Show numerically that the columns of Z in part (a) satisfy part (d) and then display $(Z'Z)^{-1}$ for this case.
(c) If x_1, x_2, x_3, x_4 are four observations, write the estimates of the four betas, $\beta_1(0), \beta_1(1/4), \beta_2(1/4), \beta_1(1/2)$, in terms of the observations.
(d) Verify that for any positive integer n and $j, k = 0, 1, \ldots, \llbracket n/2 \rrbracket$, where $\llbracket \cdot \rrbracket$ denotes the greatest integer function:[10]
  (i) Except for $j = 0$ or $j = n/2$,

$$\sum_{t=1}^{n} \cos^2(2\pi tj/n) = \sum_{t=1}^{n} \sin^2(2\pi tj/n) = n/2$$

  (ii) When $j = 0$ or $j = n/2$,

$$\sum_{t=1}^{n} \cos^2(2\pi tj/n) = n \quad \text{but} \quad \sum_{t=1}^{n} \sin^2(2\pi tj/n) = 0.$$

  (iii) For $j \neq k$,

$$\sum_{t=1}^{n} \cos(2\pi tj/n)\cos(2\pi tk/n) = \sum_{t=1}^{n} \sin(2\pi tj/n)\sin(2\pi tk/n) = 0.$$

  Also, for any j and k,

$$\sum_{t=1}^{n} \cos(2\pi tj/n)\sin(2\pi tk/n) = 0.$$

[10] Some useful facts: $2\cos(\alpha) = e^{i\alpha} + e^{-i\alpha}$, $2i\sin(\alpha) = e^{i\alpha} - e^{-i\alpha}$, and $\sum_{t=1}^{n} z^t = z(1 - z^n)/(1 - z)$ for $z \neq 1$.

Section 2.4

2.11 Consider the two weekly time series `oil` and `gas`. The oil series is in dollars per barrel, while the gas series is in cents per gallon; see Appendix R for details.

(a) Plot the data on the same graph. Which of the simulated series displayed in §1.3 do these series most resemble? Do you believe the series are stationary (explain your answer)?
(b) In economics, it is often the percentage change in price (termed *growth rate* or *return*), rather than the absolute price change, that is important. Argue that a transformation of the form $y_t = \nabla \log x_t$ might be applied to the data, where x_t is the oil or gas price series [see the hint in Problem 2.8(d)].
(c) Transform the data as described in part (b), plot the data on the same graph, look at the sample ACFs of the transformed data, and comment. [Hint: `poil = diff(log(oil))` and `pgas = diff(log(gas))`.]
(d) Plot the CCF of the transformed data and comment The small, but significant values when `gas` leads `oil` might be considered as feedback. [Hint: `ccf(poil, pgas)` will have `poil` leading for negative lag values.]
(e) Exhibit scatterplots of the oil and gas growth rate series for up to three weeks of lead time of oil prices; include a nonparametric smoother in each plot and comment on the results (e.g., Are there outliers? Are the relationships linear?). [Hint: `lag.plot2(poil, pgas, 3)`.]
(f) There have been a number of studies questioning whether gasoline prices respond more quickly when oil prices are rising than when oil prices are falling ("asymmetry"). We will attempt to explore this question here with simple lagged regression; we will ignore some obvious problems such as outliers and autocorrelated errors, so this will not be a definitive analysis. Let G_t and O_t denote the gas and oil growth rates.
   (i) Fit the regression (and comment on the results)

$$G_t = \alpha_1 + \alpha_2 I_t + \beta_1 O_t + \beta_2 O_{t-1} + w_t,$$

   where $I_t = 1$ if $O_t \geq 0$ and 0 otherwise (I_t is the indicator of no growth or positive growth in oil price). Hint:

```
indi = ifelse(poil < 0, 0, 1)
mess = ts.intersect(pgas, poil, poilL = lag(poil,-1), indi)
summary(fit <- lm(pgas~ poil + poilL + indi, data=mess))
```

   (ii) What is the fitted model when there is negative growth in oil price at time t? What is the fitted model when there is no or positive growth in oil price? Do these results support the asymmetry hypothesis?
   (iii) Analyze the residuals from the fit and comment.

2.12 Use two different smoothing techniques described in §2.4 to estimate the trend in the global temperature series displayed in Figure 1.2. Comment.

3

ARIMA Models

3.1 Introduction

In Chapters 1 and 2, we introduced autocorrelation and cross-correlation functions (ACFs and CCFs) as tools for clarifying relations that may occur within and between time series at various lags. In addition, we explained how to build linear models based on classical regression theory for exploiting the associations indicated by large values of the ACF or CCF. The time domain, or regression, methods of this chapter are appropriate when we are dealing with possibly nonstationary, shorter time series; these series are the rule rather than the exception in many applications. In addition, if the emphasis is on forecasting future values, then the problem is easily treated as a regression problem. This chapter develops a number of regression techniques for time series that are all related to classical ordinary and weighted or correlated least squares.

Classical regression is often insufficient for explaining all of the interesting dynamics of a time series. For example, the ACF of the residuals of the simple linear regression fit to the global temperature data (see Example 2.4 of Chapter 2) reveals additional structure in the data that the regression did not capture. Instead, the introduction of correlation as a phenomenon that may be generated through lagged linear relations leads to proposing the autoregressive (AR) and autoregressive moving average (ARMA) models. Adding nonstationary models to the mix leads to the autoregressive integrated moving average (ARIMA) model popularized in the landmark work by Box and Jenkins (1970). The Box–Jenkins method for identifying a plausible ARIMA model is given in this chapter along with techniques for parameter estimation and forecasting for these models. A partial theoretical justification of the use of ARMA models is discussed in Appendix B, §B.4.

R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications: With R Examples*, Springer Texts in Statistics, DOI 10.1007/978-1-4419-7865-3_3,

3.2 Autoregressive Moving Average Models

The classical regression model of Chapter 2 was developed for the static case, namely, we only allow the dependent variable to be influenced by current values of the independent variables. In the time series case, it is desirable to allow the dependent variable to be influenced by the past values of the independent variables and possibly by its own past values. If the present can be plausibly modeled in terms of only the past values of the independent inputs, we have the enticing prospect that forecasting will be possible.

Introduction to Autoregressive Models

Autoregressive models are based on the idea that the current value of the series, x_t, can be explained as a function of p past values, $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$, where p determines the number of steps into the past needed to forecast the current value. As a typical case, recall Example 1.10 in which data were generated using the model

$$x_t = x_{t-1} - .90x_{t-2} + w_t,$$

where w_t is white Gaussian noise with $\sigma_w^2 = 1$. We have now assumed the current value is a particular *linear* function of past values. The regularity that persists in Figure 1.9 gives an indication that forecasting for such a model might be a distinct possibility, say, through some version such as

$$x_{n+1}^n = x_n - .90x_{n-1},$$

where the quantity on the left-hand side denotes the forecast at the next period $n+1$ based on the observed data, $x_1, x_2, \ldots, x_n$. We will make this notion more precise in our discussion of forecasting (§3.5).

The extent to which it might be possible to forecast a real data series from its own past values can be assessed by looking at the autocorrelation function and the lagged scatterplot matrices discussed in Chapter 2. For example, the lagged scatterplot matrix for the Southern Oscillation Index (SOI), shown in Figure 2.7, gives a distinct indication that lags 1 and 2, for example, are linearly associated with the current value. The ACF shown in Figure 1.14 shows relatively large positive values at lags 1, 2, 12, 24, and 36 and large negative values at 18, 30, and 42. We note also the possible relation between the SOI and Recruitment series indicated in the scatterplot matrix shown in Figure 2.8. We will indicate in later sections on transfer function and vector AR modeling how to handle the dependence on values taken by other series.

The preceding discussion motivates the following definition.

Definition 3.1 *An* **autoregressive model** *of order p, abbreviated* **AR(p)**, *is of the form*

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \tag{3.1}$$

where x_t is stationary, and $\phi_1, \phi_2, \ldots, \phi_p$ are constants ($\phi_p \neq 0$). Although it is not necessary yet, we assume that w_t is a Gaussian white noise series with mean zero and variance σ_w^2, unless otherwise stated. The mean of x_t in (3.1) is zero. If the mean, μ, of x_t is not zero, replace x_t by $x_t - \mu$ in (3.1),

$$x_t - \mu = \phi_1(x_{t-1} - \mu) + \phi_2(x_{t-2} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + w_t,$$

or write

$$x_t = \alpha + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots + \phi_p x_{t-p} + w_t, \tag{3.2}$$

where $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$.

We note that (3.2) is similar to the regression model of §2.2, and hence the term auto (or self) regression. Some technical difficulties, however, develop from applying that model because the regressors, $x_{t-1}, \ldots, x_{t-p}$, are random components, whereas $\boldsymbol{z}_t$ was assumed to be fixed. A useful form follows by using the backshift operator (2.33) to write the AR(p) model, (3.1), as

$$(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p)x_t = w_t, \tag{3.3}$$

or even more concisely as

$$\phi(B)x_t = w_t. \tag{3.4}$$

The properties of $\phi(B)$ are important in solving (3.4) for x_t. This leads to the following definition.

Definition 3.2 *The* **autoregressive operator** *is defined to be*

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p. \tag{3.5}$$

We initiate the investigation of AR models by considering the first-order model, AR(1), given by $x_t = \phi x_{t-1} + w_t$. Iterating backwards k times, we get

$$\begin{aligned} x_t &= \phi x_{t-1} + w_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t \\ &= \phi^2 x_{t-2} + \phi w_{t-1} + w_t \\ &\vdots \\ &= \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j}. \end{aligned}$$

This method suggests that, by continuing to iterate backward, and provided that $|\phi| < 1$ and x_t is stationary, we can represent an AR(1) model as a linear process given by[1]

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}. \tag{3.6}$$

[1] Note that $\lim_{k\to\infty} E\left(x_t - \sum_{j=0}^{k-1} \phi^j w_{t-j}\right)^2 = \lim_{k\to\infty} \phi^{2k} E\left(x_{t-k}^2\right) = 0$, so (3.6) exists in the mean square sense (see Appendix A for a definition).

The AR(1) process defined by (3.6) is stationary with mean

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

and autocovariance function,

$$\begin{aligned}\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) &= E\left[\left(\sum_{j=0}^{\infty} \phi^j w_{t+h-j}\right)\left(\sum_{k=0}^{\infty} \phi^k w_{t-k}\right)\right] \\ &= E\left[\left(w_{t+h} + \cdots + \phi^h w_t + \phi^{h+1} w_{t-1} + \cdots\right)\left(w_t + \phi w_{t-1} + \cdots\right)\right] \\ &= \sigma_w^2 \sum_{j=0}^{\infty} \phi^{h+j}\phi^j = \sigma_w^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j} = \frac{\sigma_w^2 \phi^h}{1-\phi^2}, \quad h \geq 0.\end{aligned} \tag{3.7}$$

Recall that $\gamma(h) = \gamma(-h)$, so we will only exhibit the autocovariance function for $h \geq 0$. From (3.7), the ACF of an AR(1) is

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h, \quad h \geq 0, \tag{3.8}$$

and $\rho(h)$ satisfies the recursion

$$\rho(h) = \phi\, \rho(h-1), \quad h = 1, 2, \ldots . \tag{3.9}$$

We will discuss the ACF of a general AR(p) model in §3.4.

Example 3.1 The Sample Path of an AR(1) Process

Figure 3.1 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 with each other. This result means that observations at contiguous time points will tend to be close in value to each other; this fact shows up in the top of Figure 3.1 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 3.1, 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 3.1:

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

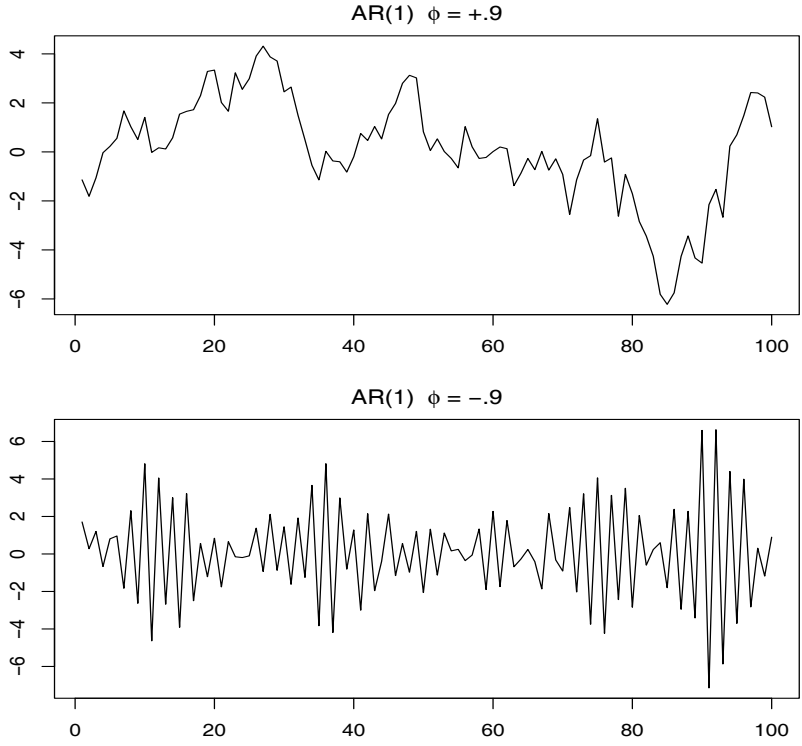

Fig. 3.1. Simulated AR(1) models: $\phi = .9$ (top); $\phi = -.9$ (bottom).

Example 3.2 Explosive AR Models and Causality

In Example 1.18, it was discovered that the random walk $x_t = x_{t-1} + w_t$ is not stationary. We might wonder whether there is a stationary AR(1) process with $|\phi| > 1$. Such processes are called explosive because the values of the time series quickly become large in magnitude. Clearly, because $|\phi|^j$ increases without bound as $j \to \infty$, $\sum_{j=0}^{k-1} \phi^j w_{t-j}$ will not converge (in mean square) as $k \to \infty$, so the intuition used to get (3.6) will not work directly. We can, however, modify that argument to obtain a stationary model as follows. Write $x_{t+1} = \phi x_t + w_{t+1}$, in which case,

$$\begin{aligned} x_t &= \phi^{-1} x_{t+1} - \phi^{-1} w_{t+1} = \phi^{-1} \left(\phi^{-1} x_{t+2} - \phi^{-1} w_{t+2} \right) - \phi^{-1} w_{t+1} \\ &\vdots \\ &= \phi^{-k} x_{t+k} - \sum_{j=1}^{k-1} \phi^{-j} w_{t+j}, \end{aligned} \tag{3.10}$$

by iterating forward k steps. Because $|\phi|^{-1} < 1$, this result suggests the stationary future dependent AR(1) model

$$x_t = -\sum_{j=1}^{\infty} \phi^{-j} w_{t+j}. \tag{3.11}$$

The reader can verify that this is stationary and of the AR(1) form $x_t = \phi x_{t-1} + w_t$. Unfortunately, this model is useless because it requires us to know the future to be able to predict the future. When a process does not depend on the future, such as the AR(1) when $|\phi| < 1$, we will say the process is causal. In the explosive case of this example, the process is stationary, but it is also future dependent, and not causal.

Example 3.3 Every Explosion Has a Cause

Excluding explosive models from consideration is not a problem because the models have causal counterparts. For example, if

$$x_t = \phi x_{t-1} + w_t \quad \text{with} \quad |\phi| > 1$$

and $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$, then using (3.11), $\{x_t\}$ is a non-causal stationary Gaussian process with $E(x_t) = 0$ and

$$\begin{aligned}\gamma_x(h) = \mathrm{cov}(x_{t+h}, x_t) &= \mathrm{cov}\left(-\sum_{j=1}^{\infty} \phi^{-j} w_{t+h+j}, -\sum_{k=1}^{\infty} \phi^{-k} w_{t+k}\right) \\ &= \sigma_w^2 \phi^{-2}\, \phi^{-h}/(1-\phi^{-2}).\end{aligned}$$

Thus, using (3.7), the causal process defined by

$$y_t = \phi^{-1} y_{t-1} + v_t$$

where $v_t \sim$ iid $\mathrm{N}(0, \sigma_w^2 \phi^{-2})$ is stochastically equal to the x_t process (i.e., all finite distributions of the processes are the same). For example, if $x_t = 2x_{t-1} + w_t$ with $\sigma_w^2 = 1$, then $y_t = \frac{1}{2} y_{t-1} + v_t$ with $\sigma_v^2 = 1/4$ is an equivalent causal process (see Problem 3.3). This concept generalizes to higher orders, but it is easier to show using Chapter 4 techniques; see Example 4.7.

The technique of iterating backward to get an idea of the stationary solution of AR models works well when $p = 1$, but not for larger orders. A general technique is that of matching coefficients. Consider the AR(1) model in operator form

$$\phi(B) x_t = w_t, \tag{3.12}$$

where $\phi(B) = 1 - \phi B$, and $|\phi| < 1$. Also, write the model in equation (3.6) using operator form as

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B) w_t, \tag{3.13}$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$ and $\psi_j = \phi^j$. Suppose we did not know that $\psi_j = \phi^j$. We could substitute $\psi(B)w_t$ from (3.13) for x_t in (3.12) to obtain

$$\phi(B)\psi(B)w_t = w_t. \tag{3.14}$$

The coefficients of B on the left-hand side of (3.14) must be equal to those on right-hand side of (3.14), which means

$$(1 - \phi B)(1 + \psi_1 B + \psi_2 B^2 + \cdots + \psi_j B^j + \cdots) = 1. \tag{3.15}$$

Reorganizing the coefficients in (3.15),

$$1 + (\psi_1 - \phi)B + (\psi_2 - \psi_1\phi)B^2 + \cdots + (\psi_j - \psi_{j-1}\phi)B^j + \cdots = 1,$$

we see that for each $j = 1, 2, \ldots$, the coefficient of B^j on the left must be zero because it is zero on the right. The coefficient of B on the left is $(\psi_1 - \phi)$, and equating this to zero, $\psi_1 - \phi = 0$, leads to $\psi_1 = \phi$. Continuing, the coefficient of B^2 is $(\psi_2 - \psi_1\phi)$, so $\psi_2 = \phi^2$. In general,

$$\psi_j = \psi_{j-1}\phi,$$

with $\psi_0 = 1$, which leads to the solution $\psi_j = \phi^j$.

Another way to think about the operations we just performed is to consider the AR(1) model in operator form, $\phi(B)x_t = w_t$. Now multiply both sides by $\phi^{-1}(B)$ (assuming the inverse operator exists) to get

$$\phi^{-1}(B)\phi(B)x_t = \phi^{-1}(B)w_t,$$

or

$$x_t = \phi^{-1}(B)w_t.$$

We know already that

$$\phi^{-1}(B) = 1 + \phi B + \phi^2 B^2 + \cdots + \phi^j B^j + \cdots,$$

that is, $\phi^{-1}(B)$ is $\psi(B)$ in (3.13). Thus, we notice that working with operators is like working with polynomials. That is, consider the polynomial $\phi(z) = 1 - \phi z$, where z is a complex number and $|\phi| < 1$. Then,

$$\phi^{-1}(z) = \frac{1}{(1 - \phi z)} = 1 + \phi z + \phi^2 z^2 + \cdots + \phi^j z^j + \cdots, \quad |z| \leq 1,$$

and the coefficients of B^j in $\phi^{-1}(B)$ are the same as the coefficients of z^j in $\phi^{-1}(z)$. In other words, we may treat the backshift operator, B, as a complex number, z. These results will be generalized in our discussion of ARMA models. We will find the polynomials corresponding to the operators useful in exploring the general properties of ARMA models.

Introduction to Moving Average Models

As an alternative to the autoregressive representation in which the x_t on the left-hand side of the equation are assumed to be combined linearly, the moving average model of order q, abbreviated as MA(q), assumes the white noise w_t on the right-hand side of the defining equation are combined linearly to form the observed data.

Definition 3.3 *The* **moving average model** *of order q, or* **MA(q)** *model, is defined to be*

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}, \tag{3.16}$$

where there are q lags in the moving average and $\theta_1, \theta_2, \ldots, \theta_q$ ($\theta_q \neq 0$) are parameters.[2] *Although it is not necessary yet, we assume that w_t is a Gaussian white noise series with mean zero and variance σ_w^2, unless otherwise stated.*

The system is the same as the infinite moving average defined as the linear process (3.13), where $\psi_0 = 1$, $\psi_j = \theta_j$, for $j = 1, \ldots, q$, and $\psi_j = 0$ for other values. We may also write the MA(q) process in the equivalent form

$$x_t = \theta(B) w_t, \tag{3.17}$$

using the following definition.

Definition 3.4 *The* **moving average operator** *is*

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q. \tag{3.18}$$

Unlike the autoregressive process, the moving average process is stationary for any values of the parameters $\theta_1, \ldots, \theta_q$; details of this result are provided in §3.4.

Example 3.4 The MA(1) Process

Consider the MA(1) model $x_t = w_t + \theta w_{t-1}$. Then, $E(x_t) = 0$,

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

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 3.1). 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)

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

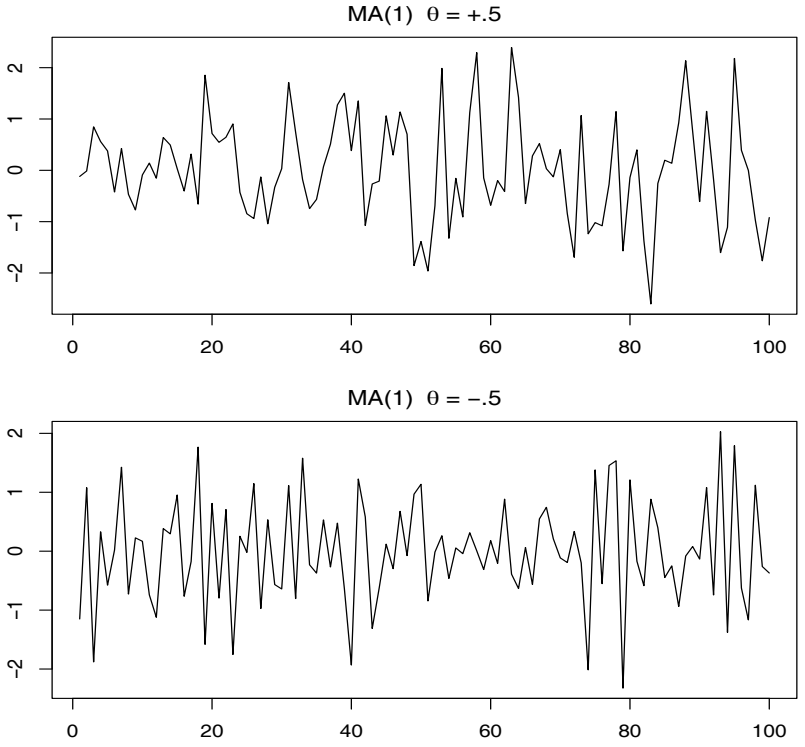

Fig. 3.2. Simulated MA(1) models: $\theta = .5$ (top); $\theta = -.5$ (bottom).

model in which the correlation between x_t and x_{t-k} is never zero. When $\theta = .5$, for example, x_t and x_{t-1} are positively correlated, and $\rho(1) = .4$. When $\theta = -.5$, x_t and x_{t-1} are negatively correlated, $\rho(1) = -.4$. Figure 3.2 shows a time plot of these two processes with $\sigma_w^2 = 1$. The series in Figure 3.2 where $\theta = .5$ is smoother than the series in Figure 3.2, where $\theta = -.5$.

A figure similar to Figure 3.2 can be created in R as follows:

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

Example 3.5 Non-uniqueness of MA Models and Invertibility

Using Example 3.4, we note that for an MA(1) model, $\rho(h)$ is the same for θ and $\frac{1}{\theta}$; try 5 and $\frac{1}{5}$, for example. In addition, 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

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

and

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

are the same because of normality (i.e., all finite distributions are 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 the criterion of 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 $w_t = -\theta w_{t-1} + x_t$. Following the steps that led to (3.6), if $|\theta| < 1$, then $w_t = \sum_{j=0}^{\infty} (-\theta)^j x_{t-j}$, which is the desired infinite AR 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.

As in the AR case, the polynomial, $\theta(z)$, corresponding to the moving average operators, $\theta(B)$, will be useful in exploring general properties of MA processes. For example, following the steps of equations (3.12)–(3.15), we can write the MA(1) model as $x_t = \theta(B)w_t$, where $\theta(B) = 1 + \theta B$. If $|\theta| < 1$, then we can write the model as $\pi(B)x_t = w_t$, where $\pi(B) = \theta^{-1}(B)$. Let $\theta(z) = 1 + \theta z$, for $|z| \le 1$, then $\pi(z) = \theta^{-1}(z) = 1/(1 + \theta z) = \sum_{j=0}^{\infty} (-\theta)^j z^j$, and we determine that $\pi(B) = \sum_{j=0}^{\infty} (-\theta)^j B^j$.

Autoregressive Moving Average Models

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

Definition 3.5 *A time series* $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ *is* **ARMA(p, q)** *if it is stationary and*

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}, \tag{3.19}$$

with $\phi_p \neq 0$, $\theta_q \neq 0$, *and* $\sigma_w^2 > 0$. *The parameters* p *and* q *are called the autoregressive and the moving average orders, respectively. If* x_t *has a nonzero mean* μ, *we set* $\alpha = \mu(1 - \phi_1 - \cdots - \phi_p)$ *and write the model as*

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}. \tag{3.20}$$

Although it is not necessary yet, we assume that w_t is a Gaussian white noise series with mean zero and variance σ_w^2, unless otherwise stated.

As previously noted, 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). To aid in the investigation of ARMA models, it will be useful to write them using the AR operator, (3.5), and the MA operator, (3.18). In particular, the ARMA(p, q) model in (3.19) can then be written in concise form as

$$\phi(B)x_t = \theta(B)w_t. \tag{3.21}$$

Before we discuss the conditions under which (3.19) is causal and invertible, we point out a potential problem with the ARMA model.

Example 3.6 Parameter Redundancy

Consider a white noise process $x_t = w_t$. Equivalently, we can write this as $.5x_{t-1} = .5w_{t-1}$ by shifting back one unit of time and multiplying by .5. Now, subtract the two representations to obtain

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

or

$$x_t = .5x_{t-1} - .5w_{t-1} + w_t, \tag{3.22}$$

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 (3.22)], but we have hidden the fact that x_t is white noise because of the parameter redundancy or over-parameterization. Write the parameter redundant model in operator form as $\phi(B)x_t = \theta(B)w_t$, or

$$(1 - .5B)x_t = (1 - .5B)w_t.$$

Apply the operator $\phi(B)^{-1} = (1 - .5B)^{-1}$ to both sides to obtain

$$x_t = (1 - .5B)^{-1}(1 - .5B)x_t = (1 - .5B)^{-1}(1 - .5B)w_t = w_t,$$

which is the original model. We can easily detect the problem of over-parameterization with the use of the operators or their associated polynomials. That is, write the AR polynomial $\phi(z) = (1 - .5z)$, the MA polynomial $\theta(z) = (1 - .5z)$, and note that both polynomials have a common factor, namely $(1 - .5z)$. This common factor immediately identifies the parameter redundancy. Discarding the common factor in each leaves $\phi(z) = 1$ and $\theta(z) = 1$, from which we conclude $\phi(B) = 1$ and $\theta(B) = 1$, and we deduce that the model is actually white noise. The consideration of parameter redundancy will be crucial when we discuss estimation for general ARMA models. As this example points out, we might fit an ARMA$(1, 1)$ model to white noise data and find that the parameter estimates are significant. If we were unaware of parameter redundancy, we might claim the data are correlated when in fact they are not (Problem 3.20).

Examples 3.2, 3.5, and 3.6 point to a number of problems with the general definition of ARMA(p, q) models, as given by (3.19), or, equivalently, by (3.21). To summarize, we have seen the following problems:

(i) parameter redundant models,
(ii) stationary AR models that depend on the future, and
(iii) MA models that are not unique.

To overcome these problems, we will require some additional restrictions on the model parameters. First, we make the following definitions.

Definition 3.6 *The* **AR and MA polynomials** *are defined as*

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p, \quad \phi_p \neq 0, \tag{3.23}$$

and

$$\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q, \quad \theta_q \neq 0, \tag{3.24}$$

respectively, where z is a complex number.

To address the first problem, we will henceforth refer to an ARMA(p, q) model to mean that it is in its simplest form. That is, in addition to the original definition given in equation (3.19), we will also require that $\phi(z)$ and $\theta(z)$ have no common factors. So, the process, $x_t = .5x_{t-1} - .5w_{t-1} + w_t$, discussed in Example 3.6 is not referred to as an ARMA$(1, 1)$ process because, in its reduced form, x_t is white noise.

To address the problem of future-dependent models, we formally introduce the concept of causality.

Definition 3.7 *An ARMA(p, q) model is said to be* **causal***, if the time series $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ can be written as a one-sided linear process:*

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} = \psi(B) w_t, \tag{3.25}$$

where $\psi(B) = \sum_{j=0}^{\infty} \psi_j B^j$, and $\sum_{j=0}^{\infty} |\psi_j| < \infty$; we set $\psi_0 = 1$.

In Example 3.2, the AR(1) process, $x_t = \phi x_{t-1} + w_t$, is causal only when $|\phi| < 1$. Equivalently, the process is causal only when the root of $\phi(z) = 1 - \phi z$ is bigger than one in absolute value. That is, the root, say, z_0, of $\phi(z)$ is $z_0 = 1/\phi$ (because $\phi(z_0) = 0$) and $|z_0| > 1$ because $|\phi| < 1$. In general, we have the following property.

Property 3.1 Causality of an ARMA(p, q) Process

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 (3.25) can be determined by solving

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}, \quad |z| \leq 1.$$

Another way to phrase Property 3.1 is that *an ARMA process is causal only when the roots of $\phi(z)$ lie outside the unit circle*; that is, $\phi(z) = 0$ only when $|z| > 1$. Finally, to address the problem of uniqueness discussed in Example 3.5, we choose the model that allows an infinite autoregressive representation.

Definition 3.8 *An ARMA(p, q) model is said to be* **invertible**, *if the time series $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ can be written as*

$$\pi(B)x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} = w_t, \tag{3.26}$$

where $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$, and $\sum_{j=0}^{\infty} |\pi_j| < \infty$; we set $\pi_0 = 1$.

Analogous to Property 3.1, we have the following property.

Property 3.2 Invertibility of an ARMA(p, q) Process
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 (3.26) can be determined by solving

$$\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\phi(z)}{\theta(z)}, \quad |z| \leq 1.$$

Another way to phrase Property 3.2 is that *an ARMA process is invertible only when the roots of $\theta(z)$ lie outside the unit circle*; that is, $\theta(z) = 0$ only when $|z| > 1$. The proof of Property 3.1 is given in Appendix B (the proof of Property 3.2 is similar and, hence, is not provided). The following examples illustrate these concepts.

Example 3.7 Parameter Redundancy, Causality, Invertibility
Consider 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, the associated polynomials

$$\phi(z) = 1 - .4z - .45z^2 = (1 + .5z)(1 - .9z)$$

$$\theta(z) = (1 + z + .25z^2) = (1 + .5z)^2$$

have a common factor that can be canceled. After cancellation, the polynomials become $\phi(z) = (1 - .9z)$ and $\theta(z) = (1 + .5z)$, so the model is an ARMA$(1, 1)$ model, $(1 - .9B)x_t = (1 + .5B)w_t$, or

$$x_t = .9x_{t-1} + .5w_{t-1} + w_t. \tag{3.27}$$

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 3.1, $\phi(z)\psi(z) = \theta(z)$, or

$$(1 - .9z)(\psi_0 + \psi_1 z + \psi_2 z^2 + \cdots) = (1 + .5z).$$

Matching coefficients we get $\psi_0 = 1$, $\psi_1 = .5 + .9 = 1.4$, and $\psi_j = .9\psi_{j-1}$ for $j > 1$. Thus, $\psi_j = 1.4(.9)^{j-1}$ for $j \geq 1$ and (3.27) can be written as

$$x_t = w_t + 1.4 \sum_{j=1}^{\infty} .9^{j-1} w_{t-j}.$$

Similarly, the invertible representation using Property 3.2 is

$$x_t = 1.4 \sum_{j=1}^{\infty} (-.5)^{j-1} x_{t-j} + w_t.$$

Example 3.8 Causal Conditions for an AR(2) Process

For an AR(1) model, $(1-\phi B)x_t = w_t$, to be causal, the root of $\phi(z) = 1-\phi z$ must lie outside of the unit circle. In this case, the root (or zero) occurs at $z_0 = 1/\phi$ [i.e., $\phi(z_0) = 0$], so it is easy to go from the causal requirement on the root, $|1/\phi| > 1$, to a requirement on the parameter, $|\phi| < 1$. It is not so easy to establish this relationship for higher order models.

For example, 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. 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. If we denote those roots by z_1 and z_2, we can write $\phi(z) = (1-z_1^{-1}z)(1-z_2^{-1}z)$; note that $\phi(z_1) = \phi(z_2) = 0$. The model can be written in operator form as $(1-z_1^{-1}B)(1-z_2^{-1}B)x_t = w_t$. From this representation, it follows that $\phi_1 = (z_1^{-1} + z_2^{-1})$ and $\phi_2 = -(z_1 z_2)^{-1}$. This relationship and the fact that $|z_1| > 1$ and $|z_2| > 1$ can be used to establish the following equivalent condition for causality:

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1. \tag{3.28}$$

This causality condition specifies a triangular region in the parameter space. We leave the details of the equivalence to the reader (Problem 3.5).

3.3 Difference Equations

The study of the behavior of ARMA processes and their ACFs is greatly enhanced by a basic knowledge of difference equations, simply because they are difference equations. This topic is also useful in the study of time domain models and stochastic processes in general. We will give a brief and heuristic account of the topic along with some examples of the usefulness of the theory. For details, the reader is referred to Mickens (1990).

Suppose we have a sequence of numbers $u_0, u_1, u_2, \ldots$ such that

$$u_n - \alpha u_{n-1} = 0, \quad \alpha \neq 0, \quad n = 1, 2, \ldots \ . \tag{3.29}$$

For example, recall (3.9) in which we showed that the ACF of an AR(1) process is a sequence, $\rho(h)$, satisfying

$$\rho(h) - \phi\rho(h-1) = 0, \quad h = 1, 2, \ldots \ .$$

Equation (3.29) represents a homogeneous difference equation of order 1. To solve the equation, we write:

$$\begin{aligned} u_1 &= \alpha u_0 \\ u_2 &= \alpha u_1 = \alpha^2 u_0 \\ &\vdots \\ u_n &= \alpha u_{n-1} = \alpha^n u_0. \end{aligned}$$

Given an initial condition $u_0 = c$, we may solve (3.29), namely, $u_n = \alpha^n c$.

In operator notation, (3.29) can be written as $(1 - \alpha B)u_n = 0$. The polynomial associated with (3.29) is $\alpha(z) = 1 - \alpha z$, and the root, say, z_0, of this polynomial is $z_0 = 1/\alpha$; that is $\alpha(z_0) = 0$. We know a solution (in fact, *the* solution) to (3.29), with initial condition $u_0 = c$, is

$$u_n = \alpha^n c = \left(z_0^{-1}\right)^n c. \tag{3.30}$$

That is, the solution to the difference equation (3.29) depends only on the initial condition and the inverse of the root to the associated polynomial $\alpha(z)$.

Now suppose that the sequence satisfies

$$u_n - \alpha_1 u_{n-1} - \alpha_2 u_{n-2} = 0, \quad \alpha_2 \neq 0, \quad n = 2, 3, \ldots \tag{3.31}$$

This equation is a homogeneous difference equation of order 2. The corresponding polynomial is

$$\alpha(z) = 1 - \alpha_1 z - \alpha_2 z^2,$$

which has two roots, say, z_1 and z_2; that is, $\alpha(z_1) = \alpha(z_2) = 0$. We will consider two cases. First suppose $z_1 \neq z_2$. Then the general solution to (3.31) is

$$u_n = c_1 z_1^{-n} + c_2 z_2^{-n}, \tag{3.32}$$

where c_1 and c_2 depend on the initial conditions. The claim that is *a* solution can be verified by direct substitution of (3.32) into (3.31):

$$\begin{aligned}
&\left(c_1 z_1^{-n} + c_2 z_2^{-n}\right) - \alpha_1\left(c_1 z_1^{-(n-1)} + c_2 z_2^{-(n-1)}\right) - \alpha_2\left(c_1 z_1^{-(n-2)} + c_2 z_2^{-(n-2)}\right) \\
&\quad = c_1 z_1^{-n}\left(1 - \alpha_1 z_1 - \alpha_2 z_1^2\right) + c_2 z_2^{-n}\left(1 - \alpha_1 z_2 - \alpha_2 z_2^2\right) \\
&\quad = c_1 z_1^{-n}\alpha(z_1) + c_2 z_2^{-n}\alpha(z_2) = 0.
\end{aligned}$$

Given two initial conditions u_0 and u_1, we may solve for c_1 and c_2:

$$u_0 = c_1 + c_2 \quad \text{and} \quad u_1 = c_1 z_1^{-1} + c_2 z_2^{-1},$$

where z_1 and z_2 can be solved for in terms of α_1 and α_2 using the quadratic formula, for example.

When the roots are equal, $z_1 = z_2\ (= z_0)$, a general solution to (3.31) is

$$u_n = z_0^{-n}(c_1 + c_2 n). \tag{3.33}$$

This claim can also be verified by direct substitution of (3.33) into (3.31):

$$\begin{aligned}
&z_0^{-n}(c_1 + c_2 n) - \alpha_1\left(z_0^{-(n-1)}[c_1 + c_2(n-1)]\right) - \alpha_2\left(z_0^{-(n-2)}[c_1 + c_2(n-2)]\right) \\
&\quad = z_0^{-n}(c_1 + c_2 n)\left(1 - \alpha_1 z_0 - \alpha_2 z_0^2\right) + c_2 z_0^{-n+1}\left(\alpha_1 + 2\alpha_2 z_0\right) \\
&\quad = c_2 z_0^{-n+1}\left(\alpha_1 + 2\alpha_2 z_0\right).
\end{aligned}$$

To show that $(\alpha_1 + 2\alpha_2 z_0) = 0$, write $1 - \alpha_1 z - \alpha_2 z^2 = (1 - z_0^{-1} z)^2$, and take derivatives with respect to z on both sides of the equation to obtain $(\alpha_1 + 2\alpha_2 z) = 2z_0^{-1}(1 - z_0^{-1} z)$. Thus, $(\alpha_1 + 2\alpha_2 z_0) = 2z_0^{-1}(1 - z_0^{-1} z_0) = 0$, as was to be shown. Finally, given two initial conditions, u_0 and u_1, we can solve for c_1 and c_2:

$$u_0 = c_1 \quad \text{and} \quad u_1 = (c_1 + c_2) z_0^{-1}.$$

It can also be shown that these solutions are unique.

To summarize these results, in the case of distinct roots, the solution to the homogeneous difference equation of degree two was

$$\begin{aligned}
u_n = {} & z_1^{-n} \times (\text{a polynomial in } n \text{ of degree } m_1 - 1) \\
& + \ z_2^{-n} \times (\text{a polynomial in } n \text{ of degree } m_2 - 1),
\end{aligned} \tag{3.34}$$

where m_1 is the multiplicity of the root z_1 and m_2 is the multiplicity of the root z_2. In this example, of course, $m_1 = m_2 = 1$, and we called the polynomials of degree zero c_1 and c_2, respectively. In the case of the repeated root, the solution was

$$u_n = z_0^{-n} \times (\text{a polynomial in } n \text{ of degree } m_0 - 1), \tag{3.35}$$

where m_0 is the multiplicity of the root z_0; that is, $m_0 = 2$. In this case, we wrote the polynomial of degree one as $c_1 + c_2 n$. In both cases, we solved for c_1 and c_2 given two initial conditions, u_0 and u_1.

Example 3.9 The ACF of an AR(2) Process

Suppose $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$ is a causal AR(2) process. Multiply each side of the model by x_{t-h} for $h > 0$, and take expectation:

$$E(x_t x_{t-h}) = \phi_1 E(x_{t-1} x_{t-h}) + \phi_2 E(x_{t-2} x_{t-h}) + E(w_t x_{t-h}).$$

The result is

$$\gamma(h) = \phi_1 \gamma(h-1) + \phi_2 \gamma(h-2), \quad h = 1, 2, \ldots . \tag{3.36}$$

In (3.36), we used the fact that $E(x_t) = 0$ and for $h > 0$,

$$E(w_t x_{t-h}) = E\Big(w_t \sum_{j=0}^{\infty} \psi_j w_{t-h-j}\Big) = 0.$$

Divide (3.36) through by $\gamma(0)$ to obtain the difference equation for the ACF of the process:

$$\rho(h) - \phi_1 \rho(h-1) - \phi_2 \rho(h-2) = 0, \quad h = 1, 2, \ldots . \tag{3.37}$$

The initial conditions are $\rho(0) = 1$ and $\rho(-1) = \phi_1/(1-\phi_2)$, which is obtained by evaluating (3.37) for $h = 1$ and noting that $\rho(1) = \rho(-1)$.

Using the results for the homogeneous difference equation of order two, let z_1 and z_2 be the roots of the associated polynomial, $\phi(z) = 1 - \phi_1 z - \phi_2 z^2$. Because the model is causal, we know the roots are outside the unit circle: $|z_1| > 1$ and $|z_2| > 1$. Now, consider the solution for three cases:

(i) When z_1 and z_2 are real and distinct, then

$$\rho(h) = c_1 z_1^{-h} + c_2 z_2^{-h},$$

so $\rho(h) \to 0$ exponentially fast as $h \to \infty$.

(ii) When $z_1 = z_2\,(= z_0)$ are real and equal, then

$$\rho(h) = z_0^{-h}(c_1 + c_2 h),$$

so $\rho(h) \to 0$ exponentially fast as $h \to \infty$.

(iii) When $z_1 = \bar{z}_2$ are a complex conjugate pair, then $c_2 = \bar{c}_1$ (because $\rho(h)$ is real), and

$$\rho(h) = c_1 z_1^{-h} + \bar{c}_1 \bar{z}_1^{-h}.$$

Write c_1 and z_1 in polar coordinates, for example, $z_1 = |z_1| e^{i\theta}$, where θ is the angle whose tangent is the ratio of the imaginary part and the real part of z_1 (sometimes called $\arg(z_1)$; the range of θ is $[-\pi, \pi]$). Then, using the fact that $e^{i\alpha} + e^{-i\alpha} = 2\cos(\alpha)$, the solution has the form

$$\rho(h) = a|z_1|^{-h} \cos(h\theta + b),$$

where a and b are determined by the initial conditions. Again, $\rho(h)$ dampens to zero exponentially fast as $h \to \infty$, but it does so in a sinusoidal fashion. The implication of this result is shown in the next example.

Example 3.10 An AR(2) with Complex Roots

Figure 3.3 shows $n = 144$ observations from the AR(2) model

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

with $\sigma_w^2 = 1$, and with complex roots 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 §3.4, Figure 3.4.

To calculate the roots of the polynomial and solve for arg in R:

```
z = c(1,-1.5,.75)      # coefficients of the polynomial
(a = polyroot(z)[1])   # print one root: 1+0.57735i = 1 + i/sqrt(3)
arg = Arg(a)/(2*pi)    # arg in cycles/pt
1/arg                  # = 12,  the pseudo period
```

To reproduce Figure 3.3:

```
set.seed(90210)
ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
plot(1:144/12, ar2, type="l", xlab="Time (one unit = 12 points)")
abline(v=0:12, lty="dotted", lwd=2)
```

To calculate and display the ACF for this model:

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

We now exhibit the solution for the general homogeneous difference equation of order p:

$$u_n - \alpha_1 u_{n-1} - \cdots - \alpha_p u_{n-p} = 0, \quad \alpha_p \neq 0, \quad n = p, p+1, \ldots . \tag{3.38}$$

The associated polynomial is

$$\alpha(z) = 1 - \alpha_1 z - \cdots - \alpha_p z^p.$$

Suppose $\alpha(z)$ has r distinct roots, z_1 with multiplicity m_1, z_2 with multiplicity m_2, …, and z_r with multiplicity m_r, such that $m_1 + m_2 + \cdots + m_r = p$. The general solution to the difference equation (3.38) is

$$u_n = z_1^{-n} P_1(n) + z_2^{-n} P_2(n) + \cdots + z_r^{-n} P_r(n), \tag{3.39}$$

where $P_j(n)$, for $j = 1, 2, \ldots, r$, is a polynomial in n, of degree $m_j - 1$. Given p initial conditions $u_0, \ldots, u_{p-1}$, we can solve for the $P_j(n)$ explicitly.

Example 3.11 The ψ-weights for an ARMA Model

For a causal ARMA(p, q) model, $\phi(B)x_t = \theta(B)w_t$, where the zeros of $\phi(z)$ are outside the unit circle, recall that we may write

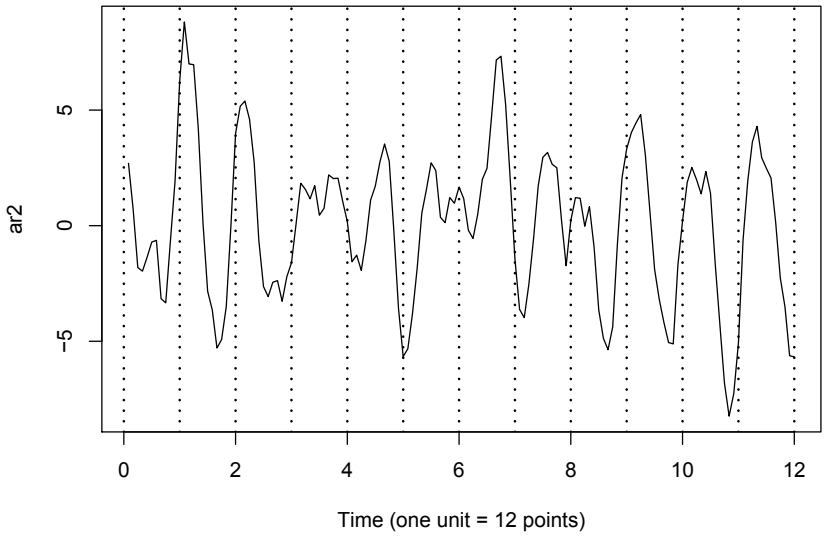

Fig. 3.3. Simulated AR(2) model, $n = 144$ with $\phi_1 = 1.5$ and $\phi_2 = -.75$.

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

where the ψ-weights are determined using Property 3.1.

For the pure MA(q) model, $\psi_0 = 1$, $\psi_j = \theta_j$, for $j = 1, \ldots, q$, and $\psi_j = 0$, otherwise. For the general case of ARMA(p, q) models, the task of solving for the ψ-weights is much more complicated, as was demonstrated in Example 3.7. The use of the theory of homogeneous difference equations can help here. To solve for the ψ-weights in general, we must match the coefficients in $\phi(z)\psi(z) = \theta(z)$:

$$(1 - \phi_1 z - \phi_2 z^2 - \cdots)(\psi_0 + \psi_1 z + \psi_2 z^2 + \cdots) = (1 + \theta_1 z + \theta_2 z^2 + \cdots).$$

The first few values are

$$\begin{aligned} \psi_0 &= 1 \\ \psi_1 - \phi_1\psi_0 &= \theta_1 \\ \psi_2 - \phi_1\psi_1 - \phi_2\psi_0 &= \theta_2 \\ \psi_3 - \phi_1\psi_2 - \phi_2\psi_1 - \phi_3\psi_0 &= \theta_3 \\ &\vdots \end{aligned}$$

where we would take $\phi_j = 0$ for $j > p$, and $\theta_j = 0$ for $j > q$. The ψ-weights satisfy the homogeneous difference equation given by

$$\psi_j - \sum_{k=1}^{p} \phi_k \psi_{j-k} = 0, \quad j \geq \max(p, q+1), \tag{3.40}$$

with initial conditions

$$\psi_j - \sum_{k=1}^{j} \phi_k \psi_{j-k} = \theta_j, \quad 0 \le j < \max(p, q+1). \tag{3.41}$$

The general solution depends on the roots of the AR polynomial $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$, as seen from (3.40). The specific solution will, of course, depend on the initial conditions.

Consider the ARMA process given in (3.27), $x_t = .9x_{t-1} + .5w_{t-1} + w_t$. Because $\max(p, q+1) = 2$, using (3.41), we have $\psi_0 = 1$ and $\psi_1 = .9 + .5 = 1.4$. By (3.40), for $j = 2, 3, \ldots$, the ψ-weights satisfy $\psi_j - .9\psi_{j-1} = 0$. The general solution is $\psi_j = c\,.9^j$. To find the specific solution, use the initial condition $\psi_1 = 1.4$, so $1.4 = .9c$ or $c = 1.4/.9$. Finally, $\psi_j = 1.4(.9)^{j-1}$, for $j \ge 1$, as we saw in Example 3.7.

To view, for example, the first 50 ψ-weights in R, use:

```
ARMAtoMA(ar=.9, ma=.5, 50)          # for a list
plot(ARMAtoMA(ar=.9, ma=.5, 50))    # for a graph
```

3.4 Autocorrelation and Partial Autocorrelation

We begin by exhibiting the ACF of an MA(q) process, $x_t = \theta(B)w_t$, where $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$. Because x_t is a finite linear combination of white noise terms, the process is stationary with mean

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

where we have written $\theta_0 = 1$, and with autocovariance function

$$\begin{aligned} \gamma(h) = \operatorname{cov}(x_{t+h}, x_t) &= \operatorname{cov}\Big(\sum_{j=0}^{q} \theta_j w_{t+h-j}, \sum_{k=0}^{q} \theta_k w_{t-k} \Big) \\ &= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h} \theta_j \theta_{j+h}, & 0 \le h \le q \\ 0 & h > q. \end{cases} \end{aligned} \tag{3.42}$$

Recall that $\gamma(h) = \gamma(-h)$, so we will only display the values for $h \ge 0$. The cutting off of $\gamma(h)$ after q lags is the signature of the MA(q) model. Dividing (3.42) by $\gamma(0)$ yields the ACF of an MA(q):

$$\rho(h) = \begin{cases} \dfrac{\sum_{j=0}^{q-h} \theta_j \theta_{j+h}}{1 + \theta_1^2 + \cdots + \theta_q^2} & 1 \le h \le q \\ 0 & h > q. \end{cases} \tag{3.43}$$

For a causal ARMA(p, q) model, $\phi(B)x_t = \theta(B)w_t$, where the zeros of $\phi(z)$ are outside the unit circle, write

$$x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}. \tag{3.44}$$

It follows immediately that $E(x_t) = 0$. Also, the autocovariance function of x_t can be written as

$$\gamma(h) = \mathrm{cov}(x_{t+h}, x_t) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+h}, \quad h \geq 0. \tag{3.45}$$

We could then use (3.40) and (3.41) to solve for the ψ-weights. In turn, we could solve for $\gamma(h)$, and the ACF $\rho(h) = \gamma(h)/\gamma(0)$. As in Example 3.9, it is also possible to obtain a homogeneous difference equation directly in terms of $\gamma(h)$. First, we write

$$\begin{aligned} \gamma(h) &= \mathrm{cov}(x_{t+h}, x_t) = \mathrm{cov}\Big(\sum_{j=1}^{p} \phi_j x_{t+h-j} + \sum_{j=0}^{q} \theta_j w_{t+h-j},\, x_t \Big) \\ &= \sum_{j=1}^{p} \phi_j \gamma(h-j) + \sigma_w^2 \sum_{j=h}^{q} \theta_j \psi_{j-h}, \quad h \geq 0, \end{aligned} \tag{3.46}$$

where we have used the fact that, for $h \geq 0$,

$$\mathrm{cov}(w_{t+h-j},\, x_t) = \mathrm{cov}\Big(w_{t+h-j},\, \sum_{k=0}^{\infty} \psi_k w_{t-k} \Big) = \psi_{j-h} \sigma_w^2.$$

From (3.46), we can write a general homogeneous equation for the ACF of a causal ARMA process:

$$\gamma(h) - \phi_1 \gamma(h-1) - \cdots - \phi_p \gamma(h-p) = 0, \quad h \geq \max(p, q+1), \tag{3.47}$$

with initial conditions

$$\gamma(h) - \sum_{j=1}^{p} \phi_j \gamma(h-j) = \sigma_w^2 \sum_{j=h}^{q} \theta_j \psi_{j-h}, \quad 0 \leq h < \max(p, q+1). \tag{3.48}$$

Dividing (3.47) and (3.48) through by $\gamma(0)$ will allow us to solve for the ACF, $\rho(h) = \gamma(h)/\gamma(0)$.

Example 3.12 The ACF of an AR(p)

In Example 3.9 we considered the case where $p = 2$. For the general case, it follows immediately from (3.47) that

$$\rho(h) - \phi_1 \rho(h-1) - \cdots - \phi_p \rho(h-p) = 0, \quad h \geq p. \tag{3.49}$$

Let $z_1, \ldots, z_r$ denote the roots of $\phi(z)$, each with multiplicity $m_1, \ldots, m_r$, respectively, where $m_1 + \cdots + m_r = p$. Then, from (3.39), the general solution is

$$\rho(h) = z_1^{-h} P_1(h) + z_2^{-h} P_2(h) + \cdots + z_r^{-h} P_r(h), \quad h \geq p, \tag{3.50}$$

where $P_j(h)$ is a polynomial in h of degree $m_j - 1$.

Recall that for a causal model, all of the roots are outside the unit circle, $|z_i| > 1$, for $i = 1, \ldots, r$. If all the roots are real, then $\rho(h)$ dampens exponentially fast to zero as $h \to \infty$. If some of the roots are complex, then they will be in conjugate pairs and $\rho(h)$ will dampen, in a sinusoidal fashion, exponentially fast to zero as $h \to \infty$. In the case of complex roots, the time series will appear to be cyclic in nature. This, of course, is also true for ARMA models in which the AR part has complex roots.

Example 3.13 The ACF of an ARMA$(1,1)$

Consider the ARMA$(1,1)$ process $x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$, where $|\phi| < 1$. Based on (3.47), the autocovariance function satisfies

$$\gamma(h) - \phi\gamma(h-1) = 0, \quad h = 2, 3, \ldots,$$

and it follows from (3.29)–(3.30) that the general solution is

$$\gamma(h) = c\,\phi^h, \quad h = 1, 2, \ldots\ . \tag{3.51}$$

To obtain the initial conditions, we use (3.48):

$$\gamma(0) = \phi\gamma(1) + \sigma_w^2[1 + \theta\phi + \theta^2] \quad \text{and} \quad \gamma(1) = \phi\gamma(0) + \sigma_w^2\theta.$$

Solving for $\gamma(0)$ and $\gamma(1)$, we obtain:

$$\gamma(0) = \sigma_w^2 \frac{1 + 2\theta\phi + \theta^2}{1 - \phi^2} \quad \text{and} \quad \gamma(1) = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}.$$

To solve for c, note that from (3.51), $\gamma(1) = c\,\phi$ or $c = \gamma(1)/\phi$. Hence, the specific solution for $h \geq 1$ is

$$\gamma(h) = \frac{\gamma(1)}{\phi}\,\phi^h = \sigma_w^2 \frac{(1 + \theta\phi)(\phi + \theta)}{1 - \phi^2}\,\phi^{h-1}.$$

Finally, dividing through by $\gamma(0)$ yields the ACF

$$\rho(h) = \frac{(1 + \theta\phi)(\phi + \theta)}{1 + 2\theta\phi + \theta^2}\,\phi^{h-1}, \quad h \geq 1. \tag{3.52}$$

Notice that the general pattern of $\rho(h)$ in (3.52) is not different from that of an AR(1) given in (3.8). 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.

The Partial Autocorrelation Function (PACF)

We have seen in (3.43), 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).

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

$$\begin{aligned}\gamma_x(2) = \operatorname{cov}(x_t, x_{t-2}) &= \operatorname{cov}(\phi x_{t-1} + w_t, x_{t-2}) \\ &= \operatorname{cov}(\phi^2 x_{t-2} + \phi w_{t-1} + w_t, x_{t-2}) = \phi^2 \gamma_x(0).\end{aligned}$$

This result follows from causality because x_{t-2} involves $\{w_{t-2}, w_{t-3}, \ldots\}$, which are all uncorrelated with w_t and w_{t-1}. 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 partial out) the effect 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}. In fact,

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

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.

To formally define the PACF for mean-zero stationary time series, let $\widehat{x}_{t+h}$, for $h \geq 2$, denote the regression[3] of x_{t+h} on $\{x_{t+h-1}, x_{t+h-2}, \ldots, x_{t+1}\}$, which we write as

$$\widehat{x}_{t+h} = \beta_1 x_{t+h-1} + \beta_2 x_{t+h-2} + \cdots + \beta_{h-1} x_{t+1}. \tag{3.53}$$

No intercept term is needed in (3.53) because the mean of x_t is zero (otherwise, replace x_t by $x_t - \mu_x$ in this discussion). In addition, let $\widehat{x}_t$ denote the regression of x_t on $\{x_{t+1}, x_{t+2}, \ldots, x_{t+h-1}\}$, then

$$\widehat{x}_t = \beta_1 x_{t+1} + \beta_2 x_{t+2} + \cdots + \beta_{h-1} x_{t+h-1}. \tag{3.54}$$

Because of stationarity, the coefficients, $\beta_1, \ldots, \beta_{h-1}$ are the same in (3.53) and (3.54); we will explain this result in the next section.

[3] The term regression here refers to regression in the population sense. That is, $\widehat{x}_{t+h}$ is the linear combination of $\{x_{t+h-1}, x_{t+h-2}, \ldots, x_{t+1}\}$ that minimizes the mean squared error $E(x_{t+h} - \sum_{j=1}^{h-1} \alpha_j x_{t+j})^2$.

Definition 3.9 *The* **partial autocorrelation function (PACF)** *of a stationary process, x_t, denoted ϕ_{hh}, for $h = 1, 2, \ldots$, is*

$$\phi_{11} = \text{corr}(x_{t+1}, x_t) = \rho(1) \tag{3.55}$$

and

$$\phi_{hh} = \text{corr}(x_{t+h} - \widehat{x}_{t+h},\, x_t - \widehat{x}_t), \quad h \geq 2. \tag{3.56}$$

Both $(x_{t+h} - \widehat{x}_{t+h})$ and $(x_t - \widehat{x}_t)$ are uncorrelated with $\{x_{t+1}, \ldots, x_{t+h-1}\}$. The PACF, ϕ_{hh}, is the correlation between x_{t+h} and x_t with the linear dependence of $\{x_{t+1}, \ldots, x_{t+h-1}\}$ on each, removed. If the process x_t is Gaussian, then $\phi_{hh} = \text{corr}(x_{t+h}, x_t \mid x_{t+1}, \ldots, x_{t+h-1})$; that is, ϕ_{hh} is the correlation coefficient between x_{t+h} and x_t in the bivariate distribution of (x_{t+h}, x_t) conditional on $\{x_{t+1}, \ldots, x_{t+h-1}\}$.

Example 3.14 The PACF of an AR(1)

Consider the PACF of the AR(1) process given by $x_t = \phi x_{t-1} + w_t$, with $|\phi| < 1$. By definition, $\phi_{11} = \rho(1) = \phi$. To calculate ϕ_{22}, consider the regression of x_{t+2} on x_{t+1}, say, $\widehat{x}_{t+2} = \beta x_{t+1}$. We choose β to minimize

$$E(x_{t+2} - \widehat{x}_{t+2})^2 = E(x_{t+2} - \beta x_{t+1})^2 = \gamma(0) - 2\beta\gamma(1) + \beta^2\gamma(0).$$

Taking derivatives with respect to β and setting the result equal to zero, we have $\beta = \gamma(1)/\gamma(0) = \rho(1) = \phi$. Next, consider the regression of x_t on x_{t+1}, say $\widehat{x}_t = \beta x_{t+1}$. We choose β to minimize

$$E(x_t - \widehat{x}_t)^2 = E(x_t - \beta x_{t+1})^2 = \gamma(0) - 2\beta\gamma(1) + \beta^2\gamma(0).$$

This is the same equation as before, so $\beta = \phi$. Hence,

$$\begin{aligned}\phi_{22} &= \text{corr}(x_{t+2} - \widehat{x}_{t+2}, x_t - \widehat{x}_t) = \text{corr}(x_{t+2} - \phi x_{t+1}, x_t - \phi x_{t+1}) \\ &= \text{corr}(w_{t+2}, x_t - \phi x_{t+1}) = 0\end{aligned}$$

by causality. Thus, $\phi_{22} = 0$. In the next example, we will see that in this case, $\phi_{hh} = 0$ for all $h > 1$.

Example 3.15 The PACF of an AR(p)

The model implies $x_{t+h} = \sum_{j=1}^{p} \phi_j x_{t+h-j} + w_{t+h}$, where the roots of $\phi(z)$ are outside the unit circle. When $h > p$, the regression of x_{t+h} on $\{x_{t+1}, \ldots, x_{t+h-1}\}$, is

$$\widehat{x}_{t+h} = \sum_{j=1}^{p} \phi_j x_{t+h-j}.$$

We have not proved this obvious result yet, but we will prove it in the next section. Thus, when $h > p$,

$$\phi_{hh} = \text{corr}(x_{t+h} - \widehat{x}_{t+h},\, x_t - \widehat{x}_t) = \text{corr}(w_{t+h},\, x_t - \widehat{x}_t) = 0,$$

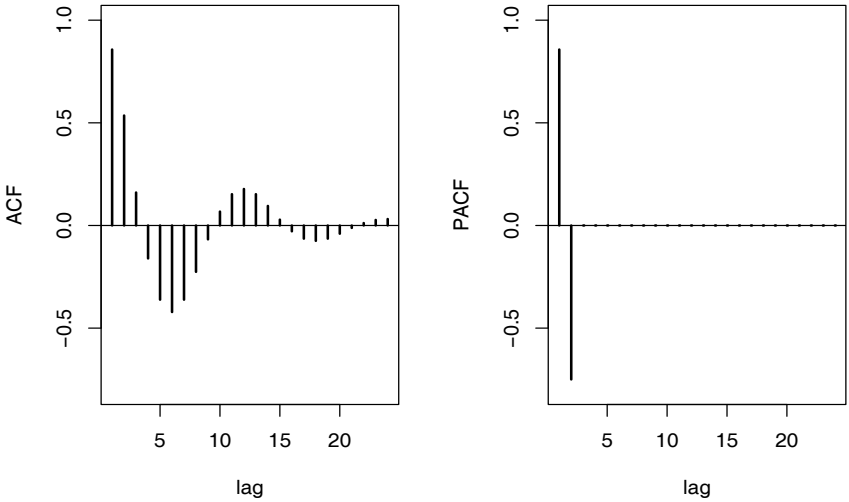

Fig. 3.4. The ACF and PACF of an AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -.75$.

because, by causality, $x_t - \widehat{x}_t$ depends only on $\{w_{t+h-1}, w_{t+h-2}, \ldots\}$; recall equation (3.54). When $h \leq p$, ϕ_{pp} is not zero, and $\phi_{11}, \ldots, \phi_{p-1,p-1}$ are not necessarily zero. We will see later that, in fact, $\phi_{pp} = \phi_p$. Figure 3.4 shows the ACF and the PACF of the AR(2) model presented in Example 3.10.

To reproduce Figure 3.4 in R, use the following commands:

```
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=c(1,2))
plot(ACF, type="h", xlab="lag", ylim=c(-.8,1));  abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.8,1));  abline(h=0)
```

Example 3.16 The PACF of an Invertible MA(q)

For an invertible MA(q), we can write $x_t = -\sum_{j=1}^{\infty} \pi_j x_{t-j} + w_t$. 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 $|\theta| < 1$, calculations similar to Example 3.14 will yield $\phi_{22} = -\theta^2/(1 + \theta^2 + \theta^4)$. For the MA(1) in general, we can show that

$$\phi_{hh} = -\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2(h+1)}}, \quad h \geq 1.$$

In the next section, we will discuss methods of calculating the PACF. 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 3.1.

Table 3.1. 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

Example 3.17 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 3.5 are consistent with the behavior of 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 3.1, these results suggest that a second-order ($p = 2$) autoregressive model might provide a good fit. Although we will discuss estimation in detail in §3.6, we ran a regression (see §2.2) 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 a model of the form

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

for $t = 3, 4, \ldots, 453$. The values of the estimates were $\widehat{\phi}_0 = 6.74_{(1.11)}$, $\widehat{\phi}_1 = 1.35_{(.04)}$, $\widehat{\phi}_2 = -.46_{(.04)}$, and $\widehat{\sigma}_w^2 = 89.72$, where the estimated standard errors are in parentheses.

The following R code can be used for this analysis. We use the script `acf2` to print and plot the ACF and PACF; see Appendix R for details.

```
acf2(rec, 48)      # will produce values and a graphic
(regr = ar.ols(rec, order=2, demean=FALSE, intercept=TRUE))
regr$asy.se.coef  # standard errors of the estimates
```

3.5 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 collected to the present, $\boldsymbol{x} = \{x_n, x_{n-1}, \ldots, x_1\}$. Throughout this section, we will assume x_t is stationary and the model parameters are known. The problem of forecasting when the model parameters are unknown will be discussed in the next section; also, see Problem 3.26. The minimum mean square error predictor of x_{n+m} is

$$x_{n+m}^n = E(x_{n+m} \mid \boldsymbol{x}) \tag{3.57}$$

because the conditional expectation minimizes the mean square error

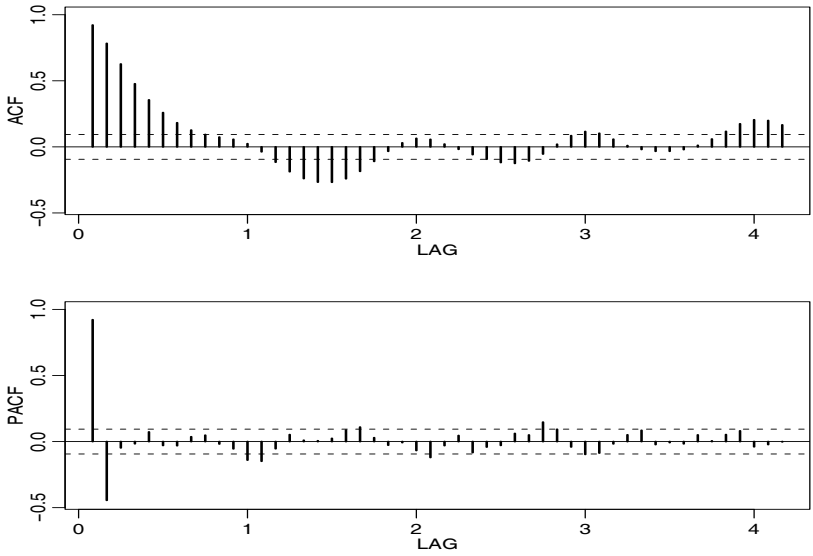

Fig. 3.5. ACF and PACF of the Recruitment series. Note that the lag axes are in terms of season (12 months in this case).

$$E\left[x_{n+m} - g(\boldsymbol{x})\right]^2, \tag{3.58}$$

where $g(\boldsymbol{x})$ is a function of the observations $\boldsymbol{x}$; see Problem 3.14.

First, we will restrict attention to predictors that are linear functions of the data, that is, predictors of the form

$$x^n_{n+m} = \alpha_0 + \sum_{k=1}^{n} \alpha_k x_k, \tag{3.59}$$

where $\alpha_0, \alpha_1, \ldots, \alpha_n$ are real numbers. Linear predictors of the form (3.59) that minimize the mean square prediction error (3.58) are called best linear predictors (BLPs). As we shall see, linear prediction depends only on the second-order moments of the process, which are easy to estimate from the data. Much of the material in this section is enhanced by the theoretical material presented in Appendix B. For example, Theorem B.3 states that if the process is Gaussian, minimum mean square error predictors and best linear predictors are the same. The following property, which is based on the Projection Theorem, Theorem B.1 of Appendix B, is a key result.

Property 3.3 Best Linear Prediction for Stationary Processes

Given data $x_1, \ldots, x_n$*, the best linear predictor,* $x^n_{n+m} = \alpha_0 + \sum_{k=1}^{n} \alpha_k x_k$*, of* x_{n+m}*, for* $m \geq 1$*, is found by solving*

$$E\left[\left(x_{n+m} - x^n_{n+m}\right) x_k\right] = 0, \quad k = 0, 1, \ldots, n, \tag{3.60}$$

where $x_0 = 1$*, for* $\alpha_0, \alpha_1, \ldots \alpha_n$*.*

The equations specified in (3.60) are called the prediction equations, and they are used to solve for the coefficients $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$. If $E(x_t) = \mu$, the first equation ($k = 0$) of (3.60) implies

$$E(x^n_{n+m}) = E(x_{n+m}) = \mu.$$

Thus, taking expectation in (3.59), we have

$$\mu = \alpha_0 + \sum_{k=1}^{n} \alpha_k \mu \qquad \text{or} \qquad \alpha_0 = \mu\Big(1 - \sum_{k=1}^{n} \alpha_k\Big).$$

Hence, the form of the BLP is

$$x^n_{n+m} = \mu + \sum_{k=1}^{n} \alpha_k (x_k - \mu).$$

Thus, until we discuss estimation, there is no loss of generality in considering the case that $\mu = 0$, in which case, $\alpha_0 = 0$.

First, consider one-step-ahead prediction. That is, given $\{x_1, \ldots, x_n\}$, we wish to forecast the value of the time series at the next time point, x_{n+1}. The BLP of x_{n+1} is of the form

$$x^n_{n+1} = \phi_{n1} x_n + \phi_{n2} x_{n-1} + \cdots + \phi_{nn} x_1, \tag{3.61}$$

where, for purposes that will become clear shortly, we have written α_k in (3.59), as $\phi_{n,n+1-k}$ in (3.61), for $k = 1, \ldots, n$. Using Property 3.3, the coefficients $\{\phi_{n1}, \phi_{n2}, \ldots, \phi_{nn}\}$ satisfy

$$E\Big[\Big(x_{n+1} - \sum_{j=1}^{n} \phi_{nj} x_{n+1-j}\Big) x_{n+1-k}\Big] = 0, \quad k = 1, \ldots, n,$$

or

$$\sum_{j=1}^{n} \phi_{nj} \gamma(k-j) = \gamma(k), \quad k = 1, \ldots, n. \tag{3.62}$$

The prediction equations (3.62) can be written in matrix notation as

$$\Gamma_n \boldsymbol{\phi}_n = \boldsymbol{\gamma}_n, \tag{3.63}$$

where $\Gamma_n = \{\gamma(k-j)\}_{j,k=1}^n$ is an $n \times n$ matrix, $\boldsymbol{\phi}_n = (\phi_{n1}, \ldots, \phi_{nn})'$ is an $n \times 1$ vector, and $\boldsymbol{\gamma}_n = (\gamma(1), \ldots, \gamma(n))'$ is an $n \times 1$ vector.

The matrix Γ_n is nonnegative definite. If Γ_n is singular, there are many solutions to (3.63), but, by the Projection Theorem (Theorem B.1), x^n_{n+1} is unique. If Γ_n is nonsingular, the elements of $\boldsymbol{\phi}_n$ are unique, and are given by

$$\boldsymbol{\phi}_n = \Gamma_n^{-1} \boldsymbol{\gamma}_n. \tag{3.64}$$

For ARMA models, the fact that $\sigma_w^2 > 0$ and $\gamma(h) \to 0$ as $h \to \infty$ is enough to ensure that Γ_n is positive definite (Problem 3.12). It is sometimes convenient to write the one-step-ahead forecast in vector notation

$$x_{n+1}^n = \boldsymbol{\phi}_n' \boldsymbol{x}, \tag{3.65}$$

where $\boldsymbol{x} = (x_n, x_{n-1}, \ldots, x_1)'$.

The mean square one-step-ahead prediction error is

$$P_{n+1}^n = E(x_{n+1} - x_{n+1}^n)^2 = \gamma(0) - \boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{\gamma}_n. \tag{3.66}$$

To verify (3.66) using (3.64) and (3.65),

$$\begin{aligned} E(x_{n+1} - x_{n+1}^n)^2 &= E(x_{n+1} - \boldsymbol{\phi}_n' \boldsymbol{x})^2 = E(x_{n+1} - \boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{x})^2 \\ &= E(x_{n+1}^2 - 2\boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{x} x_{n+1} + \boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{x}\boldsymbol{x}' \Gamma_n^{-1} \boldsymbol{\gamma}_n) \\ &= \gamma(0) - 2\boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{\gamma}_n + \boldsymbol{\gamma}_n' \Gamma_n^{-1} \Gamma_n \Gamma_n^{-1} \boldsymbol{\gamma}_n \\ &= \gamma(0) - \boldsymbol{\gamma}_n' \Gamma_n^{-1} \boldsymbol{\gamma}_n. \end{aligned}$$

Example 3.18 Prediction for an AR(2)

Suppose we have a causal AR(2) process $x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$, and one observation x_1. Then, using equation (3.64), the one-step-ahead prediction of x_2 based on x_1 is

$$x_2^1 = \phi_{11} x_1 = \frac{\gamma(1)}{\gamma(0)} x_1 = \rho(1) x_1.$$

Now, suppose we want the one-step-ahead prediction of x_3 based on two observations x_1 and x_2; i.e., $x_3^2 = \phi_{21} x_2 + \phi_{22} x_1$. We could use (3.62)

$$\begin{aligned} \phi_{21}\gamma(0) + \phi_{22}\gamma(1) &= \gamma(1) \\ \phi_{21}\gamma(1) + \phi_{22}\gamma(0) &= \gamma(2) \end{aligned}$$

to solve for ϕ_{21} and ϕ_{22}, or use the matrix form in (3.64) and solve

$$\begin{pmatrix} \phi_{21} \\ \phi_{22} \end{pmatrix} = \begin{pmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{pmatrix}^{-1} \begin{pmatrix} \gamma(1) \\ \gamma(2) \end{pmatrix},$$

but, it should be apparent from the model that $x_3^2 = \phi_1 x_2 + \phi_2 x_1$. Because $\phi_1 x_2 + \phi_2 x_1$ satisfies the prediction equations (3.60),

$$\begin{aligned} E\{[x_3 - (\phi_1 x_2 + \phi_2 x_1)] x_1\} &= E(w_3 x_1) = 0, \\ E\{[x_3 - (\phi_1 x_2 + \phi_2 x_1)] x_2\} &= E(w_3 x_2) = 0, \end{aligned}$$

it follows that, indeed, $x_3^2 = \phi_1 x_2 + \phi_2 x_1$, and by the uniqueness of the coefficients in this case, that $\phi_{21} = \phi_1$ and $\phi_{22} = \phi_2$. Continuing in this way, it is easy to verify that, for $n \geq 2$,

$$x_{n+1}^n = \phi_1 x_n + \phi_2 x_{n-1}.$$

That is, $\phi_{n1} = \phi_1, \phi_{n2} = \phi_2$, and $\phi_{nj} = 0$, for $j = 3, 4, \ldots, n$.

From Example 3.18, it should be clear (Problem 3.40) that, if the time series is a causal AR(p) process, then, for $n \geq p$,

$$x_{n+1}^n = \phi_1 x_n + \phi_2 x_{n-1} + \cdots + \phi_p x_{n-p+1}. \tag{3.67}$$

For ARMA models in general, the prediction equations will not be as simple as the pure AR case. In addition, for n large, the use of (3.64) is prohibitive because it requires the inversion of a large matrix. There are, however, iterative solutions that do not require any matrix inversion. In particular, we mention the recursive solution due to Levinson (1947) and Durbin (1960).

Property 3.4 The Durbin–Levinson Algorithm

Equations (3.64) and (3.66) can be solved iteratively as follows:

$$\phi_{00} = 0, \quad P_1^0 = \gamma(0). \tag{3.68}$$

For $n \geq 1$,

$$\phi_{nn} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k}\ \rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k}\ \rho(k)}, \quad P_{n+1}^n = P_n^{n-1}(1 - \phi_{nn}^2), \tag{3.69}$$

where, for $n \geq 2$,

$$\phi_{nk} = \phi_{n-1,k} - \phi_{nn}\phi_{n-1,n-k}, \quad k = 1, 2, \ldots, n-1. \tag{3.70}$$

The proof of Property 3.4 is left as an exercise; see Problem 3.13.

Example 3.19 Using the Durbin–Levinson Algorithm

To use the algorithm, start with $\phi_{00} = 0$, $P_1^0 = \gamma(0)$. Then, for $n = 1$,

$$\phi_{11} = \rho(1), \quad P_2^1 = \gamma(0)[1 - \phi_{11}^2].$$

For $n = 2$,

$$\phi_{22} = \frac{\rho(2) - \phi_{11}\ \rho(1)}{1 - \phi_{11}\ \rho(1)},\ \phi_{21} = \phi_{11} - \phi_{22}\phi_{11},$$
$$P_3^2 = P_2^1[1 - \phi_{22}^2] = \gamma(0)[1 - \phi_{11}^2][1 - \phi_{22}^2].$$

For $n = 3$,

$$\phi_{33} = \frac{\rho(3) - \phi_{21}\ \rho(2) - \phi_{22}\ \rho(1)}{1 - \phi_{21}\ \rho(1) - \phi_{22}\ \rho(2)},$$
$$\phi_{32} = \phi_{22} - \phi_{33}\phi_{21},\ \phi_{31} = \phi_{21} - \phi_{33}\phi_{22},$$
$$P_4^3 = P_3^2[1 - \phi_{33}^2] = \gamma(0)[1 - \phi_{11}^2][1 - \phi_{22}^2][1 - \phi_{33}^2],$$

and so on. Note that, in general, the standard error of the one-step-ahead forecast is the square root of

$$P_{n+1}^n = \gamma(0) \prod_{j=1}^{n} [1 - \phi_{jj}^2]. \tag{3.71}$$

An important consequence of the Durbin–Levinson algorithm is (see Problem 3.13) as follows.

Property 3.5 Iterative Solution for the PACF

The PACF of a stationary process x_t, can be obtained iteratively via (3.69) as ϕ_{nn}, for $n = 1, 2, \ldots$.

Using Property 3.5 and putting $n = p$ in (3.61) and (3.67), it follows that for an AR(p) model,

$$\begin{aligned} x_{p+1}^{p} &= \phi_{p1}\, x_p + \phi_{p2}\, x_{p-1} + \cdots + \phi_{pp}\, x_1 \\ &= \phi_1\, x_p + \phi_2\, x_{p-1} + \cdots + \phi_p\, x_1. \end{aligned} \tag{3.72}$$

Result (3.72) shows that for an AR(p) model, the partial autocorrelation coefficient at lag p, ϕ_{pp}, is also the last coefficient in the model, ϕ_p, as was claimed in Example 3.15.

Example 3.20 The PACF of an AR(2)

We will use the results of Example 3.19 and Property 3.5 to calculate the first three values, ϕ_{11}, ϕ_{22}, ϕ_{33}, of the PACF. Recall from Example 3.9 that $\rho(h) - \phi_1\rho(h-1) - \phi_2\rho(h-2) = 0$ for $h \geq 1$. When $h = 1, 2, 3$, we have $\rho(1) = \phi_1/(1-\phi_2)$, $\rho(2) = \phi_1\rho(1) + \phi_2$, $\rho(3) - \phi_1\rho(2) - \phi_2\rho(1) = 0$. Thus,

$$\begin{aligned} \phi_{11} &= \rho(1) = \frac{\phi_1}{1-\phi_2} \\ \phi_{22} &= \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} = \frac{\left[\phi_1\left(\frac{\phi_1}{1-\phi_2}\right) + \phi_2\right] - \left(\frac{\phi_1}{1-\phi_2}\right)^2}{1 - \left(\frac{\phi_1}{1-\phi_2}\right)^2} = \phi_2 \\ \phi_{21} &= \rho(1)[1-\phi_2] = \phi_1 \\ \phi_{33} &= \frac{\rho(3) - \phi_1\rho(2) - \phi_2\rho(1)}{1 - \phi_1\rho(1) - \phi_2\rho(2)} = 0. \end{aligned}$$

Notice that, as shown in (3.72), $\phi_{22} = \phi_2$ for an AR(2) model.

So far, we have concentrated on one-step-ahead prediction, but Property 3.3 allows us to calculate the BLP of x_{n+m} for any $m \geq 1$. Given data, $\{x_1, \ldots, x_n\}$, the m-step-ahead predictor is

$$x_{n+m}^{n} = \phi_{n1}^{(m)} x_n + \phi_{n2}^{(m)} x_{n-1} + \cdots + \phi_{nn}^{(m)} x_1, \tag{3.73}$$

where $\{\phi_{n1}^{(m)}, \phi_{n2}^{(m)}, \ldots, \phi_{nn}^{(m)}\}$ satisfy the prediction equations,

$$\sum_{j=1}^{n} \phi_{nj}^{(m)} E(x_{n+1-j} x_{n+1-k}) = E(x_{n+m} x_{n+1-k}), \quad k = 1, \ldots, n,$$

or

$$\sum_{j=1}^{n} \phi_{nj}^{(m)} \gamma(k-j) = \gamma(m+k-1), \quad k = 1, \ldots, n. \tag{3.74}$$

The prediction equations can again be written in matrix notation as

$$\Gamma_n \boldsymbol{\phi}_n^{(m)} = \boldsymbol{\gamma}_n^{(m)}, \tag{3.75}$$

where $\boldsymbol{\gamma}_n^{(m)} = (\gamma(m), \ldots, \gamma(m+n-1))'$, and $\boldsymbol{\phi}_n^{(m)} = (\phi_{n1}^{(m)}, \ldots, \phi_{nn}^{(m)})'$ are $n \times 1$ vectors.

The mean square m-step-ahead prediction error is

$$P_{n+m}^{n} = E\left(x_{n+m} - x_{n+m}^{n}\right)^2 = \gamma(0) - \boldsymbol{\gamma}_n^{(m)\prime} \Gamma_n^{-1} \boldsymbol{\gamma}_n^{(m)}. \tag{3.76}$$

Another useful algorithm for calculating forecasts was given by Brockwell and Davis (1991, Chapter 5). This algorithm follows directly from applying the projection theorem (Theorem B.1) to the innovations, $x_t - x_t^{t-1}$, for $t = 1, \ldots, n$, using the fact that the innovations $x_t - x_t^{t-1}$ and $x_s - x_s^{s-1}$ are uncorrelated for $s \neq t$ (see Problem 3.41). We present the case in which x_t is a mean-zero stationary time series.

Property 3.6 The Innovations Algorithm

The one-step-ahead predictors, x_{t+1}^t, and their mean-squared errors, P_{t+1}^t, can be calculated iteratively as

$$x_1^0 = 0, \quad P_1^0 = \gamma(0)$$

$$x_{t+1}^t = \sum_{j=1}^{t} \theta_{tj}(x_{t+1-j} - x_{t+1-j}^{t-j}), \quad t = 1, 2, \ldots \tag{3.77}$$

$$P_{t+1}^t = \gamma(0) - \sum_{j=0}^{t-1} \theta_{t,t-j}^2 P_{j+1}^j \quad t = 1, 2, \ldots, \tag{3.78}$$

where, for $j = 0, 1, \ldots, t-1$,

$$\theta_{t,t-j} = \left(\gamma(t-j) - \sum_{k=0}^{j-1} \theta_{j,j-k}\theta_{t,t-k} P_{k+1}^k\right) / \; P_{j+1}^j. \tag{3.79}$$

Given data $x_1, \ldots, x_n$, the innovations algorithm can be calculated successively for $t = 1$, then $t = 2$ and so on, in which case the calculation of x_{n+1}^n and P_{n+1}^n is made at the final step $t = n$. The m-step-ahead predictor and its mean-square error based on the innovations algorithm (Problem 3.41) are given by

$$x_{n+m}^n = \sum_{j=m}^{n+m-1} \theta_{n+m-1,j}(x_{n+m-j} - x_{n+m-j}^{n+m-j-1}), \tag{3.80}$$

$$P_{n+m}^n = \gamma(0) - \sum_{j=m}^{n+m-1} \theta_{n+m-1,j}^2 P_{n+m-j}^{n+m-j-1}, \tag{3.81}$$

where the $\theta_{n+m-1,j}$ are obtained by continued iteration of (3.79).

Example 3.21 Prediction for an MA(1)

The innovations algorithm lends itself well to prediction for moving average processes. Consider an MA(1) model, $x_t = w_t + \theta w_{t-1}$. Recall that $\gamma(0) = (1+\theta^2)\sigma_w^2$, $\gamma(1) = \theta\sigma_w^2$, and $\gamma(h) = 0$ for $h > 1$. Then, using Property 3.6, we have

$$\begin{aligned} \theta_{n1} &= \theta\sigma_w^2/P_n^{n-1} \\ \theta_{nj} &= 0, \quad j = 2, \ldots, n \\ P_1^0 &= (1+\theta^2)\sigma_w^2 \\ P_{n+1}^n &= (1+\theta^2 - \theta\theta_{n1})\sigma_w^2. \end{aligned}$$

Finally, from (3.77), the one-step-ahead predictor is

$$x_{n+1}^n = \theta\left(x_n - x_n^{n-1}\right)\sigma_w^2/P_n^{n-1}.$$

Forecasting ARMA Processes

The general prediction equations (3.60) provide little insight into forecasting for ARMA models in general. There are a number of different ways to express these forecasts, and each aids in understanding the special structure of ARMA prediction. Throughout, we assume x_t is a causal and invertible ARMA(p,q) process, $\phi(B)x_t = \theta(B)w_t$, where $w_t \sim$ iid $\mathrm{N}(0,\sigma_w^2)$. In the non-zero mean case, $E(x_t) = \mu_x$, simply replace x_t with $x_t - \mu_x$ in the model. First, we consider two types of forecasts. We write x_{n+m}^n to mean the minimum mean square error predictor of x_{n+m} based on the data $\{x_n, \ldots, x_1\}$, that is,

$$x_{n+m}^n = E(x_{n+m} \mid x_n, \ldots, x_1).$$

For ARMA models, it is easier to calculate the predictor of x_{n+m}, assuming we have the complete history of the process $\{x_n, x_{n-1}, \ldots, x_1, x_0, x_{-1}, \ldots\}$. We will denote the predictor of x_{n+m} *based on the infinite past* as

$$\widetilde{x}_{n+m} = E(x_{n+m} \mid x_n, x_{n-1}, \ldots, x_1, x_0, x_{-1}, \ldots).$$

In general, x_{n+m}^n and $\widetilde{x}_{n+m}$ are not the same, but the idea here is that, for large samples, $\widetilde{x}_{n+m}$ will provide a good approximation to x_{n+m}^n.

Now, write x_{n+m} in its causal and invertible forms:

$$x_{n+m} = \sum_{j=0}^{\infty} \psi_j w_{n+m-j}, \quad \psi_0 = 1 \tag{3.82}$$

$$w_{n+m} = \sum_{j=0}^{\infty} \pi_j x_{n+m-j}, \quad \pi_0 = 1. \tag{3.83}$$

Then, taking conditional expectations in (3.82), we have

$$\widetilde{x}_{n+m} = \sum_{j=0}^{\infty} \psi_j \widetilde{w}_{n+m-j} = \sum_{j=m}^{\infty} \psi_j w_{n+m-j}, \tag{3.84}$$

because, by causality and invertibility,

$$\widetilde{w}_t = E(w_t \mid x_n, x_{n-1}, \ldots, x_0, x_{-1}, \ldots) = \begin{cases} 0 & t > n \\ w_t & t \leq n. \end{cases}$$

Similarly, taking conditional expectations in (3.83), we have

$$0 = \widetilde{x}_{n+m} + \sum_{j=1}^{\infty} \pi_j \widetilde{x}_{n+m-j},$$

or

$$\widetilde{x}_{n+m} = -\sum_{j=1}^{m-1} \pi_j \widetilde{x}_{n+m-j} - \sum_{j=m}^{\infty} \pi_j x_{n+m-j}, \tag{3.85}$$

using the fact $E(x_t \mid x_n, x_{n-1}, \ldots, x_0, x_{-1}, \ldots) = x_t$, for $t \leq n$. Prediction is accomplished recursively using (3.85), starting with the one-step-ahead predictor, $m = 1$, and then continuing for $m = 2, 3, \ldots$. Using (3.84), we can write

$$x_{n+m} - \widetilde{x}_{n+m} = \sum_{j=0}^{m-1} \psi_j w_{n+m-j},$$

so the mean-square prediction error can be written as

$$P_{n+m}^{n} = E(x_{n+m} - \widetilde{x}_{n+m})^2 = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2. \tag{3.86}$$

Also, we note, for a fixed sample size, n, the prediction errors are correlated. That is, for $k \geq 1$,

$$E\{(x_{n+m} - \widetilde{x}_{n+m})(x_{n+m+k} - \widetilde{x}_{n+m+k})\} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j \psi_{j+k}. \tag{3.87}$$

Example 3.22 Long-Range Forecasts

Consider forecasting an ARMA process with mean μ_x. Replacing x_{n+m} with $x_{n+m} - \mu_x$ in (3.82), and taking conditional expectation as is in (3.84), we deduce that the m-step-ahead forecast can be written as

$$\widetilde{x}_{n+m} = \mu_x + \sum_{j=m}^{\infty} \psi_j w_{n+m-j}. \tag{3.88}$$

Noting that the ψ-weights dampen to zero exponentially fast, it is clear that

$$\widetilde{x}_{n+m} \to \mu_x \tag{3.89}$$

exponentially fast (in the mean square sense) as $m \to \infty$. Moreover, by (3.86), the mean square prediction error

$$P^n_{n+m} \to \sigma_w^2 \sum_{j=0}^{\infty} \psi_j^2 = \gamma_x(0) = \sigma_x^2, \tag{3.90}$$

exponentially fast as $m \to \infty$; recall (3.45).

It should be clear from (3.89) and (3.90) that ARMA forecasts quickly settle to the mean with a constant prediction error as the forecast horizon, m, grows. This effect can be seen in Figure 3.6 on page 119 where the Recruitment series is forecast for 24 months; see Example 3.24.

When n is small, the general prediction equations (3.60) can be used easily. When n is large, we would use (3.85) by truncating, because we do not observe $x_0, x_{-1}, x_{-2}, \ldots$, and only the data $x_1, x_2, \ldots, x_n$ are available. In this case, we can truncate (3.85) by setting $\sum_{j=n+m}^{\infty} \pi_j x_{n+m-j} = 0$. The truncated predictor is then written as

$$\widetilde{x}^n_{n+m} = -\sum_{j=1}^{m-1} \pi_j \widetilde{x}^n_{n+m-j} - \sum_{j=m}^{n+m-1} \pi_j x_{n+m-j}, \tag{3.91}$$

which is also calculated recursively, $m = 1, 2, \ldots$. The mean square prediction error, in this case, is approximated using (3.86).

For AR(p) models, and when $n > p$, equation (3.67) yields the exact predictor, x^n_{n+m}, of x_{n+m}, and there is no need for approximations. That is, for $n > p$, $\widetilde{x}^n_{n+m} = \widetilde{x}_{n+m} = x^n_{n+m}$. Also, in this case, the one-step-ahead prediction error is $E(x_{n+1} - x^n_{n+1})^2 = \sigma_w^2$. For pure MA($q$) or ARMA($p, q$) models, truncated prediction has a fairly nice form.

Property 3.7 Truncated Prediction for ARMA

For ARMA(p, q) models, the truncated predictors for $m = 1, 2, \ldots,$ are

$$\widetilde{x}^n_{n+m} = \phi_1 \widetilde{x}^n_{n+m-1} + \cdots + \phi_p \widetilde{x}^n_{n+m-p} + \theta_1 \widetilde{w}^n_{n+m-1} + \cdots + \theta_q \widetilde{w}^n_{n+m-q}, \tag{3.92}$$

where $\widetilde{x}^n_t = x_t$ for $1 \le t \le n$ and $\widetilde{x}^n_t = 0$ for $t \le 0$. The truncated prediction errors are given by: $\widetilde{w}^n_t = 0$ for $t \le 0$ or $t > n$, and

$$\widetilde{w}^n_t = \phi(B)\widetilde{x}^n_t - \theta_1 \widetilde{w}^n_{t-1} - \cdots - \theta_q \widetilde{w}^n_{t-q}$$

for $1 \le t \le n$.

Example 3.23 Forecasting an ARMA$(1,1)$ Series

Given data $x_1, \ldots, x_n$, for forecasting purposes, write the model as

$$x_{n+1} = \phi x_n + w_{n+1} + \theta w_n.$$

Then, based on (3.92), the one-step-ahead truncated forecast is

$$\widetilde{x}^n_{n+1} = \phi x_n + 0 + \theta \widetilde{w}^n_n.$$

For $m \geq 2$, we have

$$\widetilde{x}^n_{n+m} = \phi \widetilde{x}^n_{n+m-1},$$

which can be calculated recursively, $m = 2, 3, \ldots$.

To calculate $\widetilde{w}^n_n$, which is needed to initialize the successive forecasts, the model can be written as $w_t = x_t - \phi x_{t-1} - \theta w_{t-1}$ for $t = 1, \ldots, n$. For truncated forecasting using (3.92), put $\widetilde{w}^n_0 = 0$, $x_0 = 0$, and then iterate the errors forward in time

$$\widetilde{w}^n_t = x_t - \phi x_{t-1} - \theta \widetilde{w}^n_{t-1}, \quad t = 1, \ldots, n.$$

The approximate forecast variance is computed from (3.86) using the ψ-weights determined as in Example 3.11. In particular, the ψ-weights satisfy $\psi_j = (\phi + \theta)\phi^{j-1}$, for $j \geq 1$. This result gives

$$P^n_{n+m} = \sigma^2_w \left[1 + (\phi + \theta)^2 \sum_{j=1}^{m-1} \phi^{2(j-1)} \right] = \sigma^2_w \left[1 + \frac{(\phi + \theta)^2 (1 - \phi^{2(m-1)})}{(1 - \phi^2)} \right].$$

To assess the precision of the forecasts, prediction intervals are typically calculated along with the forecasts. In general, $(1 - \alpha)$ prediction intervals are of the form

$$x^n_{n+m} \pm c_{\frac{\alpha}{2}} \sqrt{P^n_{n+m}}, \tag{3.93}$$

where $c_{\alpha/2}$ is chosen to get the desired degree of confidence. For example, if the process is Gaussian, then choosing $c_{\alpha/2} = 2$ will yield an approximate 95% prediction interval for x_{n+m}. If we are interested in establishing prediction intervals over more than one time period, then $c_{\alpha/2}$ should be adjusted appropriately, for example, by using Bonferroni's inequality [see (4.55) in Chapter 4 or Johnson and Wichern, 1992, Chapter 5].

Example 3.24 Forecasting the Recruitment Series

Using the parameter estimates as the actual parameter values, Figure 3.6 shows the result of forecasting the Recruitment series given in Example 3.17 over a 24-month horizon, $m = 1, 2, \ldots, 24$. The actual forecasts are calculated as

$$x^n_{n+m} = 6.74 + 1.35 x^n_{n+m-1} - .46 x^n_{n+m-2}$$

for $n = 453$ and $m = 1, 2, \ldots, 12$. Recall that $x^s_t = x_t$ when $t \leq s$. The forecasts errors P^n_{n+m} are calculated using (3.86). Recall that $\widehat{\sigma}^2_w = 89.72$,

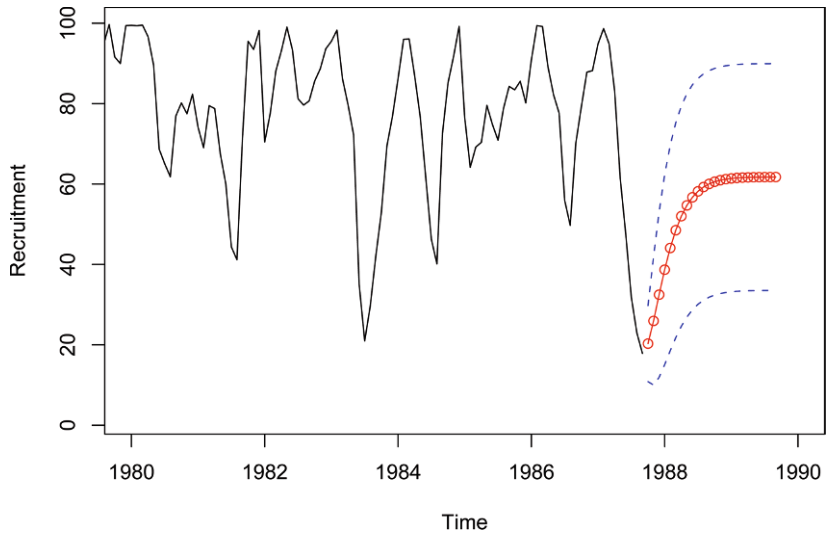

Fig. 3.6. Twenty-four month forecasts for the Recruitment series. The actual data shown are from about January 1980 to September 1987, and then the forecasts plus and minus one standard error are displayed.

and using (3.40) from Example 3.11, we have $\psi_j = 1.35\psi_{j-1} - .46\psi_{j-2}$ for $j \geq 2$, where $\psi_0 = 1$ and $\psi_1 = 1.35$. Thus, for $n = 453$,

$$\begin{aligned} P^n_{n+1} &= 89.72, \\ P^n_{n+2} &= 89.72(1 + 1.35^2), \\ P^n_{n+3} &= 89.72(1 + 1.35^2 + [1.35^2 - .46]^2), \end{aligned}$$

and so on.

Note how the forecast levels off quickly and the prediction intervals are wide, even though in this case the forecast limits are only based on one standard error; that is, $x^n_{n+m} \pm \sqrt{P^n_{n+m}}$.

To reproduce the analysis and Figure 3.6, use the following commands:

```
regr = ar.ols(rec, order=2, demean=FALSE, intercept=TRUE)
fore = predict(regr, n.ahead=24)
ts.plot(rec, fore$pred, col=1:2, xlim=c(1980,1990),
    ylab="Recruitment")
lines(fore$pred, type="p", col=2)
lines(fore$pred+fore$se, lty="dashed", col=4)
lines(fore$pred-fore$se, lty="dashed", col=4)
```

We complete this section with a brief discussion of backcasting. In backcasting, we want to predict x_{1-m}, for $m = 1, 2, \ldots$, based on the data $\{x_1, \ldots, x_n\}$. Write the backcast as

$$x_{1-m}^n = \sum_{j=1}^{n} \alpha_j x_j. \tag{3.94}$$

Analogous to (3.74), the prediction equations (assuming $\mu_x = 0$) are

$$\sum_{j=1}^{n} \alpha_j E(x_j x_k) = E(x_{1-m} x_k), \quad k = 1, \ldots, n, \tag{3.95}$$

or

$$\sum_{j=1}^{n} \alpha_j \gamma(k-j) = \gamma(m+k-1), \quad k = 1, \ldots, n. \tag{3.96}$$

These equations are precisely the prediction equations for forward prediction. That is, $\alpha_j \equiv \phi_{nj}^{(m)}$, for $j = 1, \ldots, n$, where the $\phi_{nj}^{(m)}$ are given by (3.75). Finally, the backcasts are given by

$$x_{1-m}^n = \phi_{n1}^{(m)} x_1 + \cdots + \phi_{nn}^{(m)} x_n, \quad m = 1, 2, \ldots. \tag{3.97}$$

Example 3.25 Backcasting an ARMA$(1,1)$

Consider an ARMA$(1,1)$ process, $x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$; we will call this the *forward model*. We have just seen that best linear prediction backward in time is the same as best linear prediction forward in time for stationary models. Because we are assuming ARMA models are Gaussian, we also have that minimum mean square error prediction backward in time is the same as forward in time for ARMA models.[4] Thus, the process can equivalently be generated by the *backward model*,

$$x_t = \phi x_{t+1} + \theta v_{t+1} + v_t,$$

where $\{v_t\}$ is a Gaussian white noise process with variance σ_w^2. We may write $x_t = \sum_{j=0}^{\infty} \psi_j v_{t+j}$, where $\psi_0 = 1$; this means that x_t is uncorrelated with $\{v_{t-1}, v_{t-2}, \ldots\}$, in analogy to the forward model.

Given data $\{x_1, \ldots., x_n\}$, truncate $v_n^n = E(v_n \,|\, x_1, \ldots., x_n)$ to zero and then iterate backward. That is, put $\widetilde{v}_n^n = 0$, as an initial approximation, and then generate the errors backward

$$\widetilde{v}_t^n = x_t - \phi x_{t+1} - \theta \widetilde{v}_{t+1}^n, \quad t = (n-1), (n-2), \ldots, 1.$$

Then,

$$\widetilde{x}_0^n = \phi x_1 + \theta \widetilde{v}_1^n + \widetilde{v}_0^n = \phi x_1 + \theta \widetilde{v}_1^n,$$

because $\widetilde{v}_t^n = 0$ for $t \leq 0$. Continuing, the general truncated backcasts are given by

$$\widetilde{x}_{1-m}^n = \phi \widetilde{x}_{2-m}^n, \quad m = 2, 3, \ldots \ .$$

[4] In the stationary Gaussian case, (a) the distribution of $\{x_{n+1}, x_n, \ldots, x_1\}$ is the same as (b) the distribution of $\{x_0, x_1 \ldots, x_n\}$. In forecasting we use (a) to obtain $E(x_{n+1} \,|\, x_n, \ldots, x_1)$; in backcasting we use (b) to obtain $E(x_0 \,|\, x_1, \ldots, x_n)$. Because (a) and (b) are the same, the two problems are equivalent.

3.6 Estimation

Throughout this section, we assume we have n observations, $x_1, \ldots, x_n$, from a causal and invertible Gaussian ARMA(p, q) process in which, initially, the order parameters, p and q, are known. Our goal is to estimate the parameters, $\phi_1, \ldots, \phi_p$, $\theta_1, \ldots, \theta_q$, and σ_w^2. We will discuss the problem of determining p and q later in this section.

We begin with method of moments estimators. The idea behind these estimators is that of equating population moments to sample moments and then solving for the parameters in terms of the sample moments. We immediately see that, if $E(x_t) = \mu$, then the method of moments estimator of μ is the sample average, $\bar{x}$. 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.

When the process is AR(p),

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t,$$

the first $p+1$ equations of (3.47) and (3.48) lead to the following:

Definition 3.10 *The* **Yule–Walker equations** *are given by*

$$\gamma(h) = \phi_1\gamma(h-1) + \cdots + \phi_p\gamma(h-p), \quad h = 1, 2, \ldots, p, \tag{3.98}$$

$$\sigma_w^2 = \gamma(0) - \phi_1\gamma(1) - \cdots - \phi_p\gamma(p). \tag{3.99}$$

In matrix notation, the Yule–Walker equations are

$$\Gamma_p\boldsymbol{\phi} = \boldsymbol{\gamma}_p, \quad \sigma_w^2 = \gamma(0) - \boldsymbol{\phi}'\boldsymbol{\gamma}_p, \tag{3.100}$$

where $\Gamma_p = \{\gamma(k-j)\}_{j,k=1}^p$ is a $p \times p$ matrix, $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)'$ is a $p \times 1$ vector, and $\boldsymbol{\gamma}_p = (\gamma(1), \ldots, \gamma(p))'$ is a $p \times 1$ vector. Using the method of moments, we replace $\gamma(h)$ in (3.100) by $\widehat{\gamma}(h)$ [see equation (1.34)] and solve

$$\widehat{\boldsymbol{\phi}} = \widehat{\Gamma}_p^{-1}\widehat{\boldsymbol{\gamma}}_p, \quad \widehat{\sigma}_w^2 = \widehat{\gamma}(0) - \widehat{\boldsymbol{\gamma}}_p'\widehat{\Gamma}_p^{-1}\widehat{\boldsymbol{\gamma}}_p. \tag{3.101}$$

These estimators are typically called the Yule–Walker estimators. For calculation purposes, it is sometimes more convenient to work with the sample ACF. By factoring $\widehat{\gamma}(0)$ in (3.101), we can write the Yule–Walker estimates as

$$\widehat{\boldsymbol{\phi}} = \widehat{\boldsymbol{R}}_p^{-1}\widehat{\boldsymbol{\rho}}_p, \quad \widehat{\sigma}_w^2 = \widehat{\gamma}(0)\left[1 - \widehat{\boldsymbol{\rho}}_p'\widehat{\boldsymbol{R}}_p^{-1}\widehat{\boldsymbol{\rho}}_p\right], \tag{3.102}$$

where $\widehat{R}_p = \{\widehat{\rho}(k-j)\}_{j,k=1}^p$ is a $p \times p$ matrix and $\widehat{\boldsymbol{\rho}}_p = (\widehat{\rho}(1), \ldots, \widehat{\rho}(p))'$ is a $p \times 1$ vector.

For AR(p) models, if the sample size is large, the Yule–Walker estimators are approximately normally distributed, and $\widehat{\sigma}_w^2$ is close to the true value of σ_w^2. We state these results in Property 3.8; for details, see Appendix B, §B.3.

Property 3.8 Large Sample Results for Yule–Walker Estimators

The asymptotic ($n \to \infty$) behavior of the Yule–Walker estimators in the case of causal AR(p) processes is as follows:

$$\sqrt{n}\left(\widehat{\boldsymbol{\phi}} - \boldsymbol{\phi}\right) \xrightarrow{d} N\left(\mathbf{0}, \sigma_w^2 \Gamma_p^{-1}\right), \qquad \widehat{\sigma}_w^2 \xrightarrow{p} \sigma_w^2. \tag{3.103}$$

The Durbin–Levinson algorithm, (3.68)-(3.70), can be used to calculate $\widehat{\boldsymbol{\phi}}$ without inverting $\widehat{\Gamma}_p$ or $\widehat{R}_p$, by replacing $\gamma(h)$ by $\widehat{\gamma}(h)$ in the algorithm. In running the algorithm, we will iteratively calculate the $h \times 1$ vector, $\widehat{\boldsymbol{\phi}}_h = (\widehat{\phi}_{h1}, \ldots, \widehat{\phi}_{hh})'$, for $h = 1, 2, \ldots$. Thus, in addition to obtaining the desired forecasts, the Durbin–Levinson algorithm yields $\widehat{\phi}_{hh}$, the sample PACF. Using (3.103), we can show the following property.

Property 3.9 Large Sample Distribution of the PACF

For a causal AR(p) process, asymptotically ($n \to \infty$),

$$\sqrt{n}\,\widehat{\phi}_{hh} \xrightarrow{d} N(0,1), \quad \text{for} \quad h > p. \tag{3.104}$$

Example 3.26 Yule–Walker Estimation for an AR(2) Process

The data shown in Figure 3.3 were $n = 144$ simulated observations from the AR(2) model

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

where $w_t \sim$ iid N$(0, 1)$. For these data, $\widehat{\gamma}(0) = 8.903$, $\widehat{\rho}(1) = .849$, and $\widehat{\rho}(2) = .519$. Thus,

$$\widehat{\boldsymbol{\phi}} = \begin{pmatrix} \widehat{\phi}_1 \\ \widehat{\phi}_2 \end{pmatrix} = \begin{bmatrix} 1 & .849 \\ .849 & 1 \end{bmatrix}^{-1} \begin{pmatrix} .849 \\ .519 \end{pmatrix} = \begin{pmatrix} 1.463 \\ -.723 \end{pmatrix}$$

and

$$\widehat{\sigma}_w^2 = 8.903 \left[1 - (.849, .519) \begin{pmatrix} 1.463 \\ -.723 \end{pmatrix}\right] = 1.187.$$

By Property 3.8, the asymptotic variance–covariance matrix of $\widehat{\boldsymbol{\phi}}$,

$$\frac{1}{144}\frac{1.187}{8.903} \begin{bmatrix} 1 & .849 \\ .849 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} .058^2 & -.003 \\ -.003 & .058^2 \end{bmatrix},$$

can be used to get confidence regions for, or make inferences about $\widehat{\boldsymbol{\phi}}$ and its components. For example, an approximate 95% confidence interval for ϕ_2 is $-.723 \pm 2(.058)$, or $(-.838, -.608)$, which contains the true value of $\phi_2 = -.75$.

For these data, the first three sample partial autocorrelations are $\widehat{\phi}_{11} = \widehat{\rho}(1) = .849$, $\widehat{\phi}_{22} = \widehat{\phi}_2 = -.721$, and $\widehat{\phi}_{33} = -.085$. According to Property 3.9, the asymptotic standard error of $\widehat{\phi}_{33}$ is $1/\sqrt{144} = .083$, and the observed value, $-.085$, is about only one standard deviation from $\phi_{33} = 0$.

Example 3.27 Yule–Walker Estimation of the Recruitment Series
In Example 3.17 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 in R, which are nearly identical to the values in Example 3.17.

```
rec.yw = ar.yw(rec, order=2)
rec.yw$x.mean  # = 62.26 (mean estimate)
rec.yw$ar      # =  1.33, -.44  (parameter estimates)
sqrt(diag(rec.yw$asy.var.coef))  # = .04, .04  (standard errors)
rec.yw$var.pred  # = 94.80 (error variance estimate)
```

To obtain the 24 month ahead predictions and their standard errors, and then plot the results as in Example 3.24, use the R commands:

```
rec.pr = predict(rec.yw, n.ahead=24)
U = rec.pr$pred + rec.pr$se
L = rec.pr$pred - rec.pr$se
minx = min(rec,L); maxx = max(rec,U)
ts.plot(rec, rec.pr$pred, xlim=c(1980,1990), ylim=c(minx,maxx))
lines(rec.pr$pred, col="red", type="o")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
```

In the case of AR(p) models, the Yule–Walker estimators given in (3.102) are optimal in the sense that the asymptotic distribution, (3.103), is the best asymptotic normal distribution. This is because, given initial conditions, AR(p) models are linear models, and the Yule–Walker estimators are essentially least squares estimators. If we use method of moments for MA or ARMA models, we will not get optimal estimators because such processes are nonlinear in the parameters.

Example 3.28 Method of Moments Estimation for an MA(1)
Consider the time series

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

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:

$$\widehat{\rho}(1) = \frac{\widehat{\gamma}(1)}{\widehat{\gamma}(0)} = \frac{\widehat{\theta}}{1+\widehat{\theta}^2}.$$

Two solutions exist, so we would pick the invertible one. If $|\widehat{\rho}(1)| \leq \frac{1}{2}$, 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 $|\widehat{\rho}(1)| \geq \frac{1}{2}$ because it is an estimator. For example, the following simulation in R produces a value of $\widehat{\rho}(1) = .507$ 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)
acf(ma1, plot=FALSE)[1]  # = .507 (lag 1 sample ACF)
```

When $|\widehat{\rho}(1)| < \frac{1}{2}$, the invertible estimate is

$$\widehat{\theta} = \frac{1 - \sqrt{1 - 4\widehat{\rho}(1)^2}}{2\widehat{\rho}(1)}.$$

It can be shown that[5]

$$\widehat{\theta} \sim \text{AN}\left(\theta,\ \frac{1 + \theta^2 + 4\theta^4 + \theta^6 + \theta^8}{n(1-\theta^2)^2}\right);$$

AN is read *asymptotically normal* and is defined in Definition A.5, page 515, of Appendix A. The maximum likelihood estimator (which we discuss next) of θ, in this case, has an asymptotic variance of $(1-\theta^2)/n$. When $\theta = .5$, for example, the ratio of the asymptotic variance of the method of moments estimator to the maximum likelihood estimator of θ is about 3.5. That is, for large samples, the variance of the method of moments estimator is about 3.5 times larger than the variance of the MLE of θ when $\theta = .5$.

Maximum Likelihood and Least Squares Estimation

To fix ideas, we first focus on the causal AR(1) case. Let

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t \tag{3.105}$$

where $|\phi| < 1$ and $w_t \sim$ iid $\text{N}(0, \sigma_w^2)$. Given data $x_1, x_2, \ldots, x_n$, we seek the likelihood

$$L(\mu, \phi, \sigma_w^2) = f\left(x_1, x_2, \ldots, x_n \mid \mu, \phi, \sigma_w^2\right).$$

In the case of an AR(1), we may write the likelihood as

$$L(\mu, \phi, \sigma_w^2) = f(x_1) f(x_2 \mid x_1) \cdots f(x_n \mid x_{n-1}),$$

where we have dropped the parameters in the densities, $f(\cdot)$, to ease the notation. Because $x_t \mid x_{t-1} \sim \text{N}\left(\mu + \phi(x_{t-1} - \mu),\ \sigma_w^2\right)$, we have

$$f(x_t \mid x_{t-1}) = f_w[(x_t - \mu) - \phi(x_{t-1} - \mu)],$$

where $f_w(\cdot)$ is the density of w_t, that is, the normal density with mean zero and variance σ_w^2. We may then write the likelihood as

$$L(\mu, \phi, \sigma_w) = f(x_1) \prod_{t=2}^{n} f_w\left[(x_t - \mu) - \phi(x_{t-1} - \mu)\right].$$

[5] The result follows from Theorem A.7 given in Appendix A and the delta method. See the proof of Theorem A.7 for details on the delta method.

To find $f(x_1)$, we can use the causal representation

$$x_1 = \mu + \sum_{j=0}^{\infty} \phi^j w_{1-j}$$

to see that x_1 is normal, with mean μ and variance $\sigma_w^2/(1-\phi^2)$. Finally, for an AR(1), the likelihood is

$$L(\mu, \phi, \sigma_w^2) = (2\pi\sigma_w^2)^{-n/2}(1-\phi^2)^{1/2} \exp\left[-\frac{S(\mu,\phi)}{2\sigma_w^2}\right], \tag{3.106}$$

where

$$S(\mu,\phi) = (1-\phi^2)(x_1-\mu)^2 + \sum_{t=2}^{n} \left[(x_t-\mu) - \phi(x_{t-1}-\mu)\right]^2. \tag{3.107}$$

Typically, $S(\mu,\phi)$ is called the unconditional sum of squares. We could have also considered the estimation of μ and ϕ using unconditional least squares, that is, estimation by minimizing $S(\mu,\phi)$.

Taking the partial derivative of the log of (3.106) with respect to σ_w^2 and setting the result equal to zero, we see that for any given values of μ and ϕ in the parameter space, $\sigma_w^2 = n^{-1}S(\mu,\phi)$ maximizes the likelihood. Thus, the maximum likelihood estimate of σ_w^2 is

$$\widehat{\sigma}_w^2 = n^{-1}S(\widehat{\mu},\widehat{\phi}), \tag{3.108}$$

where $\widehat{\mu}$ and $\widehat{\phi}$ are the MLEs of μ and ϕ, respectively. If we replace n in (3.108) by $n-2$, we would obtain the unconditional least squares estimate of σ_w^2.

If, in (3.106), we take logs, replace σ_w^2 by $\widehat{\sigma}_w^2$, and ignore constants, $\widehat{\mu}$ and $\widehat{\phi}$ are the values that minimize the criterion function

$$l(\mu,\phi) = \log\left[n^{-1}S(\mu,\phi)\right] - n^{-1}\log(1-\phi^2); \tag{3.109}$$

that is, $l(\mu,\phi) \propto -2\log L(\mu,\phi,\widehat{\sigma}_w^2)$.[6] Because (3.107) and (3.109) are complicated functions of the parameters, the minimization of $l(\mu,\phi)$ or $S(\mu,\phi)$ 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

$$\begin{aligned} L(\mu,\phi,\sigma_w^2 \mid x_1) &= \prod_{t=2}^{n} f_w\left[(x_t-\mu) - \phi(x_{t-1}-\mu)\right] \\ &= (2\pi\sigma_w^2)^{-(n-1)/2} \exp\left[-\frac{S_c(\mu,\phi)}{2\sigma_w^2}\right], \end{aligned} \tag{3.110}$$

[6] The criterion function is sometimes called the profile or concentrated likelihood.

where the conditional sum of squares is

$$S_c(\mu, \phi) = \sum_{t=2}^{n} [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2 . \tag{3.111}$$

The conditional MLE of σ_w^2 is

$$\widehat{\sigma}_w^2 = S_c(\widehat{\mu}, \widehat{\phi})/(n-1), \tag{3.112}$$

and $\widehat{\mu}$ and $\widehat{\phi}$ are the values that minimize the conditional sum of squares, $S_c(\mu, \phi)$. Letting $\alpha = \mu(1-\phi)$, the conditional sum of squares can be written as

$$S_c(\mu, \phi) = \sum_{t=2}^{n} [x_t - (\alpha + \phi x_{t-1})]^2 . \tag{3.113}$$

The problem is now the linear regression problem stated in §2.2. Following the results from least squares estimation, we have $\widehat{\alpha} = \bar{x}_{(2)} - \widehat{\phi}\bar{x}_{(1)}$, where $\bar{x}_{(1)} = (n-1)^{-1}\sum_{t=1}^{n-1} x_t$, and $\bar{x}_{(2)} = (n-1)^{-1}\sum_{t=2}^{n} x_t$, and the conditional estimates are then

$$\widehat{\mu} = \frac{\bar{x}_{(2)} - \widehat{\phi}\bar{x}_{(1)}}{1 - \widehat{\phi}} \tag{3.114}$$

$$\widehat{\phi} = \frac{\sum_{t=2}^{n}(x_t - \bar{x}_{(2)})(x_{t-1} - \bar{x}_{(1)})}{\sum_{t=2}^{n}(x_{t-1} - \bar{x}_{(1)})^2}. \tag{3.115}$$

From (3.114) and (3.115), we see that $\widehat{\mu} \approx \bar{x}$ and $\widehat{\phi} \approx \widehat{\rho}(1)$. That is, the Yule–Walker estimators and the conditional least squares estimators are approximately the same. The only difference is the inclusion or exclusion of terms involving the endpoints, x_1 and x_n. We can also adjust the estimate of σ_w^2 in (3.112) to be equivalent to the least squares estimator, that is, divide $S_c(\widehat{\mu}, \widehat{\phi})$ by $(n-3)$ instead of $(n-1)$ in (3.112).

For general AR(p) models, maximum likelihood estimation, unconditional least squares, and conditional least squares follow analogously to the AR(1) example. For general ARMA models, it is difficult to write the likelihood as an explicit function of the parameters. Instead, it is advantageous to write the likelihood in terms of the innovations, or one-step-ahead prediction errors, $x_t - x_t^{t-1}$. This will also be useful in Chapter 6 when we study state-space models.

For a normal ARMA(p, q) model, let $\boldsymbol{\beta} = (\mu, \phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)'$ be the $(p+q+1)$-dimensional vector of the model parameters. The likelihood can be written as

$$L(\boldsymbol{\beta}, \sigma_w^2) = \prod_{t=1}^{n} f(x_t \mid x_{t-1}, \ldots, x_1).$$

The conditional distribution of x_t given $x_{t-1}, \ldots, x_1$ is Gaussian with mean x_t^{t-1} and variance P_t^{t-1}. Recall from (3.71) that $P_t^{t-1} = \gamma(0)\prod_{j=1}^{t-1}(1 - \phi_{jj}^2)$. For ARMA models, $\gamma(0) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_j^2$, in which case we may write

$$P_t^{t-1} = \sigma_w^2 \left\{ \left[\sum_{j=0}^{\infty} \psi_j^2 \right] \left[\prod_{j=1}^{t-1} (1 - \phi_{jj}^2) \right] \right\} \stackrel{\text{def}}{=} \sigma_w^2 \, r_t,$$

where r_t is the term in the braces. Note that the r_t terms are functions only of the regression parameters and that they may be computed recursively as $r_{t+1} = (1 - \phi_{tt}^2) r_t$ with initial condition $r_1 = \sum_{j=0}^{\infty} \psi_j^2$. The likelihood of the data can now be written as

$$L(\boldsymbol{\beta}, \sigma_w^2) = (2\pi\sigma_w^2)^{-n/2} \left[r_1(\boldsymbol{\beta}) r_2(\boldsymbol{\beta}) \cdots r_n(\boldsymbol{\beta}) \right]^{-1/2} \exp\left[-\frac{S(\boldsymbol{\beta})}{2\sigma_w^2} \right], \quad (3.116)$$

where

$$S(\boldsymbol{\beta}) = \sum_{t=1}^{n} \left[\frac{(x_t - x_t^{t-1}(\boldsymbol{\beta}))^2}{r_t(\boldsymbol{\beta})} \right]. \quad (3.117)$$

Both x_t^{t-1} and r_t are functions of $\boldsymbol{\beta}$ alone, and we make that fact explicit in (3.116)-(3.117). Given values for $\boldsymbol{\beta}$ and σ_w^2, the likelihood may be evaluated using the techniques of §3.5. Maximum likelihood estimation would now proceed by maximizing (3.116) with respect to $\boldsymbol{\beta}$ and σ_w^2. As in the AR(1) example, we have

$$\widehat{\sigma}_w^2 = n^{-1} S(\widehat{\boldsymbol{\beta}}), \quad (3.118)$$

where $\widehat{\boldsymbol{\beta}}$ is the value of $\boldsymbol{\beta}$ that minimizes the concentrated likelihood

$$l(\boldsymbol{\beta}) = \log\left[n^{-1} S(\boldsymbol{\beta}) \right] + n^{-1} \sum_{t=1}^{n} \log r_t(\boldsymbol{\beta}). \quad (3.119)$$

For the AR(1) model (3.105) discussed previously, recall that $x_1^0 = \mu$ and $x_t^{t-1} = \mu + \phi(x_{t-1} - \mu)$, for $t = 2, \ldots, n$. Also, using the fact that $\phi_{11} = \phi$ and $\phi_{hh} = 0$ for $h > 1$, we have $r_1 = \sum_{j=0}^{\infty} \phi^{2j} = (1 - \phi^2)^{-1}$, $r_2 = (1 - \phi^2)^{-1}(1 - \phi^2) = 1$, and in general, $r_t = 1$ for $t = 2, \ldots, n$. Hence, the likelihood presented in (3.106) is identical to the innovations form of the likelihood given by (3.116). Moreover, the generic $S(\boldsymbol{\beta})$ in (3.117) is $S(\mu, \phi)$ given in (3.107) and the generic $l(\boldsymbol{\beta})$ in (3.119) is $l(\mu, \phi)$ in (3.109).

Unconditional least squares would be performed by minimizing (3.117) with respect to $\boldsymbol{\beta}$. Conditional least squares estimation would involve minimizing (3.117) with respect to $\boldsymbol{\beta}$ but where, to ease the computational burden, the predictions and their errors are obtained by conditioning on initial values of the data. In general, numerical optimization routines are used to obtain the actual estimates and their standard errors.

Example 3.29 The Newton–Raphson and Scoring Algorithms

Two common numerical optimization routines for accomplishing maximum likelihood estimation are Newton–Raphson and scoring. We will give a brief account of the mathematical ideas here. The actual implementation of these algorithms is much more complicated than our discussion might imply. For

details, the reader is referred to any of the *Numerical Recipes* books, for example, Press et al. (1993).

Let $l(\boldsymbol{\beta})$ be a criterion function of k parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ that we wish to minimize with respect to $\boldsymbol{\beta}$. For example, consider the likelihood function given by (3.109) or by (3.119). Suppose $l(\widehat{\boldsymbol{\beta}})$ is the extremum that we are interested in finding, and $\widehat{\boldsymbol{\beta}}$ is found by solving $\partial l(\boldsymbol{\beta})/\partial\beta_j = 0$, for $j = 1, \ldots, k$. Let $l^{(1)}(\boldsymbol{\beta})$ denote the $k \times 1$ vector of partials

$$l^{(1)}(\boldsymbol{\beta}) = \left(\frac{\partial l(\boldsymbol{\beta})}{\partial\beta_1}, \ldots, \frac{\partial l(\boldsymbol{\beta})}{\partial\beta_k}\right)'.$$

Note, $l^{(1)}(\widehat{\boldsymbol{\beta}}) = \mathbf{0}$, the $k \times 1$ zero vector. Let $l^{(2)}(\boldsymbol{\beta})$ denote the $k \times k$ matrix of second-order partials

$$l^{(2)}(\boldsymbol{\beta}) = \left\{-\frac{\partial l^2(\boldsymbol{\beta})}{\partial\beta_i\partial\beta_j}\right\}_{i,j=1}^{k},$$

and assume $l^{(2)}(\boldsymbol{\beta})$ is nonsingular. Let $\boldsymbol{\beta}_{(0)}$ be an initial estimator of $\boldsymbol{\beta}$. Then, using a Taylor expansion, we have the following approximation:

$$\mathbf{0} = l^{(1)}(\widehat{\boldsymbol{\beta}}) \approx l^{(1)}(\boldsymbol{\beta}_{(0)}) - l^{(2)}(\boldsymbol{\beta}_{(0)})\left[\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_{(0)}\right].$$

Setting the right-hand side equal to zero and solving for $\widehat{\boldsymbol{\beta}}$ [call the solution $\boldsymbol{\beta}_{(1)}$], we get

$$\boldsymbol{\beta}_{(1)} = \boldsymbol{\beta}_{(0)} + \left[l^{(2)}(\boldsymbol{\beta}_{(0)})\right]^{-1} l^{(1)}(\boldsymbol{\beta}_{(0)}).$$

The Newton–Raphson algorithm proceeds by iterating this result, replacing $\boldsymbol{\beta}_{(0)}$ by $\boldsymbol{\beta}_{(1)}$ to get $\boldsymbol{\beta}_{(2)}$, and so on, until convergence. Under a set of appropriate conditions, the sequence of estimators, $\boldsymbol{\beta}_{(1)}, \boldsymbol{\beta}_{(2)}, \ldots$, will converge to $\widehat{\boldsymbol{\beta}}$, the MLE of $\boldsymbol{\beta}$.

For maximum likelihood estimation, the criterion function used is $l(\boldsymbol{\beta})$ given by (3.119); $l^{(1)}(\boldsymbol{\beta})$ is called the score vector, and $l^{(2)}(\boldsymbol{\beta})$ is called the Hessian. In the method of scoring, we replace $l^{(2)}(\boldsymbol{\beta})$ by $E[l^{(2)}(\boldsymbol{\beta})]$, the information matrix. Under appropriate conditions, the inverse of the information matrix is the asymptotic variance–covariance matrix of the estimator $\widehat{\boldsymbol{\beta}}$. This is sometimes approximated by the inverse of the Hessian at $\widehat{\boldsymbol{\beta}}$. If the derivatives are difficult to obtain, it is possible to use quasi-maximum likelihood estimation where numerical techniques are used to approximate the derivatives.

Example 3.30 MLE for the Recruitment Series

So far, we have fit an AR(2) model to the Recruitment series using ordinary least squares (Example 3.17) and using Yule–Walker (Example 3.27). The following is an R session used to fit an AR(2) model via maximum likelihood estimation to the Recruitment series; these results can be compared to the results in Examples 3.17 and 3.27.

```
rec.mle = ar.mle(rec, order=2)
rec.mle$x.mean   # 62.26
rec.mle$ar       # 1.35, -.46
sqrt(diag(rec.mle$asy.var.coef))  # .04, .04
rec.mle$var.pred     # 89.34
```

We now discuss least squares for ARMA(p, q) models via Gauss–Newton. For general and complete details of the Gauss–Newton procedure, the reader is referred to Fuller (1996). As before, write $\boldsymbol{\beta} = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)'$, and for the ease of discussion, we will put $\mu = 0$. We write the model in terms of the errors

$$w_t(\boldsymbol{\beta}) = x_t - \sum_{j=1}^{p} \phi_j x_{t-j} - \sum_{k=1}^{q} \theta_k w_{t-k}(\boldsymbol{\beta}), \tag{3.120}$$

emphasizing the dependence of the errors on the parameters.

For conditional least squares, we approximate the residual sum of squares by conditioning on $x_1, \ldots, x_p$ (if $p > 0$) and $w_p = w_{p-1} = w_{p-2} = \cdots = w_{1-q} = 0$ (if $q > 0$), in which case, given $\boldsymbol{\beta}$, we may evaluate (3.120) for $t = p+1, p+2, \ldots, n$. Using this conditioning argument, the conditional error sum of squares is

$$S_c(\boldsymbol{\beta}) = \sum_{t=p+1}^{n} w_t^2(\boldsymbol{\beta}). \tag{3.121}$$

Minimizing $S_c(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ yields the conditional least squares estimates. If $q = 0$, the problem is linear regression and no iterative technique is needed to minimize $S_c(\phi_1, \ldots, \phi_p)$. If $q > 0$, the problem becomes nonlinear regression and we will have to rely on numerical optimization.

When n is large, conditioning on a few initial values will have little influence on the final parameter estimates. In the case of small to moderate sample sizes, one may wish to rely on unconditional least squares. The unconditional least squares problem is to choose $\boldsymbol{\beta}$ to minimize the unconditional sum of squares, which we have generically denoted by $S(\boldsymbol{\beta})$ in this section. The unconditional sum of squares can be written in various ways, and one useful form in the case of ARMA(p, q) models is derived in Box et al. (1994, Appendix A7.3). They showed (see Problem 3.19) the unconditional sum of squares can be written as

$$S(\boldsymbol{\beta}) = \sum_{t=-\infty}^{n} \widehat{w}_t^2(\boldsymbol{\beta}), \tag{3.122}$$

where $\widehat{w}_t(\boldsymbol{\beta}) = E(w_t \,|\, x_1, \ldots, x_n)$. When $t \leq 0$, the $\widehat{w}_t(\boldsymbol{\beta})$ are obtained by backcasting. As a practical matter, we approximate $S(\boldsymbol{\beta})$ by starting the sum at $t = -M + 1$, where M is chosen large enough to guarantee $\sum_{t=-\infty}^{-M} \widehat{w}_t^2(\boldsymbol{\beta}) \approx 0$. In the case of unconditional least squares estimation, a numerical optimization technique is needed even when $q = 0$.

To employ Gauss–Newton, let $\boldsymbol{\beta}_{(0)} = (\phi_1^{(0)}, \ldots, \phi_p^{(0)}, \theta_1^{(0)}, \ldots, \theta_q^{(0)})'$ be an initial estimate of $\boldsymbol{\beta}$. For example, we could obtain $\boldsymbol{\beta}_{(0)}$ by method of moments. The first-order Taylor expansion of $w_t(\boldsymbol{\beta})$ is

$$w_t(\boldsymbol{\beta}) \approx w_t(\boldsymbol{\beta}_{(0)}) - \Big(\boldsymbol{\beta} - \boldsymbol{\beta}_{(0)}\Big)' \boldsymbol{z}_t(\boldsymbol{\beta}_{(0)}), \tag{3.123}$$

where

$$\boldsymbol{z}_t(\boldsymbol{\beta}_{(0)}) = \left(-\frac{\partial w_t(\boldsymbol{\beta}_{(0)})}{\partial \beta_1}, \ldots, -\frac{\partial w_t(\boldsymbol{\beta}_{(0)})}{\partial \beta_{p+q}}\right)', \quad t = 1, \ldots, n.$$

The linear approximation of $S_c(\boldsymbol{\beta})$ is

$$Q(\boldsymbol{\beta}) = \sum_{t=p+1}^{n} \left[w_t(\boldsymbol{\beta}_{(0)}) - \Big(\boldsymbol{\beta} - \boldsymbol{\beta}_{(0)}\Big)' \boldsymbol{z}_t(\boldsymbol{\beta}_{(0)})\right]^2 \tag{3.124}$$

and this is the quantity that we will minimize. For approximate unconditional least squares, we would start the sum in (3.124) at $t = -M + 1$, for a large value of M, and work with the backcasted values.

Using the results of ordinary least squares (§2.2), we know

$$\widehat{(\boldsymbol{\beta} - \boldsymbol{\beta}_{(0)})} = \Big(n^{-1} \sum_{t=p+1}^{n} \boldsymbol{z}_t(\boldsymbol{\beta}_{(0)}) \boldsymbol{z}_t'(\boldsymbol{\beta}_{(0)})\Big)^{-1} \Big(n^{-1} \sum_{t=p+1}^{n} \boldsymbol{z}_t(\boldsymbol{\beta}_{(0)}) w_t(\boldsymbol{\beta}_{(0)})\Big) \tag{3.125}$$

minimizes $Q(\boldsymbol{\beta})$. From (3.125), we write the one-step Gauss–Newton estimate as

$$\boldsymbol{\beta}_{(1)} = \boldsymbol{\beta}_{(0)} + \Delta(\boldsymbol{\beta}_{(0)}), \tag{3.126}$$

where $\Delta(\boldsymbol{\beta}_{(0)})$ denotes the right-hand side of (3.125). Gauss–Newton estimation is accomplished by replacing $\boldsymbol{\beta}_{(0)}$ by $\boldsymbol{\beta}_{(1)}$ in (3.126). This process is repeated by calculating, at iteration $j = 2, 3, \ldots$,

$$\boldsymbol{\beta}_{(j)} = \boldsymbol{\beta}_{(j-1)} + \Delta(\boldsymbol{\beta}_{(j-1)})$$

until convergence.

Example 3.31 Gauss–Newton for an MA(1)

Consider an invertible MA(1) process, $x_t = w_t + \theta w_{t-1}$. Write the truncated errors as

$$w_t(\theta) = x_t - \theta w_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{3.127}$$

where we condition on $w_0(\theta) = 0$. Taking derivatives,

$$-\frac{\partial w_t(\theta)}{\partial \theta} = w_{t-1}(\theta) + \theta \frac{\partial w_{t-1}(\theta)}{\partial \theta}, \quad t = 1, \ldots, n, \tag{3.128}$$

where $\partial w_0(\theta)/\partial\theta = 0$. Using the notation of (3.123), we can also write (3.128) as

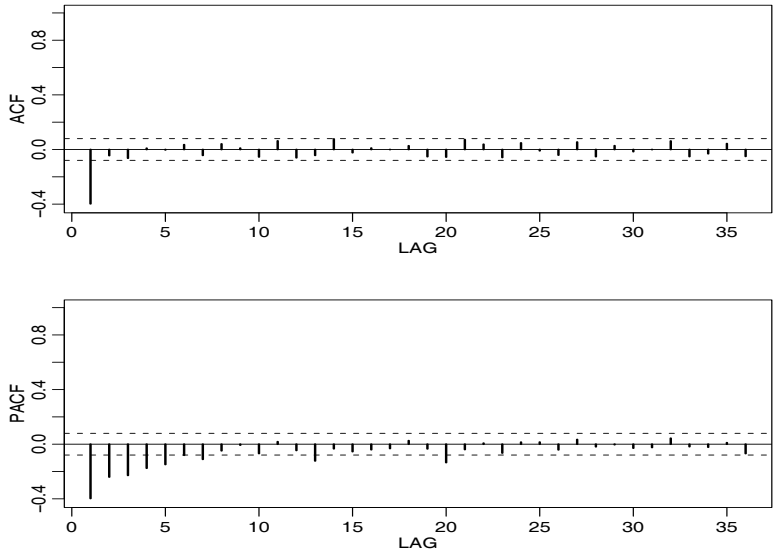

Fig. 3.7. ACF and PACF of transformed glacial varves.

$$z_t(\theta) = w_{t-1}(\theta) - \theta z_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{3.129}$$

where $z_0(\theta) = 0$.

Let $\theta_{(0)}$ be an initial estimate of θ, for example, the estimate given in Example 3.28. Then, the Gauss–Newton procedure for conditional least squares is given by

$$\theta_{(j+1)} = \theta_{(j)} + \frac{\sum_{t=1}^{n} z_t(\theta_{(j)}) w_t(\theta_{(j)})}{\sum_{t=1}^{n} z_t^2(\theta_{(j)})}, \quad j = 0, 1, 2, \ldots, \tag{3.130}$$

where the values in (3.130) are calculated recursively using (3.127) and (3.129). 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 3.32 Fitting the Glacial Varve Series

Consider the series of glacial varve thicknesses from Massachusetts for $n = 634$ years, as analyzed in Example 2.6 and in Problem 2.8, where it was argued that a first-order moving average model might fit the logarithmically transformed and differenced varve series, say,

$$\nabla \log(x_t) = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right),$$

which can be interpreted as being approximately the percentage change in the thickness.

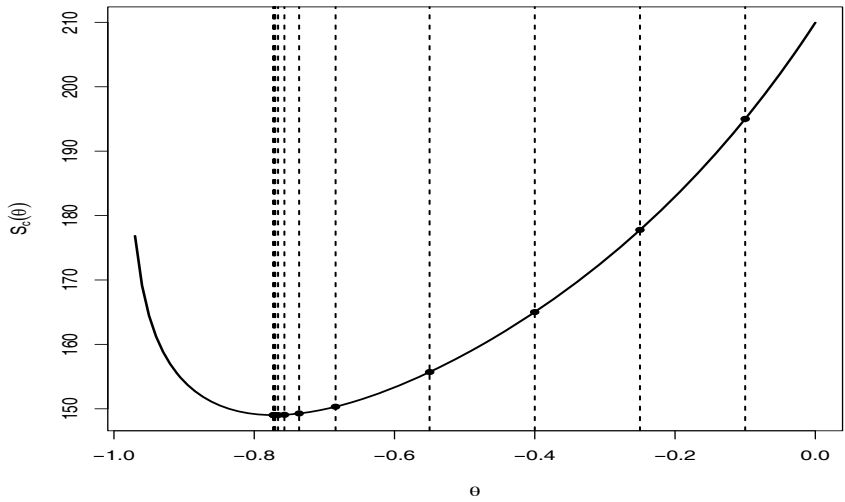

Fig. 3.8. Conditional sum of squares versus values of the moving average parameter for the glacial varve example, Example 3.32. Vertical lines indicate the values of the parameter obtained via Gauss–Newton; see Table 3.2 for the actual values.

The sample ACF and PACF, shown in Figure 3.7, confirm the tendency of $\nabla \log(x_t)$ to behave as a first-order moving average process as the ACF has only a significant peak at lag one and the PACF decreases exponentially. Using Table 3.1, this sample behavior fits that of the MA(1) very well.

The results of eleven iterations of the Gauss–Newton procedure, (3.130), starting with $\theta_{(0)} = -.10$ are given in Table 3.2. The final estimate is $\widehat{\theta} = \theta_{(11)} = -.773$; interim values and the corresponding value of the conditional sum of squares, $S_c(\theta)$ given in (3.121), are also displayed in the table. The final estimate of the error variance is $\widehat{\sigma}_w^2 = 148.98/632 = .236$ with 632 degrees of freedom (one is lost in differencing). The value of the sum of the squared derivatives at convergence is $\sum_{t=1}^{n} z_t^2(\theta_{(11)}) = 369.73$, and consequently, the estimated standard error of $\widehat{\theta}$ is $\sqrt{.236/369.73} = .025$;[7] this leads to a t-value of $-.773/.025 = -30.92$ with 632 degrees of freedom.

Figure 3.8 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. When there is only one parameter, as in this case, it would be easy to evaluate $S_c(\theta)$ on a grid of points, and then choose the appropriate value of θ from the grid search. It would be difficult, however, to perform grid searches when there are many parameters.

[7] To estimate the standard error, we are using the standard regression results from (2.9) as an approximation

Table 3.2. Gauss–Newton Results for Example 3.32

j	$\theta_{(j)}$	$S_c(\theta_{(j)})$	$\sum_{t=1}^{n} z_t^2(\theta_{(j)})$
0	−0.100	195.0010	183.3464
1	−0.250	177.7614	163.3038
2	−0.400	165.0027	161.6279
3	−0.550	155.6723	182.6432
4	−0.684	150.2896	247.4942
5	−0.736	149.2283	304.3125
6	−0.757	149.0272	337.9200
7	−0.766	148.9885	355.0465
8	−0.770	148.9812	363.2813
9	−0.771	148.9804	365.4045
10	−0.772	148.9799	367.5544
11	−0.773	148.9799	369.7314

In the general case of causal and invertible ARMA(p, q) models, maximum likelihood estimation and conditional and unconditional least squares estimation (and Yule–Walker estimation in the case of AR models) all lead to optimal estimators. The proof of this general result can be found in a number of texts on theoretical time series analysis (for example, Brockwell and Davis, 1991, or Hannan, 1970, to mention a few). We will denote the ARMA coefficient parameters by $\boldsymbol{\beta} = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)'$.

Property 3.10 Large Sample Distribution of the Estimators

Under appropriate conditions, for causal and invertible ARMA processes, the maximum likelihood, the unconditional least squares, and the conditional least squares estimators, each initialized by the method of moments estimator, all provide optimal estimators of σ_w^2 and $\boldsymbol{\beta}$, in the sense that $\widehat{\sigma}_w^2$ is consistent, and the asymptotic distribution of $\widehat{\boldsymbol{\beta}}$ is the best asymptotic normal distribution. In particular, as $n \to \infty$,

$$\sqrt{n}\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right) \xrightarrow{d} N\left(\mathbf{0}, \sigma_w^2\, \boldsymbol{\Gamma}_{p,q}^{-1}\right). \tag{3.131}$$

The asymptotic variance–covariance matrix of the estimator $\widehat{\boldsymbol{\beta}}$ is the inverse of the information matrix. In particular, the $(p+q)\times(p+q)$ matrix $\Gamma_{p,q}$, has the form

$$\Gamma_{p,q} = \begin{pmatrix} \Gamma_{\phi\phi} & \Gamma_{\phi\theta} \\ \Gamma_{\theta\phi} & \Gamma_{\theta\theta} \end{pmatrix}. \tag{3.132}$$

The $p \times p$ matrix $\Gamma_{\phi\phi}$ is given by (3.100), that is, the ij-th element of $\Gamma_{\phi\phi}$, for $i, j = 1, \ldots, p$, is $\gamma_x(i-j)$ from an AR(p) process, $\phi(B)x_t = w_t$. Similarly, $\Gamma_{\theta\theta}$ is a $q \times q$ matrix with the ij-th element, for $i, j = 1, \ldots, q$, equal to $\gamma_y(i-j)$ from an AR(q) process, $\theta(B)y_t = w_t$. The $p \times q$ matrix $\Gamma_{\phi\theta} = \{\gamma_{xy}(i-j)\}$, for $i = 1, \ldots, p$; $j = 1, \ldots, q$; that is, the ij-th element is the cross-covariance

between the two AR processes given by $\phi(B)x_t = w_t$ *and* $\theta(B)y_t = w_t$*. Finally,* $\Gamma_{\theta\phi} = \Gamma'_{\phi\theta}$ *is* $q \times p$*.*

Further discussion of Property 3.10, including a proof for the case of least squares estimators for AR(p) processes, can be found in Appendix B, §B.3.

Example 3.33 Some Specific Asymptotic Distributions

The following are some specific cases of Property 3.10.

AR(1): $\gamma_x(0) = \sigma_w^2/(1-\phi^2)$, so $\sigma_w^2\Gamma_{1,0}^{-1} = (1-\phi^2)$. Thus,

$$\widehat{\phi} \sim \text{AN}\left[\phi, n^{-1}(1-\phi^2)\right]. \tag{3.133}$$

AR(2): The reader can verify that

$$\gamma_x(0) = \left(\frac{1-\phi_2}{1+\phi_2}\right)\frac{\sigma_w^2}{(1-\phi_2)^2-\phi_1^2}$$

and $\gamma_x(1) = \phi_1\gamma_x(0) + \phi_2\gamma_x(1)$. From these facts, we can compute $\Gamma_{2,0}^{-1}$. In particular, we have

$$\begin{pmatrix}\widehat{\phi}_1\\ \widehat{\phi}_2\end{pmatrix} \sim \text{AN}\left[\begin{pmatrix}\phi_1\\ \phi_2\end{pmatrix}, n^{-1}\begin{pmatrix}1-\phi_2^2 & -\phi_1(1+\phi_2)\\ \text{sym} & 1-\phi_2^2\end{pmatrix}\right]. \tag{3.134}$$

MA(1): In this case, write $\theta(B)y_t = w_t$, or $y_t + \theta y_{t-1} = w_t$. Then, analogous to the AR(1) case, $\gamma_y(0) = \sigma_w^2/(1-\theta^2)$, so $\sigma_w^2\Gamma_{0,1}^{-1} = (1-\theta^2)$. Thus,

$$\widehat{\theta} \sim \text{AN}\left[\theta, n^{-1}(1-\theta^2)\right]. \tag{3.135}$$

MA(2): Write $y_t + \theta_1 y_{t-1} + \theta_2 y_{t-2} = w_t$, so , analogous to the AR(2) case, we have

$$\begin{pmatrix}\widehat{\theta}_1\\ \widehat{\theta}_2\end{pmatrix} \sim \text{AN}\left[\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}, n^{-1}\begin{pmatrix}1-\theta_2^2 & \theta_1(1+\theta_2)\\ \text{sym} & 1-\theta_2^2\end{pmatrix}\right]. \tag{3.136}$$

ARMA(1,1): To calculate $\Gamma_{\phi\theta}$, we must find $\gamma_{xy}(0)$, where $x_t - \phi x_{t-1} = w_t$ and $y_t + \theta y_{t-1} = w_t$. We have

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

Solving, we find, $\gamma_{xy}(0) = \sigma_w^2/(1+\phi\theta)$. Thus,

$$\begin{pmatrix}\widehat{\phi}\\ \widehat{\theta}\end{pmatrix} \sim \text{AN}\left[\begin{pmatrix}\phi\\ \theta\end{pmatrix}, n^{-1}\begin{bmatrix}(1-\phi^2)^{-1} & (1+\phi\theta)^{-1}\\ \text{sym} & (1-\theta^2)^{-1}\end{bmatrix}^{-1}\right]. \tag{3.137}$$

Example 3.34 Overfitting Caveat

The asymptotic 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, $\mathrm{var}(\widehat{\phi}_1) \approx n^{-1}(1-\phi_1^2)$. But, if we fit an AR(2) to the AR(1) process, for large n, $\mathrm{var}(\widehat{\phi}_1) \approx n^{-1}(1-\phi_2^2) = n^{-1}$ 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(2) model to the data and are satisfied with that model, then adding one more parameter and fitting an AR(3) should lead to approximately the same model as in the AR(2) fit. We will discuss model diagnostics in more detail in §3.8.

The reader might wonder, for example, why the asymptotic distributions of $\widehat{\phi}$ from an AR(1) and $\widehat{\theta}$ from an MA(1) are of the same form; compare (3.133) to (3.135). It is possible to explain this unexpected result heuristically using the intuition of linear regression. That is, for the normal regression model presented in §2.2 with no intercept term, $x_t = \beta z_t + w_t$, we know $\widehat{\beta}$ is normally distributed with mean β, and from (2.9),

$$\mathrm{var}\left\{\sqrt{n}\left(\widehat{\beta}-\beta\right)\right\} = n\sigma_w^2\left(\sum_{t=1}^{n} z_t^2\right)^{-1} = \sigma_w^2\left(n^{-1}\sum_{t=1}^{n} z_t^2\right)^{-1}.$$

For the causal AR(1) model given by $x_t = \phi x_{t-1} + w_t$, the intuition of regression tells us to expect that, for n large,

$$\sqrt{n}\left(\widehat{\phi}-\phi\right)$$

is approximately normal with mean zero and with variance given by

$$\sigma_w^2\left(n^{-1}\sum_{t=2}^{n} x_{t-1}^2\right)^{-1}.$$

Now, $n^{-1}\sum_{t=2}^{n} x_{t-1}^2$ is the sample variance (recall that the mean of x_t is zero) of the x_t, so as n becomes large we would expect it to approach $\mathrm{var}(x_t) = \gamma(0) = \sigma_w^2/(1-\phi^2)$. Thus, the large sample variance of $\sqrt{n}\left(\widehat{\phi}-\phi\right)$ is

$$\sigma_w^2\gamma_x(0)^{-1} = \sigma_w^2\left(\frac{\sigma_w^2}{1-\phi^2}\right)^{-1} = (1-\phi^2);$$

that is, (3.133) holds.

In the case of an MA(1), we may use the discussion of Example 3.31 to write an approximate regression model for the MA(1). That is, consider the approximation (3.129) as the regression model

$$z_t(\widehat{\theta}) = -\theta z_{t-1}(\widehat{\theta}) + w_{t-1},$$

where now, $z_{t-1}(\widehat{\theta})$ as defined in Example 3.31, plays the role of the regressor. Continuing with the analogy, we would expect the asymptotic distribution of $\sqrt{n}\left(\widehat{\theta} - \theta\right)$ to be normal, with mean zero, and approximate variance

$$\sigma_w^2 \left(n^{-1} \sum_{t=2}^{n} z_{t-1}^2(\widehat{\theta}) \right)^{-1}.$$

As in the AR(1) case, $n^{-1}\sum_{t=2}^{n} z_{t-1}^2(\widehat{\theta})$ is the sample variance of the $z_t(\widehat{\theta})$ so, for large n, this should be $\mathrm{var}\{z_t(\theta)\} = \gamma_z(0)$, say. But note, as seen from (3.129), $z_t(\theta)$ is approximately an AR(1) process with parameter $-\theta$. Thus,

$$\sigma_w^2 \gamma_z(0)^{-1} = \sigma_w^2 \left(\frac{\sigma_w^2}{1-(-\theta)^2} \right)^{-1} = (1-\theta^2),$$

which agrees with (3.135). Finally, the asymptotic distributions of the AR parameter estimates and the MA parameter estimates are of the same form because in the MA case, the "regressors" are the differential processes $z_t(\theta)$ that have AR structure, and it is this structure that determines the asymptotic variance of the estimators. For a rigorous account of this approach for the general case, see Fuller (1996, Theorem 5.5.4).

In Example 3.32, the estimated standard error of $\widehat{\theta}$ was .025. In that example, we used regression results to estimate the standard error as the square root of

$$n^{-1}\widehat{\sigma}_w^2 \left(n^{-1} \sum_{t=1}^{n} z_t^2(\widehat{\theta}) \right)^{-1} = \frac{\widehat{\sigma}_w^2}{\sum_{t=1}^{n} z_t^2(\widehat{\theta})},$$

where $n = 632$, $\widehat{\sigma}_w^2 = .236$, $\sum_{t=1}^{n} z_t^2(\widehat{\theta}) = 369.73$ and $\widehat{\theta} = -.773$. Using (3.135), we could have also calculated this value using the asymptotic approximation, the square root of $(1-(-.773)^2)/632$, which is also .025.

If n is small, or if the parameters are close to the boundaries, the asymptotic approximations can be quite poor. The bootstrap can be helpful in this case; for a broad treatment of the bootstrap, see Efron and Tibshirani (1994). We discuss the case of an AR(1) here and leave the general discussion for Chapter 6. For now, we give a simple example of the bootstrap for an AR(1) process.

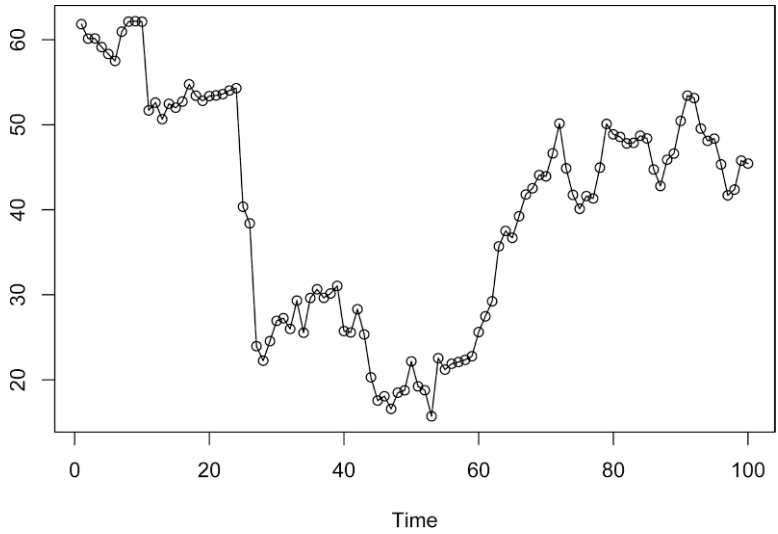

Fig. 3.9. One hundred observations generated from the model in Example 3.35.

Example 3.35 Bootstrapping an AR(1)

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

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t, \tag{3.138}$$

where $\mu = 50$, $\phi = .95$, and w_t are iid 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 3.9 shows $n = 100$ simulated observations from this process. This particular realization is interesting; the data look like they were generated from a nonstationary process with three different mean levels. In fact, the data were generated from a well-behaved, albeit non-normal, stationary and causal model. To show the advantages of the bootstrap, we will act as if we do not know the actual error distribution and we will proceed as if it were normal; of course, this means, for example, that the normal based MLE of ϕ will not be the actual MLE because the data are not normal.

Using the data shown in Figure 3.9, we obtained the Yule–Walker estimates $\widehat{\mu} = 40.05$, $\widehat{\phi} = .96$, and $s_w^2 = 15.30$, where s_w^2 is the estimate of $\text{var}(w_t)$. Based on Property 3.10, we would say that $\widehat{\phi}$ is approximately normal with mean ϕ (which we supposedly do not know) and variance $(1 - \phi^2)/100$, which we would approximate by $(1 - .96^2)/100 = .03^2$.

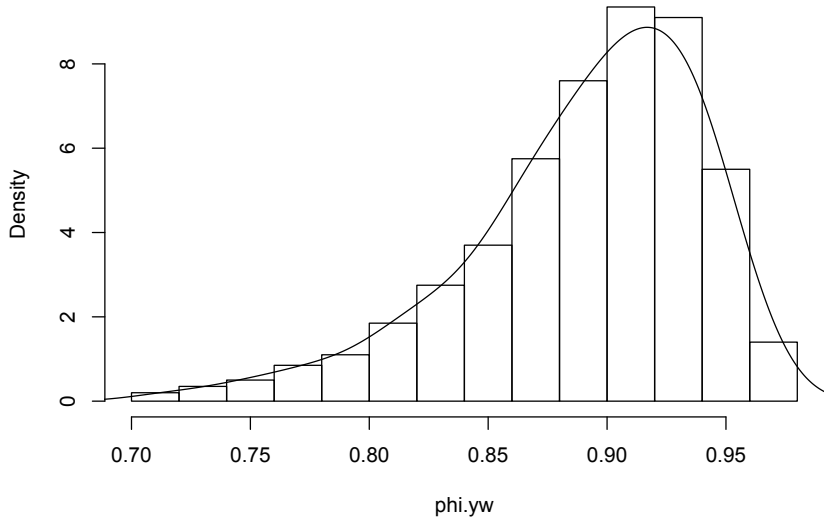

Fig. 3.10. Finite sample density of the Yule–Walker estimate of ϕ in Example 3.35.

To assess the finite sample distribution of $\widehat{\phi}$ 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 3.10. Clearly the sampling distribution is not close to normality for this sample size. The mean of the distribution shown in Figure 3.10 is .89, and the variance of the distribution is $.05^2$; these values are considerably different than the asymptotic values. Some of the quantiles of the finite sample distribution are .79 (5%), .86 (25%), .90 (50%), .93 (75%), and .95 (95%). The R code to perform the simulation and plot the histogram is as follows:

```
set.seed(111)
phi.yw = rep(NA, 1000)
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 }
hist(phi.yw, prob=TRUE, main="")
lines(density(phi.yw, bw=.015))
```

Before discussing the bootstrap, we first investigate the sample innovation process, $x_t - x_t^{t-1}$, with corresponding variances P_t^{t-1}. For the AR(1) model in this example,

$$x_t^{t-1} = \mu + \phi(x_{t-1} - \mu), \qquad t = 2, \ldots, 100.$$

From this, it follows that

$$P_t^{t-1} = E(x_t - x_t^{t-1})^2 = \sigma_w^2, \qquad t = 2, \ldots, 100.$$

When $t = 1$, we have

$$x_1^0 = \mu \quad \text{and} \quad P_1^0 = \sigma_w^2/(1-\phi^2).$$

Thus, the innovations have zero mean but different variances; in order that all of the innovations have the same variance, σ_w^2, we will write them as

$$\begin{aligned} \epsilon_1 &= (x_1 - \mu)\sqrt{(1-\phi^2)} \\ \epsilon_t &= (x_t - \mu) - \phi(x_{t-1} - \mu), \quad \text{for} \quad t = 2, \ldots, 100. \end{aligned} \tag{3.139}$$

From these equations, we can write the model in terms of the ϵ_t as

$$\begin{aligned} x_1 &= \mu + \epsilon_1/\sqrt{(1-\phi^2)} \\ x_t &= \mu + \phi(x_{t-1} - \mu) + \epsilon_t \quad \text{for} \quad t = 2, \ldots, 100. \end{aligned} \tag{3.140}$$

Next, replace the parameters with their estimates in (3.139), that is, $\widehat{\mu} = 40.048$ and $\widehat{\phi} = .957$, and denote the resulting sample innovations as $\{\widehat{\epsilon}_1, \ldots, \widehat{\epsilon}_{100}\}$. To obtain one bootstrap sample, first randomly sample, with replacement, $n = 100$ values from the set of sample innovations; call the sampled values $\{\epsilon_1^*, \ldots, \epsilon_{100}^*\}$. Now, generate a bootstrapped data set sequentially by setting

$$\begin{aligned} x_1^* &= 40.048 + \epsilon_1^*/\sqrt{(1-.957^2)} \\ x_t^* &= 40.048 + .957(x_{t-1}^* - 40.048) + \epsilon_t^*, \quad t = 2, \ldots, n. \end{aligned} \tag{3.141}$$

Next, estimate the parameters as if the data were x_t^*. Call these estimates $\widehat{\mu}(1)$, $\widehat{\phi}(1)$, and $s_w^2(1)$. Repeat this process a large number, B, of times, generating a collection of bootstrapped parameter estimates, $\{\widehat{\mu}(b), \widehat{\phi}(b), s_w^2(b), b = 1, \ldots, B\}$. We can then approximate the finite sample distribution of an estimator from the bootstrapped parameter values. For example, we can approximate the distribution of $\widehat{\phi} - \phi$ by the empirical distribution of $\widehat{\phi}(b) - \widehat{\phi}$, for $b = 1, \ldots, B$.

Figure 3.11 shows the bootstrap histogram of 200 bootstrapped estimates of ϕ using the data shown in Figure 3.9. In addition, Figure 3.11 shows a density estimate based on the bootstrap histogram, as well as the asymptotic normal density that would have been used based on Proposition 3.10. Clearly, the bootstrap distribution of $\widehat{\phi}$ is closer to the distribution of $\widehat{\phi}$ shown in Figure 3.10 than to the asymptotic normal approximation. In particular, the mean of the distribution of $\widehat{\phi}(b)$ is .92 with a variance of $.05^2$. Some quantiles of this distribution are .83 (5%), .90 (25%), .93 (50%), .95 (75%), and .98 (95%).

To perform a similar bootstrap exercise in R, use the following commands. We note that the R estimation procedure is conditional on the first observation, so the first residual is not returned. To get around this problem,

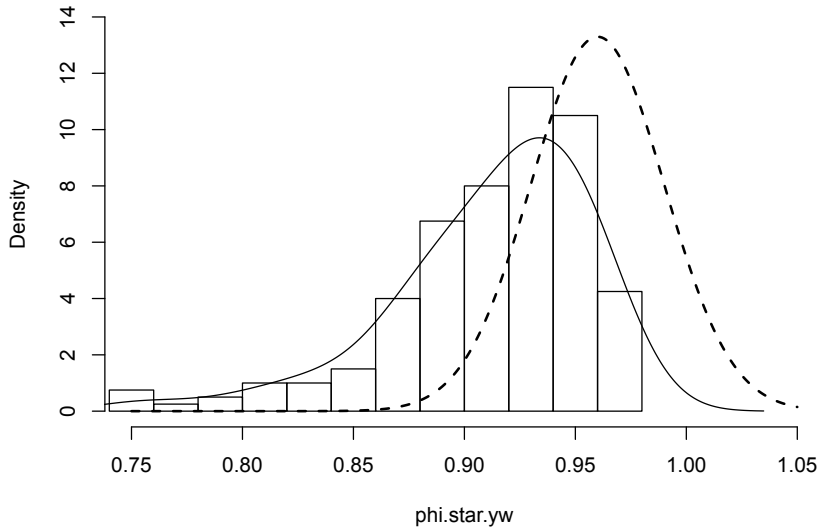

Fig. 3.11. Bootstrap histogram of $\widehat{\phi}$ based on 200 bootstraps; a density estimate based on the histogram (solid line) and the corresponding asymptotic normal density (dashed line).

we simply fix the first observation and bootstrap the remaining data. The simulated data are available in the file `ar1boot`, but you can simulate your own data as was done in the code that produced Figure 3.10.

```
x = ar1boot
m = mean(x)   # estimate of mu
fit = ar.yw(x, order=1)
phi = fit$ar  # estimate of phi
nboot = 200   # number of bootstrap replicates
resids = fit$resid[-1]  # the first resid is NA
x.star = x    # initialize x*
phi.star.yw = rep(NA, nboot)
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 }
hist(phi.star.yw, 10, main="", prob=TRUE, ylim=c(0,14),
    xlim=c(.75,1.05))
lines(density(phi.star.yw, bw=.02))
u = seq(.75, 1.05, by=.001)
lines(u, dnorm(u, mean=.96, sd=.03), lty="dashed", lwd=2)
```

3.7 Integrated Models for Nonstationary Data

In Chapters 1 and 2, 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 §2.2 we considered the model

$$x_t = \mu_t + y_t, \tag{3.142}$$

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 (3.142) is stochastic and slowly varying according to a random walk. That is,

$$\mu_t = \mu_{t-1} + v_t$$

where v_t is stationary. In this case,

$$\nabla x_t = v_t + \nabla y_t,$$

is stationary. If μ_t in (3.142) is a k-th order polynomial, $\mu_t = \sum_{j=0}^{k} \beta_j t^j$, then (Problem 3.27) the differenced series $\nabla^k y_t$ is stationary. Stochastic trend models can also lead to higher order differencing. For example, suppose

$$\mu_t = \mu_{t-1} + v_t \quad \text{and} \quad v_t = v_{t-1} + e_t,$$

where e_t is stationary. Then, $\nabla x_t = v_t + \nabla y_t$ is not stationary, but

$$\nabla^2 x_t = e_t + \nabla^2 y_t$$

is stationary.

The integrated ARMA, or ARIMA, model is a broadening of the class of ARMA models to include differencing.

Definition 3.11 *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

$$\phi(B)(1 - B)^d x_t = \theta(B) w_t. \tag{3.143}$$

If $E(\nabla^d x_t) = \mu$, we write the model as

$$\phi(B)(1 - B)^d x_t = \delta + \theta(B) w_t,$$

where $\delta = \mu(1 - \phi_1 - \cdots - \phi_p)$.

Because of the nonstationarity, care must be taken when deriving forecasts. For the sake of completeness, we discuss this issue briefly here, but we stress the fact that both the theoretical and computational aspects of the problem are best handled via state-space models. We discuss the theoretical details in Chapter 6. For information on the state-space based computational aspects in R, see the ARIMA help files (`?arima` and `?predict.Arima`); our scripts `sarima` and `sarima.for` are basically front ends for these R scripts.

It should be clear that, since $y_t = \nabla^d x_t$ is ARMA, we can use §3.5 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 used in §3.5, equation (3.86), works well. That is, the mean-squared prediction error can be approximated by

$$P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^{*2}, \tag{3.144}$$

where ψ^*_j is the coefficient of z^j in $\psi^*(z) = \theta(z)/\phi(z)(1-z)^d$.

To better understand integrated models, we examine the properties of some simple cases; Problem 3.29 covers the ARIMA$(1, 1, 0)$ case.

Example 3.36 Random Walk with Drift

To fix ideas, we begin by considering the random walk with drift model first presented in Example 1.11, that is,

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

for $t = 1, 2, \ldots$, and $x_0 = 0$. Technically, the model is not ARIMA, but we could include it trivially as an ARIMA$(0, 1, 0)$ model. Given data $x_1, \ldots, x_n$, the one-step-ahead forecast is given by

$$x^n_{n+1} = E(x_{n+1} \mid x_n, \ldots, x_1) = E(\delta + x_n + w_{n+1} \mid x_n, \ldots, x_1) = \delta + x_n.$$

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

$$x^n_{n+m} = m\,\delta + x_n, \tag{3.145}$$

To obtain the forecast errors, it is convenient to recall equation (1.4), i.e., $x_n = n\,\delta + \sum_{j=1}^n w_j$, in which case we may write

$$x_{n+m} = (n+m)\,\delta + \sum_{j=1}^{n+m} w_j = m\,\delta + x_n + \sum_{j=n+1}^{n+m} w_j.$$

From this it follows that the m-step-ahead prediction error is given by

$$P_{n+m}^{n} = E(x_{n+m} - x_{n+m}^{n})^2 = E\Big(\sum_{j=n+1}^{n+m} w_j\Big)^2 = m\,\sigma_w^2. \tag{3.146}$$

Hence, unlike the stationary case (see Example 3.22), as the forecast horizon grows, the prediction errors, (3.146), increase without bound and the forecasts follow a straight line with slope δ emanating from x_n. We note that (3.144) is exact in this case because $\psi^*(z) = 1/(1-z) = \sum_{j=0}^{\infty} z^j$ for $|z| < 1$, so that $\psi_j^* = 1$ for all j.

The w_t are Gaussian, so estimation is straightforward because the differenced data, say $y_t = \nabla x_t$, are independent and identically distributed normal variates with mean δ and variance σ_w^2. Consequently, optimal estimates of δ and σ_w^2 are the sample mean and variance of the y_t, respectively.

Example 3.37 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. In addition, the model leads to a frequently used, and abused, forecasting method called exponentially weighted moving averages (EWMA). We will write the model as

$$x_t = x_{t-1} + w_t - \lambda w_{t-1}, \tag{3.147}$$

with $|\lambda| < 1$, for $t = 1, 2, \ldots$, and $x_0 = 0$, because this model formulation is easier to work with here, and it leads to the standard representation for EWMA. We could have included a drift term in (3.147), as was done in the previous example, but for the sake of simplicity, we leave it out of the discussion. If we write

$$y_t = w_t - \lambda w_{t-1},$$

we may write (3.147) as $x_t = x_{t-1} + y_t$. Because $|\lambda| < 1$, y_t has an invertible representation, $y_t = \sum_{j=1}^{\infty} \lambda^j y_{t-j} + w_t$, and substituting $y_t = x_t - x_{t-1}$, we may write

$$x_t = \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{t-j} + w_t. \tag{3.148}$$

as an approximation for large t (put $x_t = 0$ for $t \le 0$). Verification of (3.148) is left to the reader (Problem 3.28). Using the approximation (3.148), we have that the approximate one-step-ahead predictor, using the notation of §3.5, is

$$\begin{aligned} \tilde{x}_{n+1} &= \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{n+1-j} \\ &= (1-\lambda)x_n + \lambda \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{n-j} \\ &= (1-\lambda)x_n + \lambda \tilde{x}_n. \end{aligned} \tag{3.149}$$

From (3.149), we see that the new forecast is a linear combination of the old forecast and the new observation. Based on (3.149) and the fact that we only observe $x_1, \ldots, x_n$, and consequently $y_1, \ldots, y_n$ (because $y_t = x_t - x_{t-1}$; $x_0 = 0$), the truncated forecasts are

$$\tilde{x}^n_{n+1} = (1-\lambda)x_n + \lambda\tilde{x}^{n-1}_n, \quad n \geq 1, \tag{3.150}$$

with $\tilde{x}^0_1 = x_1$ as an initial value. The mean-square prediction error can be approximated using (3.144) by noting that $\psi^*(z) = (1-\lambda z)/(1-z) = 1 + (1-\lambda)\sum_{j=1}^{\infty} z^j$ for $|z| < 1$; consequently, for large n, (3.144) leads to

$$P^n_{n+m} \approx \sigma^2_w[1 + (m-1)(1-\lambda)^2].$$

In EWMA, the parameter $1-\lambda$ is often called the smoothing parameter and is restricted to be between zero and one. Larger values of λ lead to smoother forecasts. 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. Unfortunately, as previously suggested, the method is often abused because some forecasters do not verify that the observations follow an IMA$(1,1)$ process, and often arbitrarily pick values of λ. In the following, we show how to generate 100 observations from an IMA(1,1) model with $\lambda = -\theta = .8$ and then calculate and display the fitted EWMA superimposed on the data. This is accomplished using the Holt-Winters command in R (see the help file `?HoltWinters` for details; no output is shown):

```
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)
```

3.8 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, we should construct a time plot of the data, and inspect the graph for any anomalies. If, for example, the variability in the data grows with time, it will be necessary to transform the data to stabilize the variance. In such cases, the Box–Cox class of power transformations, equation (2.37), could be employed. Also, the particular application might suggest an appropriate transformation. For example, suppose a process evolves as a fairly small and stable percent-change, such as an investment. For example, we might have

$$x_t = (1 + p_t)x_{t-1},$$

where x_t is the value of the investment at time t and p_t is the percentage-change from period $t-1$ to t, which may be negative. Taking logs we have

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

or

$$\nabla \log(x_t) = \log(1+p_t).$$

If the percent change p_t stays relatively small in magnitude, then $\log(1+p_t) \approx p_t$[8] and, thus,

$$\nabla \log(x_t) \approx p_t,$$

will be a relatively stable process. Frequently, $\nabla \log(x_t)$ is called the return or growth rate. This general idea was used in Example 3.32, and we will use it again in Example 3.38.

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. We have already addressed, in part, the problem of selecting d. 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 $\nabla^2 x_t$. 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 MA(1). In addition to time plots, the sample ACF can help in indicating whether differencing is needed. Because the polynomial $\phi(z)(1-z)^d$ has a unit root, the sample ACF, $\widehat{\rho}(h)$, will not decay to zero fast as h increases. Thus, a slow decay in $\widehat{\rho}(h)$ is an indication that differencing may be needed.

When preliminary values of d have been settled, the next step is to look at the sample ACF and PACF of $\nabla^d x_t$ for whatever values of d have been chosen. Using Table 3.1 as a guide, preliminary values of p and q are chosen. Recall that, if $p=0$ and $q>0$, the ACF cuts off after lag q, and the PACF tails off. If $q=0$ and $p>0$, the PACF cuts off after lag p, and the ACF tails off. If $p>0$ and $q>0$, both the ACF and PACF will tail 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. With this in mind, we should not worry about being so precise at this stage of the model fitting. At this stage, a few preliminary values of p, d, and q should be at hand, and we can start estimating the parameters.

Example 3.38 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

[8] $\log(1+p) = p - \frac{p^2}{2} + \frac{p^3}{3} - \cdots$ for $-1 < p \leq 1$. If p is a small percent-change, then the higher-order terms in the expansion are negligible.

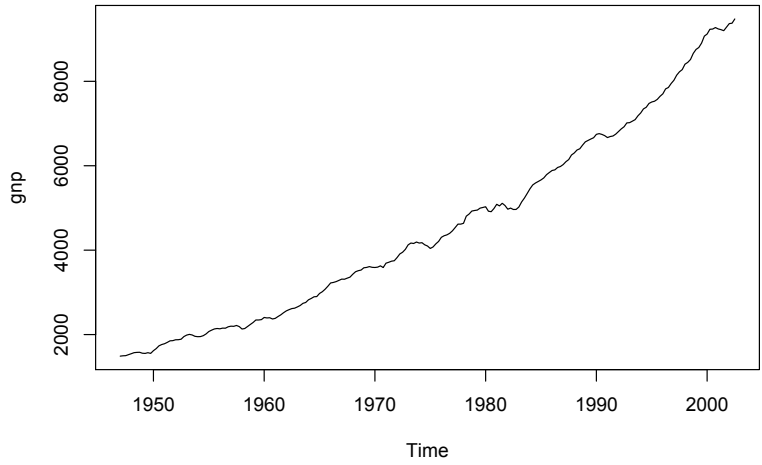

Fig. 3.12. Quarterly U.S. GNP from 1947(1) to 2002(3).

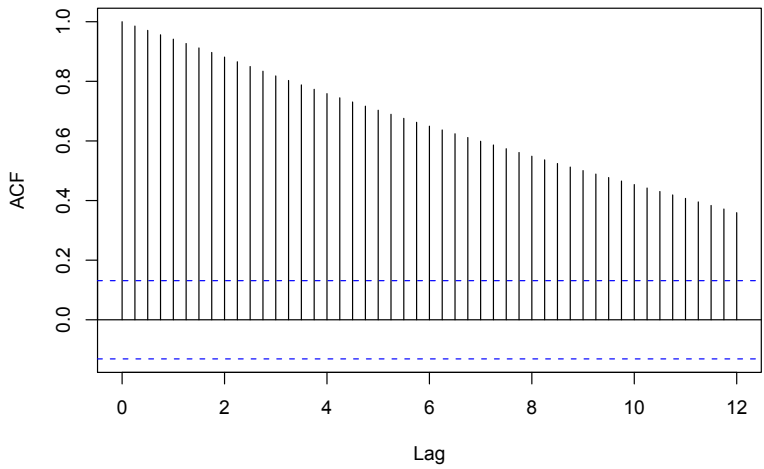

Fig. 3.13. Sample ACF of the GNP data. Lag is in terms of years.

national product in billions of chained 1996 dollars and have been seasonally adjusted. The data were obtained from the Federal Reserve Bank of St. Louis (`http://research.stlouisfed.org/`). Figure 3.12 shows a plot of the data, say, y_t. Because strong trend hides any other effect, it is not clear from Figure 3.12 that the variance is increasing with time. For the purpose of demonstration, the sample ACF of the data is displayed in Figure 3.13. Figure 3.14 shows the first difference of the data, ∇y_t, and now that the trend has been removed we are able to notice that the variability in the second half of the data is larger than in the first half of the data. Also, it appears as though a trend is still present after differencing. The growth

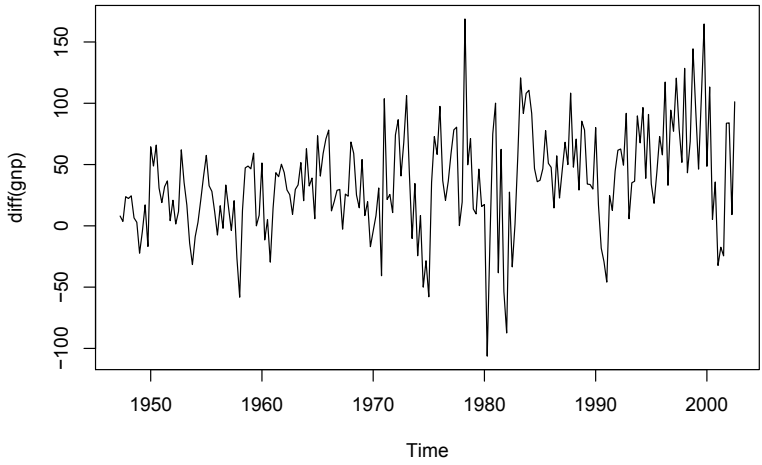

Fig. 3.14. First difference of the U.S. GNP data.

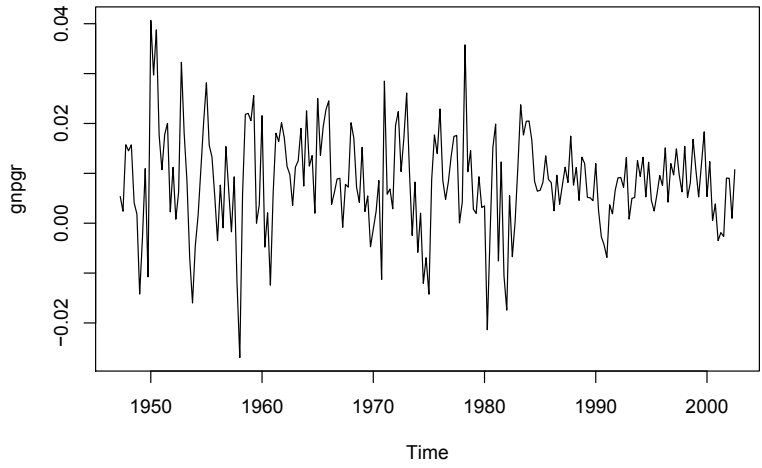

Fig. 3.15. U.S. GNP quarterly growth rate.

rate, say, $x_t = \nabla \log(y_t)$, is plotted in Figure 3.15, and, appears to be a stable process. Moreover, we may interpret the values of x_t as the percentage quarterly growth of U.S. GNP.

The sample ACF and PACF of the quarterly growth rate are plotted in Figure 3.16. 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. 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

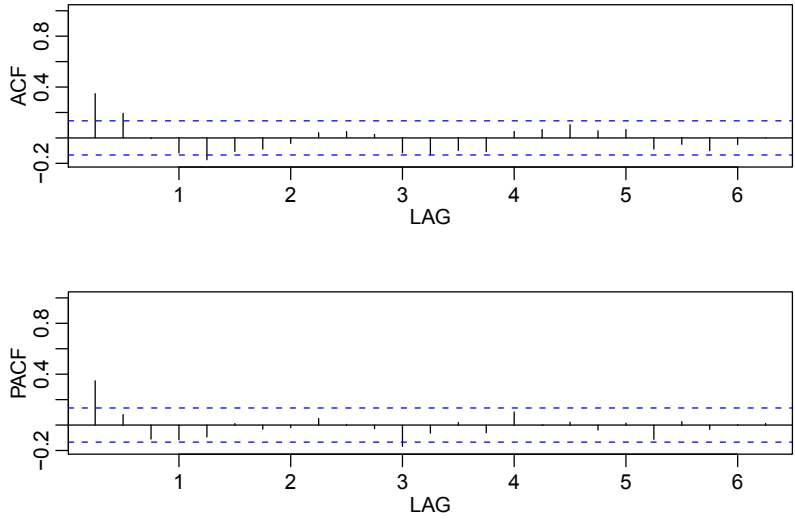

Fig. 3.16. Sample ACF and PACF of the GNP quarterly growth rate. Lag is in terms of years.

lag 1. This suggests an AR(1) model for the growth rate, or ARIMA$(1,1,0)$ for log GNP. As a preliminary analysis, we will fit both models.

Using MLE to fit the MA(2) model for the growth rate, x_t, the estimated model is

$$x_t = .008_{(.001)} + .303_{(.065)}\widehat{w}_{t-1} + .204_{(.064)}\widehat{w}_{t-2} + \widehat{w}_t, \tag{3.151}$$

where $\widehat{\sigma}_w = .0094$ is based on 219 degrees of freedom. The values in parentheses are the corresponding estimated standard errors. All of the regression coefficients are significant, including the constant. *We make a special note of this because, as a default, some computer packages do not fit a constant in a differenced model.* That is, these packages assume, by default, 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 3.15). We leave it to the reader to investigate what happens when the constant is not included.

The estimated AR(1) model is

$$x_t = .008_{(.001)}\,(1 - .347) + .347_{(.063)}x_{t-1} + \widehat{w}_t, \tag{3.152}$$

where $\widehat{\sigma}_w = .0095$ on 220 degrees of freedom; note that the constant in (3.152) is $.008\,(1 - .347) = .005$.

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

(3.151) and (3.152)? In fact, the fitted models are nearly the same. To show this, consider an AR(1) model of the form in (3.152) 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. Thus,

$$x_t \approx .35w_{t-1} + .12w_{t-2} + w_t,$$

which is similar to the fitted MA(2) model in (3.152).

The analysis can be performed in R as follows.

```
plot(gnp)
acf2(gnp, 50)
gnpgr = diff(log(gnp))  # growth rate
plot(gnpgr)
acf2(gnpgr, 24)
sarima(gnpgr, 1, 0, 0)  # AR(1)
sarima(gnpgr, 0, 0, 2)  # MA(2)
ARMAtoMA(ar=.35, ma=0, 10)  # prints psi-weights
```

The next step in model fitting is diagnostics. This investigation includes the analysis of the residuals as well as model comparisons. Again, the first step involves a time plot of the innovations (or residuals), $x_t - \widehat{x}_t^{t-1}$, or of the standardized innovations

$$e_t = \left(x_t - \widehat{x}_t^{t-1}\right) / \sqrt{\widehat{P}_t^{t-1}}, \tag{3.153}$$

where $\widehat{x}_t^{t-1}$ is the one-step-ahead prediction of x_t based on the fitted model and $\widehat{P}_t^{t-1}$ is the estimated one-step-ahead error variance. If the model fits well, the standardized residuals should behave as an iid sequence with mean zero and variance one. The time plot should be inspected for any obvious departures from this assumption. Unless the time series is Gaussian, it is not enough that the residuals are uncorrelated. For example, it is possible in the non-Gaussian case to have an uncorrelated process for which values contiguous in time are highly dependent. As an example, we mention the family of GARCH models that are discussed in Chapter 5.

Investigation of marginal normality can be accomplished visually by looking at a histogram of the residuals. In addition to this, a normal probability plot or a Q-Q plot can help in identifying departures from normality. See Johnson and Wichern (1992, Chapter 4) for details of this test as well as additional tests for multivariate normality.

There are several tests of randomness, for example the runs test, that could be applied to the residuals. We could also inspect the sample autocorrelations of the residuals, say, $\widehat{\rho}_e(h)$, for any patterns or large values. Recall that, for a white noise sequence, the sample autocorrelations are approximately independently and normally distributed with zero means and variances $1/n$. Hence, a

good check on the correlation structure of the residuals is to plot $\widehat{\rho}_e(h)$ versus h along with the error bounds of $\pm 2/\sqrt{n}$. The residuals from a model fit, however, will not quite have the properties of a white noise sequence and the variance of $\widehat{\rho}_e(h)$ can be much less than $1/n$. Details can be found in Box and Pierce (1970) and McLeod (1978). This part of the diagnostics can be viewed as a visual inspection of $\widehat{\rho}_e(h)$ with the main concern being the detection of obvious departures from the independence assumption.

In addition to plotting $\widehat{\rho}_e(h)$, we can perform a general test that takes into consideration the magnitudes of $\widehat{\rho}_e(h)$ as a group. For example, it may be the case that, individually, each $\widehat{\rho}_e(h)$ is small in magnitude, say, each one is just slightly less that $2/\sqrt{n}$ in magnitude, but, collectively, the values are large. The Ljung–Box–Pierce Q-statistic given by

$$Q = n(n+2)\sum_{h=1}^{H} \frac{\widehat{\rho}_e^2(h)}{n-h} \tag{3.154}$$

can be used to perform such a test. The value H in (3.154) is chosen somewhat arbitrarily, typically, $H = 20$. Under the null hypothesis of model adequacy, asymptotically $(n \to \infty)$, $Q \sim \chi^2_{H-p-q}$. 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. Details can be found in Box and Pierce (1970), Ljung and Box (1978), and Davies et al. (1977). The basic idea is that if w_t is white noise, then by Property 1.1, $n\widehat{\rho}_w^2(h)$, for $h = 1, \ldots, H$, are asymptotically independent χ^2_1 random variables. This means that $n\sum_{h=1}^{H} \widehat{\rho}_w^2(h)$ is approximately a χ^2_H random variable. Because the test involves the ACF of residuals from a model fit, there is a loss of $p+q$ degrees of freedom; the other values in (3.154) are used to adjust the statistic to better match the asymptotic chi-squared distribution.

Example 3.39 Diagnostics for GNP Growth Rate Example

We will focus on the MA(2) fit from Example 3.38; the analysis of the AR(1) residuals is similar. Figure 3.17 displays a plot of the standardized residuals, the ACF of the residuals, a boxplot of the standardized residuals, and the p-values associated with the Q-statistic, (3.154), at lags $H = 3$ through $H = 20$ (with corresponding degrees of freedom $H - 2$).

Inspection of the time plot of the standardized residuals in Figure 3.17 shows no obvious patterns. Notice that there are outliers, however, with a few values exceeding 3 standard deviations in magnitude. The ACF of the standardized residuals shows no apparent departure from the model assumptions, and the Q-statistic is never significant at the lags shown. The normal Q-Q plot of the residuals shows departure from normality at the tails due to the outliers that occurred primarily in the 1950s and the early 1980s.

The model appears to fit well except for the fact that a distribution with heavier tails than the normal distribution should be employed. We discuss

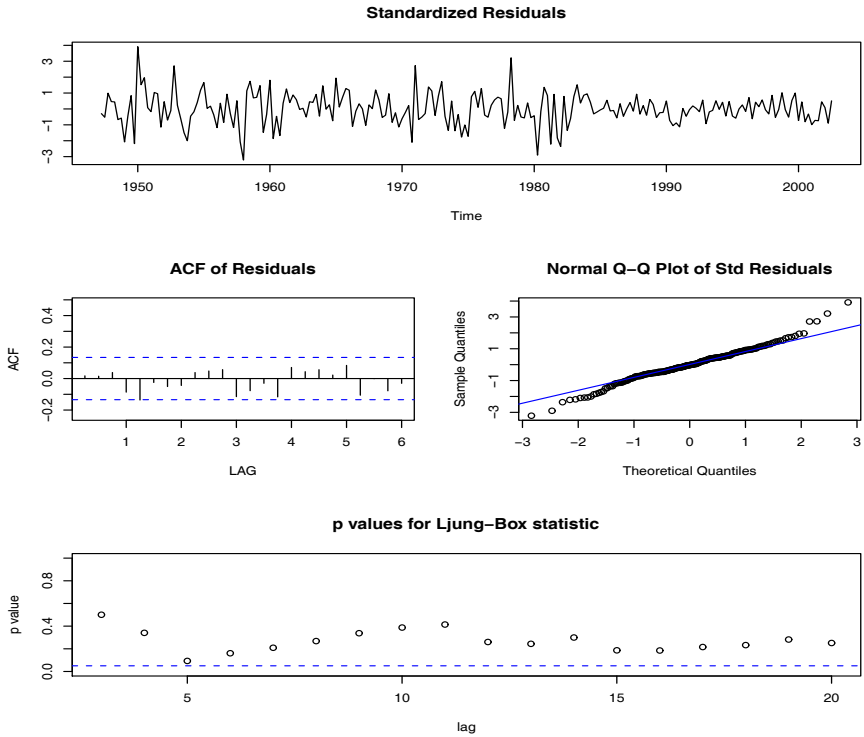

Fig. 3.17. Diagnostics of the residuals from MA(2) fit on GNP growth rate.

some possibilities in Chapters 5 and 6. The diagnostics shown in Figure 3.17 are a by-product of the `sarima` command from the previous example.[9]

Example 3.40 Diagnostics for the Glacial Varve Series

In Example 3.32, 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 3.18.

To adjust for this problem, we fit an ARIMA$(1, 1, 1)$ to the logged varve data and obtained the estimates

$$\widehat{\phi} = .23_{(.05)},\ \widehat{\theta} = -.89_{(.03)},\ \widehat{\sigma}_w^2 = .23.$$

Hence the AR term is significant. The Q-statistic p-values for this model are also displayed in Figure 3.18, 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

[9] The script `tsdiag` is available in R to run diagnostics for an ARIMA object, however, the script has errors and we do not recommend using it.

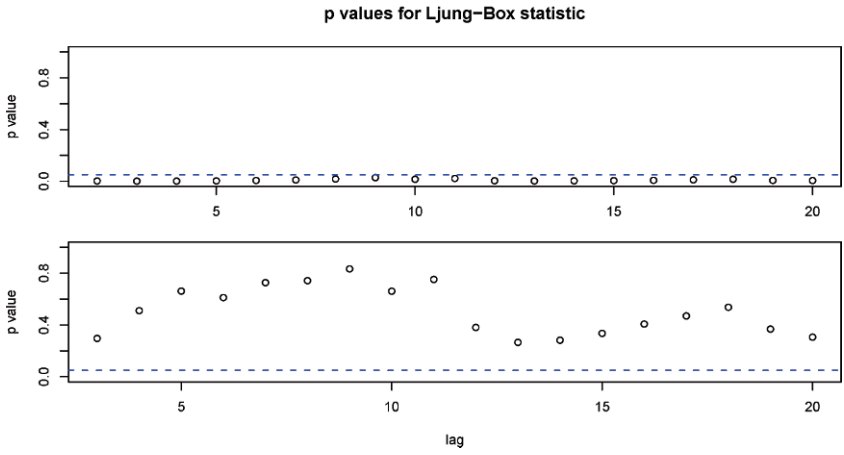

Fig. 3.18. Q-statistic p-values for the ARIMA$(0,1,1)$ fit [top] and the ARIMA$(1,1,1)$ fit [bottom] to the logged varve data.

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:

```
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)
```

In Example 3.38, 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 3.41 A Problem with Overfitting

Figure 3.19 shows the U.S. population by official census, every ten years from 1910 to 1990, as points. If we use these nine observations to predict the future population, we can use an eight-degree polynomial so the fit to the nine observations is perfect. 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, which is plotted in the figure, passes through the nine observations. The model predicts that the population of the United States will be close to zero in the year 2000, and will cross zero sometime in the year 2002!

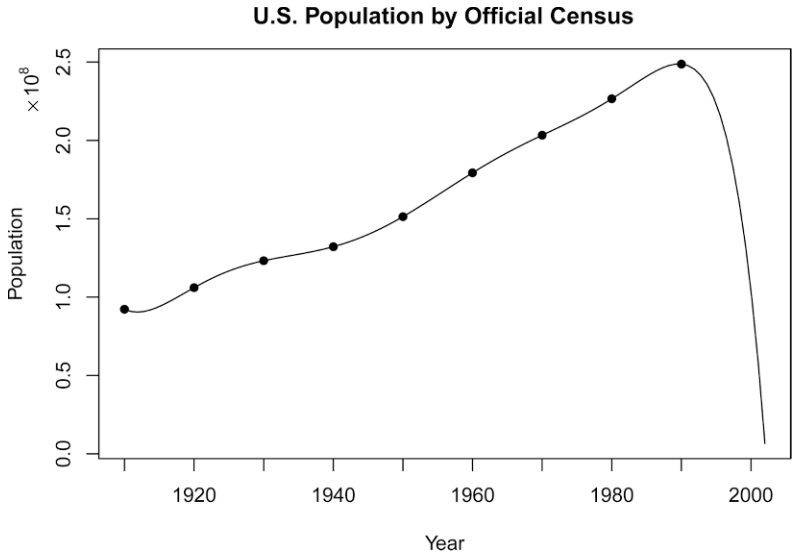

Fig. 3.19. A perfect fit and a terrible forecast.

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 §2.2 in the context of regression models.

Example 3.42 Model Choice for the U.S. GNP Series

Returning to the analysis of the U.S. GNP data presented in Examples 3.38 and 3.39, recall that two models, an AR(1) and an MA(2), fit the GNP growth rate well. 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 displayed at the end of Example 3.38, but for convenience, we display them again here (recall the growth rate data are in `gnpgr`):

```
1 sarima(gnpgr, 1, 0, 0)  # AR(1)
     $AIC: -8.294403   $AICc: -8.284898   $BIC: -9.263748
2 sarima(gnpgr, 0, 0, 2)  # MA(2)
     $AIC: -8.297693   $AICc: -8.287854   $BIC: -9.251711
```

The AIC and AICc both prefer the MA(2) fit, whereas the BIC prefers the simpler AR(1) model. It is often the case that the BIC will select a model of smaller order than the AIC or AICc. It would not be unreasonable in this case to retain the AR(1) because pure autoregressive models are easier to work with.

3.9 Multiplicative Seasonal ARIMA Models

In this section, we introduce several modifications made to the ARIMA model to account for seasonal and nonstationary 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

$$\Phi_P(B^s)x_t = \Theta_Q(B^s)w_t, \tag{3.155}$$

with the following definition.

Definition 3.12 *The operators*

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \tag{3.156}$$

and

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \tag{3.157}$$

are the **seasonal autoregressive operator** *and the* **seasonal moving average operator** *of orders P and Q, respectively, with seasonal period s.*

Analogous to the properties of nonseasonal ARMA models, the pure seasonal ARMA$(P, Q)_s$ is causal only when the roots of $\Phi_P(z^s)$ lie outside the unit circle, and it is invertible only when the roots of $\Theta_Q(z^s)$ lie outside the unit circle.

Example 3.43 A Seasonal ARMA Series

A first-order seasonal autoregressive moving average series that might run over months could be written as

$$(1 - \Phi B^{12})x_t = (1 + \Theta B^{12})w_t$$

or

$$x_t = \Phi x_{t-12} + w_t + \Theta w_{t-12}.$$

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 from the above form 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 $|\Phi| < 1$, and the invertible condition requires $|\Theta| < 1$.

Table 3.3. 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.

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

$$\rho(\pm 12k) = \Phi^k, \quad k = 0, 1, 2, \ldots \ .$$

These results can be verified using the general result that $\gamma(h) = \Phi\gamma(h - 12)$, 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.

As an initial diagnostic criterion, we can use the properties for the pure seasonal autoregressive and moving average series listed in Table 3.3. These properties may be considered as generalizations of the properties for nonseasonal models that were presented in Table 3.1.

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

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t \tag{3.158}$$

as the overall model. Although the diagnostic properties in Table 3.3 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 Tables 3.1 and 3.3. In fitting such models, focusing on the seasonal autoregressive and moving average components first generally leads to more satisfactory results.

Example 3.44 A Mixed Seasonal Model

Consider an ARMA$(0,1) \times (1,0)_{12}$ 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

$$\gamma(0) = \frac{1+\theta^2}{1-\Phi^2}\,\sigma_w^2.$$

In addition, 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 ACF for this model is

$$\begin{aligned} \rho(12h) &= \Phi^h \quad h = 1, 2, \ldots \\ \rho(12h-1) = \rho(12h+1) &= \frac{\theta}{1+\theta^2}\Phi^h \quad h = 0, 1, 2, \ldots, \\ \rho(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

The ACF and PACF for this model, with $\Phi = .8$ and $\theta = -.5$, are shown in Figure 3.20. These type of correlation relationships, although idealized here, are typically seen with seasonal data.

To reproduce Figure 3.20 in R, use the following commands:

```
phi = c(rep(0,11),.8)
ACF = ARMAacf(ar=phi, ma=-.5, 50)[-1]      # [-1] removes 0 lag
PACF = ARMAacf(ar=phi, ma=-.5, 50, pacf=TRUE)
par(mfrow=c(1,2))
plot(ACF,  type="h", xlab="lag", ylim=c(-.4,.8));  abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.4,.8));  abline(h=0)
```

Seasonal nonstationarity can occur, for example, when the process is nearly periodic in the season. For example, with average monthly temperatures over the years, each January would be approximately the same, each February would be approximately the same, and so on. In this case, we might think of average monthly temperature x_t as being modeled as

$$x_t = S_t + w_t,$$

where S_t is a seasonal component that varies slowly from one year to the next, according to a random walk,

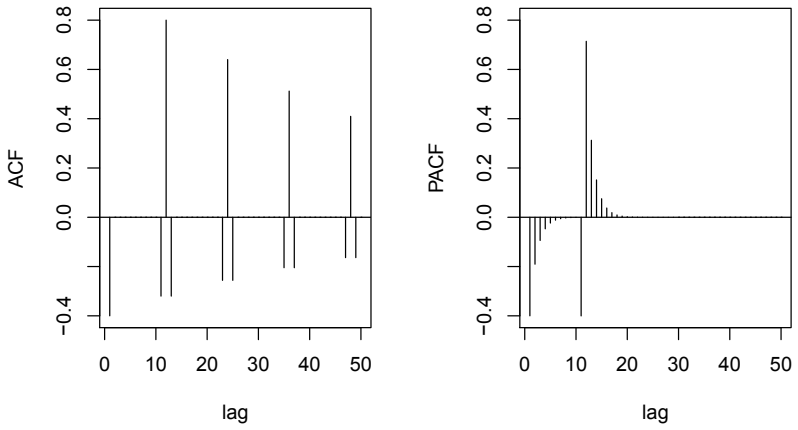

Fig. 3.20. ACF and PACF of the mixed seasonal ARMA model $x_t = .8x_{t-12} + w_t - .5w_{t-1}$.

$$S_t = S_{t-12} + v_t.$$

In this model, w_t and v_t are uncorrelated white noise processes. The tendency of data to follow this type of model will be exhibited in a sample ACF that is large and decays very slowly at lags $h = 12k$, for $k = 1, 2, \ldots$. If we subtract the effect of successive years from each other, we find that

$$(1 - B^{12})x_t = x_t - x_{t-12} = v_t + w_t - w_{t-12}.$$

This model is a stationary $\text{MA}(1)_{12}$, and its ACF will have a peak only at lag 12. In general, seasonal differencing can be indicated when the ACF decays slowly at multiples of some season s, but is negligible between the periods. Then, a seasonal difference of order D is defined as

$$\nabla_s^D x_t = (1 - B^s)^D x_t, \tag{3.159}$$

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 3.13 *The multiplicative* **seasonal autoregressive integrated moving average** *model, or* **SARIMA** *model is given by*

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t, \tag{3.160}$$

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 polynomials $\phi(B)$ and $\theta(B)$ of orders p and q, respectively [see (3.5) and (3.18)], and the seasonal autoregressive and moving average components by $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ [see (3.156) and (3.157)] of orders P and Q and ordinary and seasonal difference components by $\nabla^d = (1 - B)^d$ and $\nabla_s^D = (1 - B^s)^D$.*

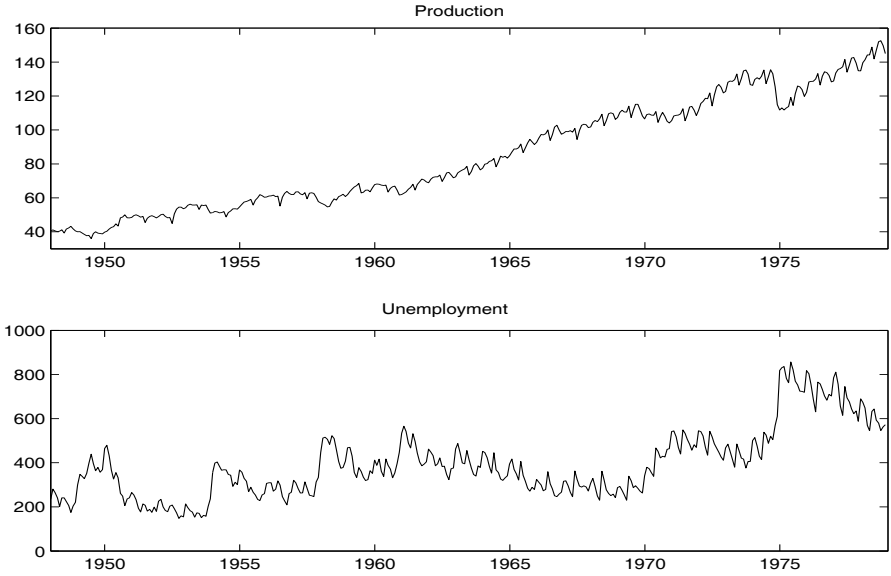

Fig. 3.21. Values of the Monthly Federal Reserve Board Production Index and Unemployment (1948-1978, $n = 372$ months).

Example 3.45 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 ARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$ in the notation given above, where the seasonal fluctuations occur every 12 months. Then, the model (3.160) becomes

$$(1 - B^{12})(1 - B)x_t = (1 + \Theta B^{12})(1 + \theta B)w_t. \tag{3.161}$$

Expanding both sides of (3.161) 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 from all of those represented by the general form (3.160) is a daunting task, and we usually

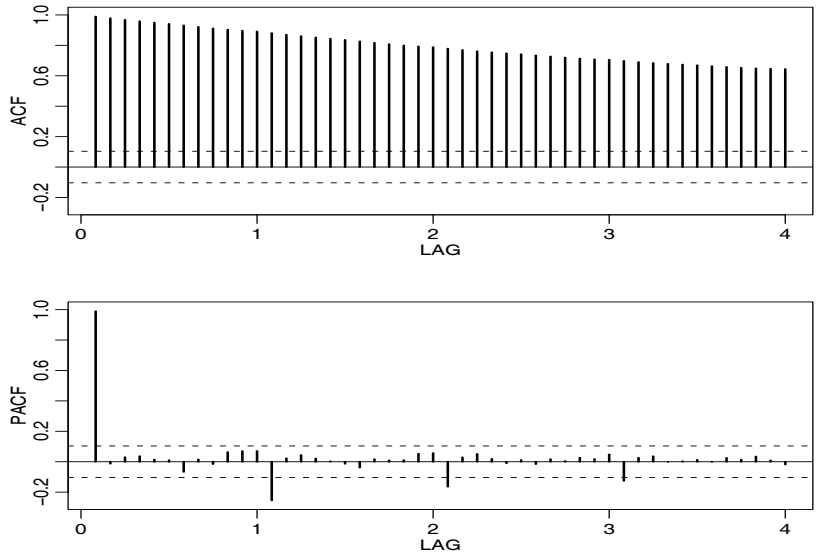

Fig. 3.22. ACF and PACF of the production series.

think first in terms of finding difference operators that produce a roughly stationary series and then in terms of finding a set of simple autoregressive moving average or multiplicative seasonal ARMA to fit the resulting residual series. Differencing operations are applied first, and then the residuals are constructed from a series of reduced length. Next, the ACF and the PACF of these residuals are evaluated. Peaks that appear in these functions can often be eliminated by fitting an autoregressive or moving average component in accordance with the general properties of Tables 3.1 and 3.2. In considering whether the model is satisfactory, the diagnostic techniques discussed in §3.8 still apply.

Example 3.46 The Federal Reserve Board Production Index

A problem of great interest in economics involves first identifying a model within the Box–Jenkins class for a given time series and then producing forecasts based on the model. For example, we might consider applying this methodology to the Federal Reserve Board Production Index shown in Figure 3.21. For demonstration purposes only, the ACF and PACF for this series are shown in Figure 3.22. We note that the trend in the data, the slow decay in the ACF, and the fact that the PACF at the first lag is nearly 1, all indicate nonstationary behavior.

Following the recommended procedure, a first difference was taken, and the ACF and PACF of the first difference

$$\nabla x_t = x_t - x_{t-1}$$

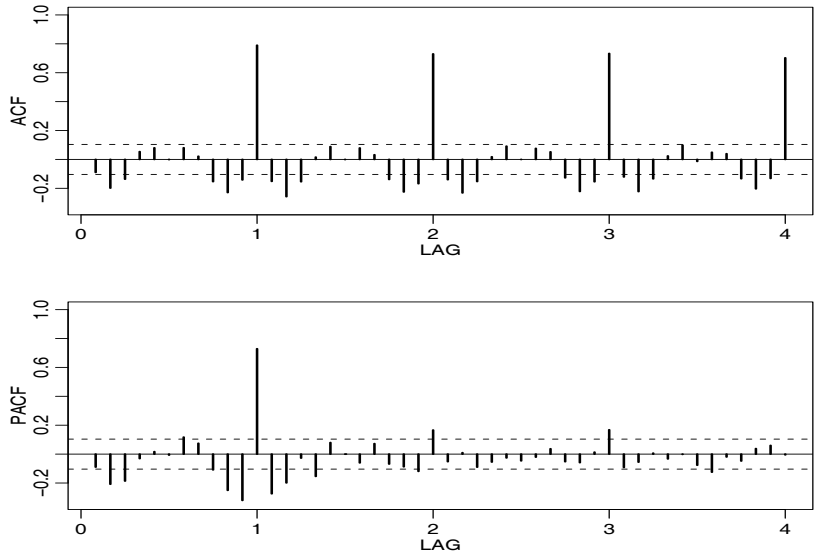

Fig. 3.23. ACF and PACF of differenced production, $(1 - B)x_t$.

are shown in Figure 3.23. Noting the peaks at seasonal lags, $h = 1s, 2s, 3s, 4s$ where $s = 12$ (i.e., $h = 12, 24, 36, 48$) with relatively slow decay suggests a seasonal difference. Figure 3.24 shows the ACF and PACF of the seasonal difference of the differenced production, say,

$$\nabla_{12}\nabla x_t = (1 - B^{12})(1 - B)x_t.$$

First, concentrating on the seasonal ($s = 12$) lags, the characteristics of the ACF and PACF of this series tend to show a strong peak at $h = 1s$ in the autocorrelation function, with smaller peaks appearing at $h = 2s, 3s$, combined with peaks at $h = 1s, 2s, 3s, 4s$ in the partial autocorrelation function. It appears that either

(i) the ACF is cutting off after lag $1s$ and the PACF is tailing off in the seasonal lags,
(ii) the ACF is cutting off after lag $3s$ and the PACF is tailing off in the seasonal lags, or
(iii) the ACF and PACF are both tailing off in the seasonal lags.

Using Table 3.3, this suggests either (i) an SMA of order $Q = 1$, (ii) an SMA of order $Q = 3$, or (iii) an SARMA of orders $P = 2$ (because of the two spikes in the PACF) and $Q = 1$.

Next, inspecting the ACF and the PACF at the within season lags, $h = 1, \ldots, 11$, it appears that either (a) both the ACF and PACF are tailing off, or (b) that the PACF cuts off at lag 2. Based on Table 3.1, this result indicates that we should either consider fitting a model (a) with both $p > 0$ and $q > 0$ for the nonseasonal components, say $p = 1, q = 1$, or (b) $p =$

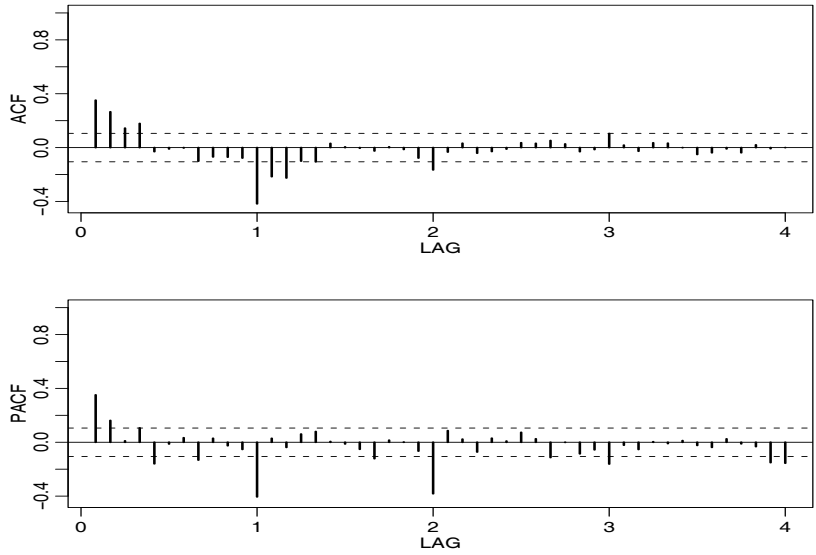

Fig. 3.24. ACF and PACF of first differenced and then seasonally differenced production, $(1-B)(1-B^{12})x_t$.

$2, q = 0$. It turns out that there is little difference in the results for case (a) and (b), but that (b) is slightly better, so we will concentrate on case (b).

Fitting the three models suggested by these observations we obtain:

(i) ARIMA$(2,1,0) \times (0,1,1)_{12}$:
AIC= 1.372, AICc= 1.378, BIC= .404

(ii) ARIMA$(2,1,0) \times (0,1,3)_{12}$:
AIC= 1.299, AICc= 1.305, BIC= .351

(iii) ARIMA$(2,1,0) \times (2,1,1)_{12}$:
AIC= 1.326, AICc= 1.332, BIC= .379

The ARIMA$(2,1,0) \times (0,1,3)_{12}$ is the preferred model, and the fitted model in this case is

$$(1 - .30_{(.05)}B - .11_{(.05)}B^2)\nabla_{12}\nabla\widehat{x}_t = (1 - .74_{(.05)}B^{12} - .14_{(.06)}B^{24} + .28_{(.05)}B^{36})\widehat{w}_t$$

with $\widehat{\sigma}_w^2 = 1.312$.

The diagnostics for the fit are displayed in Figure 3.25. We note the few outliers in the series as exhibited in the plot of the standardized residuals and their normal Q-Q plot, and a small amount of autocorrelation that still remains (although not at the seasonal lags) but otherwise, the model fits well. Finally, forecasts based on the fitted model for the next 12 months are shown in Figure 3.26.

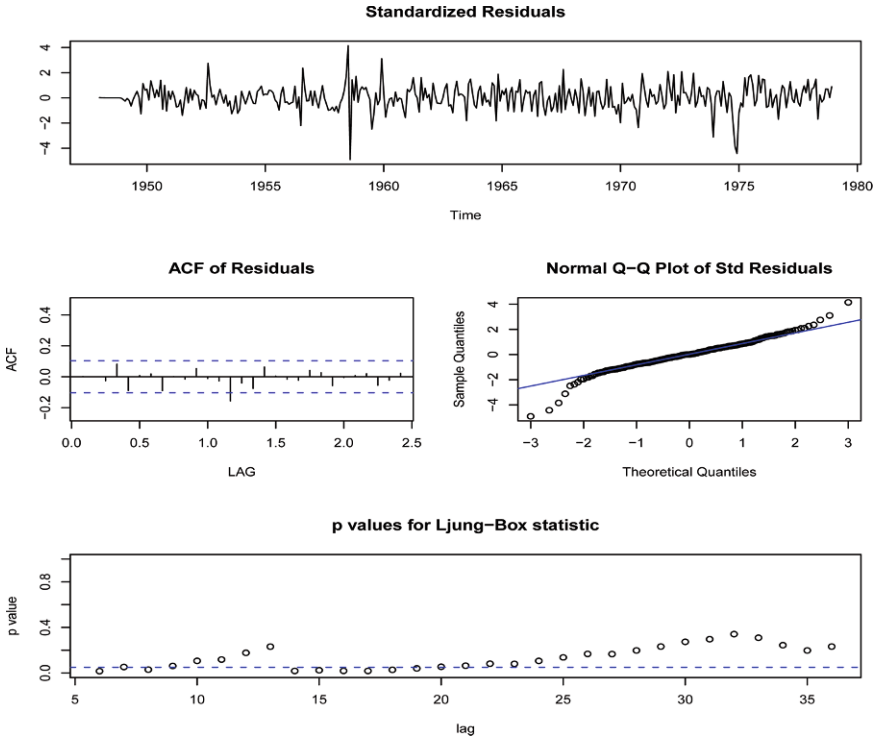

Fig. 3.25. Diagnostics for the ARIMA$(2,1,0) \times (0,1,3)_{12}$ fit on the Production Index.

The following R code can be used to perform the analysis.

```
acf2(prodn, 48)
acf2(diff(prodn), 48)
acf2(diff(diff(prodn), 12), 48)
sarima(prodn, 2, 1, 1, 0, 1, 3, 12)  # fit model (ii)
sarima.for(prodn, 12, 2, 1, 1, 0, 1, 3, 12) # forecast
```

Problems

Section 3.2

3.1 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?

3.2 Let w_t be white noise with variance σ_w^2 and let $|\phi| < 1$ be a constant. Consider the process $x_1 = w_1$, and

$$x_t = \phi x_{t-1} + w_t, \quad t = 2, 3, \ldots \ .$$

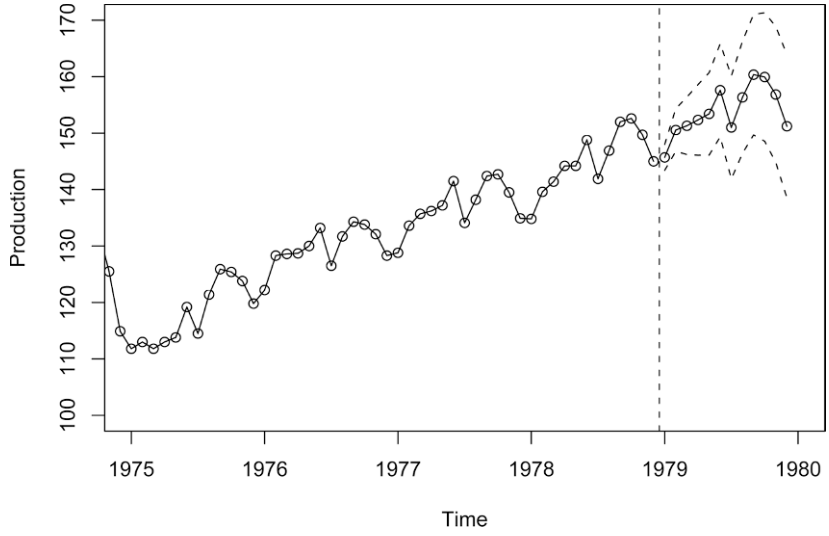

Fig. 3.26. Forecasts and limits for production index. The vertical dotted line separates the data from the predictions.

(a) Find the mean and the variance of $\{x_t,\ t = 1, 2, \ldots\}$. Is x_t stationary?
(b) Show

$$\text{corr}(x_t, x_{t-h}) = \phi^h \left[\frac{\text{var}(x_{t-h})}{\text{var}(x_t)}\right]^{1/2}$$

for $h \geq 0$.
(c) Argue that for large t,

$$\text{var}(x_t) \approx \frac{\sigma_w^2}{1-\phi^2}$$

and

$$\text{corr}(x_t, x_{t-h}) \approx \phi^h, \quad h \geq 0,$$

so in a sense, x_t is "asymptotically stationary."
(d) 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.
(e) Now suppose $x_1 = w_1/\sqrt{1-\phi^2}$. Is this process stationary?

3.3 Verify the calculations made in Example 3.3:

(a) Let $x_t = \phi x_{t-1} + w_t$ where $|\phi| > 1$ and $w_t \sim$ iid $\text{N}(0, \sigma_w^2)$. Show $E(x_t) = 0$ and $\gamma_x(h) = \sigma_w^2 \phi^{-2}\, \phi^{-h}/(1-\phi^{-2})$.
(b) Let $y_t = \phi^{-1} y_{t-1} + v_t$ where $v_t \sim$ iid $\text{N}(0, \sigma_w^2 \phi^{-2})$ and ϕ and σ_w are as in part (a). Argue that y_t is causal with the same mean function and autocovariance function as x_t.

3.4 Identify the following models as ARMA(p, q) models (watch out for parameter redundancy), and determine whether they are causal and/or invertible:

(a) $x_t = .80x_{t-1} - .15x_{t-2} + w_t - .30w_{t-1}$.
(b) $x_t = x_{t-1} - .50x_{t-2} + w_t - w_{t-1}$.

3.5 Verify the causal conditions for an AR(2) model given in (3.28). That is, show that an AR(2) is causal if and only if (3.28) holds.

Section 3.3

3.6 For the AR(2) model given by $x_t = -.9x_{t-2} + w_t$, find the roots of the autoregressive polynomial, and then sketch the ACF, $\rho(h)$.

3.7 For the AR(2) series shown below, use the results of Example 3.9 to determine a set of difference equations that can be used to find the ACF $\rho(h)$, $h = 0, 1, \ldots$; solve for the constants in the ACF using the initial conditions. Then plot the ACF values to lag 10 (use `ARMAacf` as a check on your answers).

(a) $x_t + 1.6x_{t-1} + .64x_{t-2} = w_t$.
(b) $x_t - .40x_{t-1} - .45x_{t-2} = w_t$.
(c) $x_t - 1.2x_{t-1} + .85x_{t-2} = w_t$.

Section 3.4

3.8 Verify the calculations for the autocorrelation function of an ARMA$(1, 1)$ process given in Example 3.13. Compare the form with that of the ACF for the ARMA$(1, 0)$ and the ARMA$(0, 1)$ series. Plot (or sketch) the ACFs of the three series on the same graph for $\phi = .6$, $\theta = .9$, and comment on the diagnostic capabilities of the ACF in this case.

3.9 Generate $n = 100$ observations from each of the three models discussed in Problem 3.8. Compute the sample ACF for each model and compare it to the theoretical values. Compute the sample PACF for each of the generated series and compare the sample ACFs and PACFs with the general results given in Table 3.1.

Section 3.5

3.10 Let x_t represent the cardiovascular mortality series (`cmort`) discussed in Chapter 2, Example 2.2.

(a) Fit an AR(2) to x_t using linear regression as in Example 3.17.
(b) Assuming the fitted model in (a) 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.

3.11 Consider the MA(1) series

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

where w_t is white noise with variance σ_w^2.

(a) Derive the minimum mean-square error one-step forecast based on the infinite past, and determine the mean-square error of this forecast.
(b) Let $\widetilde{x}_{n+1}^n$ be the truncated one-step-ahead forecast as given in (3.92). Show that

$$E\left[(x_{n+1} - \widetilde{x}_{n+1}^n)^2\right] = \sigma^2(1 + \theta^{2+2n}).$$

Compare the result with (a), and indicate how well the finite approximation works in this case.

3.12 In the context of equation (3.63), show that, if $\gamma(0) > 0$ and $\gamma(h) \to 0$ as $h \to \infty$, then Γ_n is positive definite.

3.13 Suppose x_t is stationary with zero mean and recall the definition of the PACF given by (3.55) and (3.56). That is, let

$$\epsilon_t = x_t - \sum_{i=1}^{h-1} a_i x_{t-i}$$

and

$$\delta_{t-h} = x_{t-h} - \sum_{j=1}^{h-1} b_j x_{t-j}$$

be the two residuals where $\{a_1, \ldots, a_{h-1}\}$ and $\{b_1, \ldots, b_{h-1}\}$ are chosen so that they minimize the mean-squared errors

$$E[\epsilon_t^2] \quad \text{and} \quad E[\delta_{t-h}^2].$$

The PACF at lag h was defined as the cross-correlation between ϵ_t and δ_{t-h}; that is,

$$\phi_{hh} = \frac{E(\epsilon_t \delta_{t-h})}{\sqrt{E(\epsilon_t^2)E(\delta_{t-h}^2)}}.$$

Let R_h be the $h \times h$ matrix with elements $\rho(i-j), i, j = 1, \ldots, h$, and let $\boldsymbol{\rho}_h = (\rho(1), \rho(2), \ldots, \rho(h))'$ be the vector of lagged autocorrelations, $\rho(h) = \mathrm{corr}(x_{t+h}, x_t)$. Let $\widetilde{\boldsymbol{\rho}}_h = (\rho(h), \rho(h-1), \ldots, \rho(1))'$ be the reversed vector. In addition, let x_t^h denote the BLP of x_t given $\{x_{t-1}, \ldots, x_{t-h}\}$:

$$x_t^h = \alpha_{h1} x_{t-1} + \cdots + \alpha_{hh} x_{t-h},$$

as described in Property 3.3. Prove

$$\phi_{hh} = \frac{\rho(h) - \widetilde{\boldsymbol{\rho}}_{h-1}' R_{h-1}^{-1} \boldsymbol{\rho}_h}{1 - \widetilde{\boldsymbol{\rho}}_{h-1}' R_{h-1}^{-1} \widetilde{\boldsymbol{\rho}}_{h-1}} = \alpha_{hh}.$$

In particular, this result proves Property 3.4.

Hint: Divide the prediction equations [see (3.63)] by $\gamma(0)$ and write the matrix equation in the partitioned form as

$$\begin{pmatrix} R_{h-1} & \widetilde{\boldsymbol{\rho}}_{h-1} \\ \widetilde{\boldsymbol{\rho}}_{h-1}' & \rho(0) \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \alpha_{hh} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\rho}_{h-1} \\ \rho(h) \end{pmatrix},$$

where the $h \times 1$ vector of coefficients $\boldsymbol{\alpha} = (\alpha_{h1}, \ldots, \alpha_{hh})'$ is partitioned as $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1', \alpha_{hh})'$.

3.14 Suppose we wish to find a prediction function $g(x)$ that minimizes

$$MSE = E[(y - g(x))^2],$$

where x and y are jointly distributed random variables with density function $f(x, y)$.

(a) Show that MSE is minimized by the choice

$$g(x) = E(y \mid x).$$

Hint:

$$MSE = \int \left[\int (y - g(x))^2 f(y|x) dy \right] f(x) dx.$$

(b) Apply the above result to the model

$$y = x^2 + z,$$

where x and z are independent zero-mean normal variables with variance one. Show that $MSE = 1$.

(c) Suppose we restrict our choices for the function $g(x)$ to linear functions of the form

$$g(x) = a + bx$$

and determine a and b to minimize MSE. Show that $a = 1$ and

$$b = \frac{E(xy)}{E(x^2)} = 0$$

and $MSE = 3$. What do you interpret this to mean?

3.15 For an AR(1) model, determine the general form of the m-step-ahead forecast x_{t+m}^t and show

$$E[(x_{t+m} - x_{t+m}^t)^2] = \sigma_w^2 \frac{1 - \phi^{2m}}{1 - \phi^2}.$$

3.16 Consider the ARMA(1,1) model discussed in Example 3.7, equation (3.27); that is, $x_t = .9x_{t-1} + .5w_{t-1} + w_t$. Show that truncated prediction as defined in (3.91) is equivalent to truncated prediction using the recursive formula (3.92).

3.17 Verify statement (3.87), that for a fixed sample size, the ARMA prediction errors are correlated.

Section 3.6

3.18 Fit an AR(2) model to the cardiovascular mortality series (`cmort`) discussed in Chapter 2, Example 2.2. using linear regression and using Yule–Walker.

(a) Compare the parameter estimates obtained by the two methods.
(b) Compare the estimated standard errors of the coefficients obtained by linear regression with their corresponding asymptotic approximations, as given in Property 3.10.

3.19 Suppose $x_1, \ldots, x_n$ are observations from an AR(1) process with $\mu = 0$.

(a) Show the backcasts can be written as $x_t^n = \phi^{1-t}x_1$, for $t \leq 1$.
(b) In turn, show, for $t \leq 1$, the backcasted errors are

$$\widehat{w}_t(\phi) = x_t^n - \phi x_{t-1}^n = \phi^{1-t}(1-\phi^2)x_1.$$

(c) Use the result of (b) to show $\sum_{t=-\infty}^{1} \widehat{w}_t^2(\phi) = (1-\phi^2)x_1^2$.
(d) Use the result of (c) to verify the unconditional sum of squares, $S(\phi)$, can be written as $\sum_{t=-\infty}^{n} \widehat{w}_t^2(\phi)$.
(e) Find x_t^{t-1} and r_t for $1 \leq t \leq n$, and show that

$$S(\phi) = \sum_{t=1}^{n} (x_t - x_t^{t-1})^2 \,/\, r_t.$$

3.20 Repeat the following numerical exercise three times. Generate $n = 500$ observations from the ARMA model given by

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

with $w_t \sim$ iid $\mathrm{N}(0,1)$. Plot the simulated data, compute the sample ACF and PACF of the simulated data, and fit an ARMA$(1,1)$ model to the data. What happened and how do you explain the results?

3.21 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.

3.22 Generate $n = 50$ observations from a Gaussian AR(1) model with $\phi = .99$ and $\sigma_w = 1$. Using an estimation technique of your choice, compare the approximate asymptotic distribution of your estimate (the one you would use for inference) with the results of a bootstrap experiment (use $B = 200$).

3.23 Using Example 3.31 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? *Hint:* Write the model as $w_t(\phi) = x_t - \phi x_{t-1}$; your solution should work out to be a non-recursive procedure.

3.24 Consider the stationary series generated by

$$x_t = \alpha + \phi x_{t-1} + w_t + \theta w_{t-1},$$

where $E(x_t) = \mu$, $|\theta| < 1, |\phi| < 1$ and the w_t are iid random variables with zero mean and variance σ_w^2.

(a) Determine the mean as a function of α for the above model. Find the autocovariance and ACF of the process x_t, and show that the process is weakly stationary. Is the process strictly stationary?
(b) Prove the limiting distribution as $n \to \infty$ of the sample mean,

$$\bar{x} = n^{-1} \sum_{t=1}^{n} x_t,$$

is normal, and find its limiting mean and variance in terms of α, ϕ, θ, and σ_w^2. (Note: This part uses results from Appendix A.)

3.25 A problem of interest in the analysis of geophysical time series involves a simple model for observed data containing a signal and a reflected version of the signal with unknown amplification factor a and unknown time delay δ. For example, the depth of an earthquake is proportional to the time delay δ for the P wave and its reflected form pP on a seismic record. Assume the signal, say s_t, is white and Gaussian with variance σ_s^2, and consider the generating model

$$x_t = s_t + a s_{t-\delta}.$$

(a) Prove the process x_t is stationary. If $|a| < 1$, show that

$$s_t = \sum_{j=0}^{\infty} (-a)^j x_{t-\delta j}$$

is a mean square convergent representation for the signal s_t, for $t = 1, \pm 1, \pm 2, \ldots$.
(b) If the time delay δ is assumed to be known, suggest an approximate computational method for estimating the parameters a and σ_s^2 using maximum likelihood and the Gauss–Newton method.

(c) If the time delay δ is an unknown integer, specify how we could estimate the parameters including δ. Generate a $n = 500$ point series with $a = .9$, $\sigma_w^2 = 1$ and $\delta = 5$. Estimate the integer time delay δ by searching over $\delta = 3, 4, \ldots, 7$.

3.26 *Forecasting with estimated parameters:* Let $x_1, x_2, \ldots, x_n$ be a sample of size n from a causal AR(1) process, $x_t = \phi x_{t-1} + w_t$. Let $\widehat{\phi}$ be the Yule–Walker estimator of ϕ.

(a) Show $\widehat{\phi} - \phi = O_p(n^{-1/2})$. See Appendix A for the definition of $O_p(\cdot)$.
(b) Let x_{n+1}^n be the one-step-ahead forecast of x_{n+1} given the data $x_1, \ldots, x_n$, based on the known parameter, ϕ, and let $\widehat{x}_{n+1}^n$ be the one-step-ahead forecast when the parameter is replaced by $\widehat{\phi}$. Show $x_{n+1}^n - \widehat{x}_{n+1}^n = O_p(n^{-1/2})$.

Section 3.7

3.27 Suppose

$$y_t = \beta_0 + \beta_1 t + \cdots + \beta_q t^q + x_t, \quad \beta_q \neq 0,$$

where x_t is stationary. First, show that $\nabla^k x_t$ is stationary for any $k = 1, 2, \ldots$, and then show that $\nabla^k y_t$ is not stationary for $k < q$, but is stationary for $k \geq q$.

3.28 Verify that the IMA(1,1) model given in (3.147) can be inverted and written as (3.148).

3.29 For the ARIMA$(1, 1, 0)$ model with drift, $(1 - \phi B)(1 - B)x_t = \delta + w_t$, let $y_t = (1 - B)x_t = \nabla x_t$.

(a) Noting that y_t is AR(1), show that, for $j \geq 1$,

$$y_{n+j}^n = \delta\left[1 + \phi + \cdots + \phi^{j-1}\right] + \phi^j\, y_n.$$

(b) Use part (a) to show that, for $m = 1, 2, \ldots$,

$$x_{n+m}^n = x_n + \frac{\delta}{1-\phi}\Big[m - \frac{\phi(1-\phi^m)}{(1-\phi)}\Big] + (x_n - x_{n-1})\frac{\phi(1-\phi^m)}{(1-\phi)}.$$

Hint: From (a), $x_{n+j}^n - x_{n+j-1}^n = \delta\frac{1-\phi^j}{1-\phi} + \phi^j(x_n - x_{n-1})$. Now sum both sides over j from 1 to m.

(c) Use (3.144) to find P_{n+m}^n by first showing that $\psi_0^* = 1$, $\psi_1^* = (1 + \phi)$, and $\psi_j^* - (1 + \phi)\psi_{j-1}^* + \phi\psi_{j-2}^* = 0$ for $j \geq 2$, in which case $\psi_j^* = \frac{1-\phi^{j+1}}{1-\phi}$, for $j \geq 1$. Note that, as in Example 3.36, equation (3.144) is exact here.

3.30 For the logarithm of the glacial varve data, say, x_t, presented in Example 3.32, use the first 100 observations and calculate the EWMA, $\widetilde{x}_{t+1}^t$, given in (3.150) for $t = 1, \ldots, 100$, using $\lambda = .25, .50$, and $.75$, and plot the EWMAs and the data superimposed on each other. Comment on the results.

Section 3.8

3.31 In Example 3.39, we presented the diagnostics for the MA(2) fit to the GNP growth rate series. Using that example as a guide, complete the diagnostics for the AR(1) fit.

3.32 Crude oil prices in dollars per barrel are in `oil`; see Appendix R for more details. Fit an ARIMA(p, d, q) model to the growth rate performing all necessary diagnostics. Comment.

3.33 Fit an ARIMA(p, d, q) model to the global temperature data `gtemp` performing all of the necessary diagnostics. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

3.34 One of the series collected along with particulates, temperature, and mortality described in Example 2.2 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.

Section 3.9

3.35 Consider the ARIMA model

$$x_t = w_t + \Theta w_{t-2}.$$

(a) Identify the model using the notation ARIMA$(p, d, q) \times (P, D, Q)_s$.
(b) Show that the series is invertible for $|\Theta| < 1$, and find the coefficients in the representation

$$w_t = \sum_{k=0}^{\infty} \pi_k x_{t-k}.$$

(c) Develop equations for the m-step ahead forecast, $\widetilde{x}_{n+m}$, and its variance based on the infinite past, $x_n, x_{n-1}, \ldots$.

3.36 Plot (or sketch) the ACF of the seasonal ARIMA$(0, 1) \times (1, 0)_{12}$ model with $\Phi = .8$ and $\theta = .5$.

3.37 Fit a seasonal ARIMA model of your choice to the unemployment data (`unemp`) displayed in Figure 3.21. Use the estimated model to forecast the next 12 months.

3.38 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.

3.39 Fit an appropriate seasonal ARIMA model to the log-transformed Johnson and Johnson earnings series (`jj`) of Example 1.1. Use the estimated model to forecast the next 4 quarters.

The following problems require supplemental material given in Appendix B.

3.40 Suppose $x_t = \sum_{j=1}^{p} \phi_j x_{t-j} + w_t$, where $\phi_p \neq 0$ and w_t is white noise such that w_t is uncorrelated with $\{x_k; k < t\}$. Use the Projection Theorem to show that, for $n > p$, the BLP of x_{n+1} on $\overline{\text{sp}}\{x_k, k \leq n\}$ is

$$\widehat{x}_{n+1} = \sum_{j=1}^{p} \phi_j x_{n+1-j}.$$

3.41 Use the Projection Theorem to derive the Innovations Algorithm, Property 3.6, equations (3.77)-(3.79). Then, use Theorem B.2 to derive the m-step-ahead forecast results given in (3.80) and (3.81).

3.42 Consider the series $x_t = w_t - w_{t-1}$, where w_t is a white noise process with mean zero and variance σ_w^2. Suppose we consider the problem of predicting x_{n+1}, based on only $x_1, \ldots, x_n$. Use the Projection Theorem to answer the questions below.

(a) Show the best linear predictor is

$$x_{n+1}^n = -\frac{1}{n+1} \sum_{k=1}^{n} k\, x_k.$$

(b) Prove the mean square error is

$$E(x_{n+1} - x_{n+1}^n)^2 = \frac{n+2}{n+1}\, \sigma_w^2.$$

3.43 Use Theorem B.2 and B.3 to verify (3.116).

3.44 Prove Theorem B.2.

3.45 Prove Property 3.2.

Appendix R
R Supplement

R.1 First Things First

If you do not already have 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. You can find helpful tutorials by following CRAN's link to *Contributed Documentation*. If you are new to R/S-PLUS, then *R for Beginners* by Emmanuel Paradis is a great introduction. There is also a lot of advice out there in cyberspace, but some of it will be outdated because R goes through many revisions.

Next, point your browser to `http://www.stat.pitt.edu/stoffer/tsa3/`, the website for the text, or one of its mirrors, download `tsa3.rda` and put it in a convenient place (e.g., the working directory of R).[1] This file contains the data sets and scripts that are used in the text. Then, start R and issue the command

```
load("tsa3.rda")
```

Once you have loaded `tsa3.rda`, all the files will stay in R as long as you save the workspace when you close R (details in §R.2). If you don't save the workspace, you will have to reload it. To see what is included in `tsa3.rda` type

```
ls()           # to get a listing of your objects, and
tsa3.version   # to check the version
```

Periodically check that your version matches the latest version number (year-month-day) on the website. Please note that `tsa3.rda` is subject to change.

You are free to use the data or to alter the scripts in any way you see fit. We only have two requests: reference the text if you use something from it, and contact us if you find any errors or have suggestions for improvement of the code.

[1] See §R.2, page 567, on how to get the current working directory and how to change it, or page 569 on how to read files from other directories.

R.1.1 Included Data Sets

The data sets included in `tsa3.rda`, listed by the chapter in which they are first presented, are as follows.

CHAPTER 1

- jj - Johnson & Johnson quarterly earnings per share, 84 quarters (21 years) measured from the first quarter of 1960 to the last quarter of 1980.
- EQ5 - Seismic trace of an earthquake [two phases or arrivals along the surface, the primary wave ($t = 1, \ldots, 1024$) and the shear wave ($t = 1025, \ldots, 2048$)] recorded at a seismic station.
- EXP6 - Seismic trace of an explosion (similar details as EQ5).
- gtemp - Global mean land-ocean temperature deviations (from 1951-1980 average), measured in degrees centigrade, for the years 1880-2009; data taken from `http://data.giss.nasa.gov/gistemp/graphs/`
- fmri1 - A data frame that consists of FMRI BOLD signals at eight locations (in columns 2-9, column 1 is time period), when a stimulus was applied for 32 seconds and then stopped for 32 seconds. The signal period is 64 seconds and the sampling rate was one observation every 2 seconds for 256 seconds ($n = 128$).
- soi - Southern Oscillation Index (SOI) for a period of 453 months ranging over the years 1950-1987.
- rec - Recruitment (number of new fish) associated with SOI.
- speech - A small .1 second (1000 points) sample of recorded speech for the phrase $aaa \cdots hhh$.
- nyse - Returns of the New York Stock Exchange (NYSE) from February 2, 1984 to December 31, 1991.
- soiltemp - A 64×36 matrix of surface soil temperatures.

CHAPTER 2

- oil - Crude oil, WTI spot price FOB (in dollars per barrel), weekly data from 2000 to mid-2010. For definitions and more details, see `http://tonto.eia.doe.gov/dnav/pet/pet_pri_spt_s1_w.htm`.
- gas - New York Harbor conventional regular gasoline weekly spot price FOB (in cents per gallon) over the same time period as `oil`.
- varve - Sedimentary deposits from one location in Massachusetts for 634 years, beginning nearly 12,000 years ago.
- cmort - Average weekly cardiovascular mortality in Los Angeles County; 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.
- tempr - Temperature series corresponding to cmort.
- part - Particulate series corresponding to cmort.
- so2 - Sulfur dioxide series corresponding to cmort.

CHAPTER 3

- prodn - Monthly Federal Reserve Board Production Index (1948-1978, $n = 372$ months).
- unemp - Unemployment series corresponding to prodn.
- ar1boot - Data used in Example 3.35 on page 137.

`gnp` - Quarterly U.S. GNP from 1947(1) to 2002(3), $n = 223$ observations.

`birth` - Monthly live births (adjusted) in thousands for the United States, 1948-1979.

Chapter 4

`sunspotz` - Biannual smoothed (12-month moving average) number of sunspots from June 1749 to December 1978; $n = 459$. The "z" on the end is to distinguish this series from the one included with R (called `sunspots`).

`salt` - Salt profiles taken over a spatial grid set out on an agricultural field, 64 rows at 17-ft spacing.

`saltemp` - Temperature profiles corresponding to `salt`.

Chapter 5

`arf` - 1000 simulated observations from an ARFIMA(1, 1, 0) model with $\phi = .75$ and $d = .4$.

`flu` - Monthly pneumonia and influenza deaths per 10,000 people in the United States for 11 years, 1968 to 1978.

`sales` - Sales (with `lead`, a leading indicator), 150 months; taken from Box & Jenkins (1970).

`lead` - See `sales`.

`econ5` - Data set containing quarterly U.S. unemployment, GNP, consumption, and government and private investment, from 1948-III to 1988-II.

Chapter 6

`ar1miss` - Data for Problem 6.14 on page 403.

`gtemp2` - Similar to `gtemp` but the data are based only on surface air temperature data obtained from meteorological stations.

`qinfl` - Quarterly inflation rate in the Consumer Price Index from 1953-I to 1980-II, $n = 110$ observations; from Newbold and Bos (1985).

`qintr` - Quarterly interest rate recorded for Treasury bills over the same period as `qinfl`.

`WBC` - Measurements made for 91 days on the three variables, log(white blood count) [`WBC`], log(platelet) [`PLT`] and hematocrit [`HCT`]; taken from Jones (1984).

`PLT` - See `WBC`.

`HCT` - See `WBC`.

Chapter 7

`beamd` - Infrasonic signals from a nuclear explosion. This is a data frame consisting of three columns (which are not time series objects) that are data from different channels. The series names are `sensor1, sensor2, sensor3`. See Example 7.2 on page 421 for more information.

`bnrf1ebv` - Nucleotide sequence of the BNRF1 gene of the Epstein-Barr virus (EBV): 1=A, 2=C, 3=G, 4=T. The data are used in §7.9.

`bnrf1hvs` - Nucleotide sequence of the BNRF1 gene of the herpes virus saimiri (HVS); same codes as for EBV.

`fmri` - Data (as a vector list) from an fMRI experiment in pain, listed by location and stimulus. The specfic locations of the brain where the signal was measured were [1] Cortex 1: Primary Somatosensory, Contralateral, [2] Cortex 2: Primary Somatosensory, Ipsilateral, [3] Cortex 3: Secondary Somatosensory, Contralateral, [4] Cortex 4: Secondary Somatosensory, Ipsilateral, [5] Caudate, [6] Thalamus 1: Contralateral, [7] Thalamus 2: Ipsilateral, [8] Cerebellum 1: Contralateral and [9] Cerebellum 2: Ipsilateral. The stimuli (and number of subjects in each condition) are [1] Awake-Brush (5 subjects), [2] Awake-Heat (4 subjects), [3] Awake-Shock (5 subjects), [4] Low-Brush (3 subjects), [5] Low-Heat (5 subjects), and [6] Low-Shock (4 subjects). Issue the command `summary(fmri)` for further details. As an example, `fmri$L1T6` (location 1, treatment 6) will show the data for the four subjects receiving the Low-Shock treatment at the Cortex 1 location; note that `fmri[[6]]` will display the same data. See Examples 7.7–7.9 for examples.

`climhyd` - Lake Shasta inflow data; see Example 7.1. This is a data frame with column names: `Temp, DewPt, CldCvr, WndSpd, Precip, Inflow`.

`eqexp` - This is a data frame of the earthquake and explosion seismic series used throughout the text. The matrix has 17 columns, the first eight are earthquakes, the second eight are explosions, and the last column is the Novaya Zemlya series. The column names are: `EQ1, EQ2,...,EQ8; EX1, EX2,...,EX8; NZ`.

R.1.2 Included Scripts

The following scripts are included in `tsa3.rda`. At the end of the description of each script, a text example that demonstrates the use of the script is given.

`lag.plot2(series1, series2, max.lag=0, corr=TRUE, smooth=TRUE)`

Produces a grid of scatterplots of one series versus another. If (x_t, y_t) is a vector time series, then `lag.plot2(x,y,m)` will generate a grid of scatterplots of x_{t-h} versus y_t for $h = 0, 1, ..., m$, along with the cross-correlation values (`corr=TRUE`) and a lowess fit (`smooth=TRUE`) assuming x_t is in `x` and y_t is in `y`. Note that the first series, x_t, is the one that gets lagged. If you just want the scatterplots and nothing else, then use `lag.plot2(x,y,m,corr=FALSE,smooth=FALSE)`. See Example 2.7 on page 64 for a demonstration.

`lag.plot1(series, max.lag=1, corr=TRUE, smooth=TRUE)`

Produces a grid of scatterplots of a series versus lagged values of the series. Similar to `lag.plot2`, the call `lag.plot1(x,m)` will generate a grid of scatterplots of x_{t-h} versus x_t for $h = 1, ..., m$, along with the autocorrelation values (`corr=TRUE`) and a lowess fit (`smooth=TRUE`). The defaults are the same as `lag.plot2`; if you don't want either the correlation values or the lowess fit, you can either use `lag.plot1(x,m,corr=FALSE,smooth=FALSE)` or R's `lag.plot`. See Example 2.7 on page 64 for a demonstration.

`acf2(series, max.lag=NULL)`

Produces a simultaneous plot (and a printout) of the sample ACF and PACF on the same scale. If `x` contains n observations, `acf2(x)` will print and plot the ACF and PACF of `x` to the default lag of $\sqrt{n} + 10$ (unless n is smaller than 50). The number of lags may be specified, e.g., `acf2(x, 33)`. See Example 3.17 on page 108.

```
sarima(series, p, d, q, P=0, D=0, Q=0, S=-1, details=TRUE,
       tol=sqrt(.Machine$double.eps), no.constant=FALSE)
```

Fits ARIMA models including diagnostics in a short command. If your time series is in `x` and you want to fit an ARIMA(p, d, q) model to the data, the basic call is `sarima(x,p,d,q)`. The results are the parameter estimates, standard errors, AIC, AICc, BIC (as defined in Chapter 2) and diagnostics. To fit a seasonal ARIMA model, the basic call is `sarima(x,p,d,q,P,D,Q,S)`. So, for example, `sarima(x,2,1,0)` will fit an ARIMA$(2, 1, 0)$ model to the series in `x`, and `sarima(x,2,1,0,0,1,1,12)` will fit a seasonal ARIMA$(2, 1, 0) \times (0, 1, 1)_{12}$ model to the series in `x`. If you want to look at the innovations (i.e., the residuals) from the fit, they're stored in `innov`.

There are three additional options that can be included in the call.

- `details` turns on/off the output from the nonlinear optimization routine, which is `optim`. The default is `TRUE`, use `details=FALSE` to turn off the output; e.g., `sarima(x,2,1,0,details=FALSE)`.
- `tol` controls the relative tolerance (reltol) used to assess convergence in `sarima` and `sarima.for`. The default is `tol=sqrt(.Machine$double.eps)`, the R default. For details, see the help file for `optim` under the `control` arguments. For example, `sarima(rec,2,0,0,tol=.0001)` will speed up the convergence. If there are many parameters to estimate (e.g., seasonal models), the analysis may take a long time using the default.
- `no.constant` can control whether or not `sarima` includes a constant in the model. In particular, with `sarima`, if there is no differencing ($d = 0$ and $D = 0$) you get the mean estimate. If there's differencing of order one (either $d = 1$ or $D = 1$, but not both), a constant term is included in the model; this may be overridden by setting this to `TRUE`; e.g., `sarima(x,1,1,0,no.constant=TRUE)`. In any other situation, no constant or mean term is included in the model. The idea is that if you difference more than once ($d + D > 1$), any drift is likely to be removed.

See Examples 3.38, 3.39, 3.40, 3.42, and 3.46 on pages 145–159 for demonstrations.

```
sarima.for(series, n.ahead, p, d, q, P=0, D=0, Q=0, S=-1,
           tol=sqrt(.Machine$double.eps), no.constant=FALSE)
```

Gives ARIMA forecasts. Similar to `sarima`, to forecast `n.ahead` time points from an ARIMA fit to the data in `x`, the form is `sarima.for(x, n.ahead, p, d, q)` or `sarima.for(x, n.ahead, p, d, q, P, D, Q, S)` for a seasonal model. For example, `sarima.for(x,5,1,0,1)` will forecast five time points ahead for an ARMA(1,1) fit to `x`. The output prints the forecasts and the standard errors of the forecasts, and supplies a graphic of the forecast with ± 2 prediction error

bounds. The options `tol` and `no.constant` are also available. See Example 3.46 on page 159.

`spec.arma(ar=0, ma=0, var.noise=1, n.freq=500, ...)`

Gives the ARMA spectrum (on a log scale), tests for causality, invertibility, and common zeros. The basic call is `spec.arma(ar, ma)` where `ar` and `ma` are vectors containing the model parameters. Use `log="no"` if you do not want the plot on a log scale. If the model is not causal or invertible an error message is given. If there are common zeros, a spectrum will be displayed and a warning will be given; e.g., `spec.arma(ar= .9, ma= -.9)` will yield a warning and the plot will be the spectrum of white noise. The variance of the noise can be changed with `var.noise`. Finally, the frequencies and the spectral density ordinates are returned invisibly, e.g., `spec.arma(ar=.9)$freq` and `spec.arma(ar=.9)$spec`, if you're interested in the actual values. See Example 4.6 on page 184.

`LagReg(input, output, L=c(3,3), M=20, threshold=0, inverse=FALSE)`

Performs lagged regression as discussed in Chapter 4, §4.10. For a bivariate series, `input` is the input series and `output` is the output series. The degree of smoothing for the spectral estimate is given by `L`; see `spans` in the help file for `spec.pgram`. The number of terms used in the lagged regression approximation is given by `M`, which must be even. The `threshold` value is the cut-off used to set small (in absolute value) regression coefficients equal to zero (it is easiest to run LagReg twice, once with the default threshold of zero, and then again after inspecting the resulting coefficients and the corresponding values of the CCF). Setting `inverse=TRUE` will fit a forward-lagged regression; the default is to run a backward-lagged regression. The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta. See Example 4.24 on page 244 for a demonstration.

`SigExtract(series, L=c(3,3), M=50, max.freq=.05)`

Performs signal extraction and optimal filtering as discussed in Chapter 4, §4.11. The basic function of the script, and the default setting, is to remove frequencies above 1/20 (and, in particular, the seasonal frequency of 1 cycle every 12 time points). The time series to be filtered is `series`, and its sampling frequency is set to unity ($\Delta = 1$). The values of `L` and `M` are the same as in `LagReg` and `max.freq` denotes the truncation frequency, which must be larger than 1/M. The filtered series is returned silently; e.g., `f.x = SigExtract(x)` will store the extracted signal in `f.x`. The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta. See Example 4.25 on page 249 for a demonstration.

`Kfilter0(n, y, A, mu0, Sigma0, Phi, cQ, cR)`

Returns the filtered values in Property 6.1 on page 326 for the state-space model, (6.1)–(6.2). In addition, the script returns the evaluation of the likelihood at the given parameter values and the innovation sequence. The inputs are `n`: number of

observations; `y`: data matrix; `A`: observation matrix (assumed constant); `mu0`: initial state mean; `Sigma0`: initial state variance-covariance matrix; `Phi`: state transition matrix; `cQ`: Cholesky decomposition of Q [`cQ=chol(Q)`]; `cR`: Cholesky decomposition of R [`cR=chol(R)`]. Note: The script requires only that Q or R may be reconstructed as `t(cQ)%*%cQ` or `t(cR)%*%cR`, which offers a little more flexibility than requiring Q or R to be positive definite. For demonstrations, see Example 6.6 on page 336, Example 6.8 on page 342, and Example 6.10 on page 350.

`Ksmooth0(n, y, A, mu0, Sigma0, Phi, cQ, cR)`

Returns both the filtered values in Property 6.1 on page 326 and the smoothed values in Property 6.2 on page 330 for the state-space model, (6.1)–(6.2). The inputs are the same as `Kfilter0`. For demonstrations, see Example 6.5 on page 331, and Example 6.10 on page 350.

`EM0(n, y, A, mu0, Sigma0, Phi, cQ, cR, max.iter=50, tol=.01)`

Estimation of the parameters in the model (6.1)–(6.2) via the EM algorithm. Most of the inputs are the same as for `Ksmooth0` and the script uses `Ksmooth0`. To control the number of iterations, use `max.iter` (set to 50 by default) and to control the relative tolerance for determining convergence, use `tol` (set to .01 by default). For a demonstration, see Example 6.8 on page 342.

`Kfilter1(n, y, A, mu0, Sigma0, Phi, Ups, Gam, cQ, cR, input)`

Returns the filtered values in Property 6.1 on page 326 for the state-space model, (6.3)–(6.4). In addition, the script returns the evaluation of the likelihood at the given parameter values and the innovation sequence. The inputs are `n`: number of observations; `y`: data matrix; `A`: observation matrix, an array with `dim=c(q,p,n)`; `mu0`: initial state mean; `Sigma0`: initial state variance-covariance matrix; `Phi`: state transition matrix; `Ups`: state input matrix; `Gam`: observation input matrix; `cQ`: Cholesky decomposition of Q; `cR`: Cholesky decomposition of R [the note in `Kfilter0` applies here]; `input`: matrix of inputs having the same row dimension as `y`. Set `Ups` or `Gam` or `input` to 0 (zero) if they are not used. For demonstrations, see Example 6.7 on page 338 and Example 6.9 on page 348.

`Ksmooth1(n, y, A, mu0, Sigma0, Phi, Ups, Gam, cQ, cR, input)`

Returns both the filtered values in Property 6.1 on page 326 and the smoothed values in Property 6.2 on page 330 for the state-space model, (6.3)–(6.4). The inputs are the same as `Kfilter1`. See Example 6.7 on page 338 and Example 6.9 on page 348.

`EM1(n, y, A, mu0, Sigma0, Phi, Ups, Gam, cQ, cR, input, max.iter=50, tol=.01)`

Estimation of the parameters in the model (6.3)–(6.4) via the EM algorithm. Most of the inputs are the same as for `Ksmooth1` and the script uses `Ksmooth1`. To control the number of iterations, use `max.iter` (set to 50 by default) and to control the relative tolerance for determining convergence, use `tol` (set to .01 by default). For a demonstration, see Example 6.12 on page 357.

`Kfilter2(n, y, A, mu0, Sigma0, Phi, Ups, Gam, Theta, cQ, cR, S, input)`

Returns the filtered values in Property 6.5 on page 354 for the state-space model, (6.97)–(6.99). In addition, the script returns the evaluation of the likelihood at the given parameter values and the innovation sequence. The inputs are similar to `Kfilter1`, except that the noise covariance matrix, `S` must be included. For demonstrations, see Example 6.11 on page 356 and Example 6.13 on page 361.

`Ksmooth2(n, y, A, mu0, Sigma0, Phi, Ups, Gam, Theta, cQ, cR, S, input)`

This is the smoother companion to `Kfilter2`.

`SVfilter(n, y, phi0, phi1, sQ, alpha, sR0, mu1, sR1)`

Performs the special case switching filter for the stochastic volatility model, (6.173), (6.175)–(6.176). The state parameters are `phi0, phi1, sQ` $[\phi_0, \phi_1, \sigma_w]$, and `alpha, sR0, mu1, sR1` $[\alpha, \sigma_0, \mu_1, \sigma_1]$ are observation equation parameters as presented in Section 6.9. See Example 6.18 page 380 and Example 6.19 page 383.

```
mvspec(x, spans = NULL, kernel = NULL, taper = 0, pad = 0, fast = TRUE,
        demean = TRUE, detrend = FALSE, plot = FALSE,
        na.action = na.fail, ...)
```

This is `spec.pgram` with a few changes in the defaults and written so you can extract the estimate of the multivariate spectral matrix as `fxx`. For example, if `x` contains a p-variate time series (i.e., the p columns of `x` are time series), and you issue the command `spec = mvspec(x, spans=3)` say, then `spec$fxx` is an array with dimensions `dim=c(p,p,nfreq)`, where `nfreq` is the number of frequencies used. If you print `spec$fxx`, you will see `nfreq` $p \times p$ spectral matrix estimates. See Example 7.12 on page 461 for a demonstration.

`FDR(pvals, qlevel=0.001)`

Computes the basic false discovery rate given a vector of p-values; see Example 7.4 on page 427 for a demonstration.

`stoch.reg(data, cols.full, cols.red, alpha, L, M, plot.which)`

Performs frequency domain stochastic regression discussed in §7.3. Enter the entire data matrix (`data`), and then the corresponding columns of input series in the full model (`cols.full`) and in the reduced model (`cols.red`; use `NULL` if there are no inputs under the reduced model). The response variable should be the *last* column of the data matrix, and this need not be specified among the inputs. Other arguments are `alpha` (test size), `L` (smoothing), `M` (number of points in the discretization of the integral) and `plot.which = coh` or `F.stat`, to plot either the squared-coherencies or the F-statistics. The coefficients of the impulse response function are returned and plotted. The script is based on code that was contributed by Professor Doug Wiens, Department of Mathematical and Statistical Sciences, University of Alberta. See Example 7.1 on page 417 for a demonstration.

R.2 Getting Started

If you are experienced with R/S-PLUS you can skip this section, and perhaps the rest of this appendix. Otherwise, it is essential to have R up and running before you start this tutorial. The best way to use the rest of this appendix is to start up R and enter the example code as it is presented. Also, you can use the results and help files to get a better understanding of how R works (or doesn't work). The character # is used for comments.

The convention throughout the text is that R code is in `typewriter font` with a small line number in the left margin. Get comfortable, then start her up and try some simple tasks.

```
1 2+2       # addition
  [1] 5
2 5*5 + 2   # multiplication and addition
  [1] 27
3 5/5 - 3   # division and subtraction
  [1] -2
4 log(exp(pi)) # log, exponential, pi
  [1] 3.141593
5 sin(pi/2) # sinusoids
  [1] 1
6 exp(1)^(-2)  # power
  [1] 0.1353353
7 sqrt(8)   # square root
  [1] 2.828427
8 1:5       # sequences
  [1] 1 2 3 4 5
9 seq(1, 10, by=2)  # sequences
  [1] 1 3 5 7 9
10 rep(2,3)  # repeat 2 three times
  [1] 2 2 2
```

Next, we'll use *assignment* to make some *objects*:

```
1 x <- 1 + 2  # put 1 + 2 in object x
2 x = 1 + 2   # same as above with fewer keystrokes
3 1 + 2 -> x  # same
4 x           # view object x
  [1] 3
5 (y = 9*3)   # put 9 times 3 in y and view the result
  [1] 27
6 (z = rnorm(5,0,1))  # put 5 standard normals into z and print z
  [1]  0.96607946  1.98135811 -0.06064527  0.31028473  0.02046853
```

To list your objects, remove objects, get help, find out which directory is current (or to change it) or to quit, use the following commands:

```
ls()       # list all objects
  [1] "dummy" "mydata" "x" "y" "z"
ls(pattern = "my")  # list every object that contains "my"
  [1] "dummy" "mydata"
rm(dummy)  # remove object "dummy"
help.start()  # html help and documentation (use it)
help(exp)  # specific help  (?exp is the same)
getwd()    # get working directory
setwd("/TimeSeries/")  # change working directory to TimeSeries
q()        # end the session (keep reading)
```

When you quit, R will prompt you to save an image of your current workspace. Answering "yes" will save all the work you have done so far, and load it up when you next start R. Our suggestion is to answer "yes" even though you will also be loading irrelevant past analyses every time you start R. Keep in mind that you can remove items via `rm()`. If you do not save the workspace, you will have to reload `tsa3.rda` as described in §R.1.

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]   # the third element
  [1] 3
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
dim(mydata)     # no dimensions
 NULL
mydata = as.matrix(mydata)  # make it a matrix
dim(mydata)     # now it has dimensions
  [1] 5 1
```

It is worth pointing out R's *recycling rule* for doing arithmetic. The rule is extremely helpful for shortening code, but it can also lead to mistakes if you are not careful. Here are some examples.

```
x = c(1, 2, 3, 4);  y = c(2, 4, 6, 8);  z = c(10, 20)
x*y  # it's 1*2, 2*4, 3*6, 4*8
  [1]  2  8 18 32
x/z  # it's 1/10, 2/20, 3/10, 4/20
  [1] 0.1 0.1 0.3 0.2
```

```
4 x+z  # guess
  [1] 11 22 13 24
```

If you have an external data set, you can use `scan` or `read.table` to input the data. For example, suppose you have an ASCII (text) data file called `dummy.dat` in a directory called `TimeSeries` in your root directory, and the file looks like this:

```
1 2 3 2 1
9 0 2 1 0
```

```
1 dummy = scan("dummy.dat")  # if TimeSeries is the working directory
2 (dummy = scan("/TimeSeries/dummy.dat"))  # if not, do this
  Read 10 items
  [1] 1 2 3 2 1 9 0 2 1 0
3 (dummy = read.table("/TimeSeries/dummy.dat"))
  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. In this case, if you want to list (or use) the second variate, `V2`, you would use

```
4 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 in line 4 above.

```
5 colnames(dummy) = c("Dog", "Cat", "Rat", "Pig", "Man")
6 attach(dummy)
7 Cat
  [1] 2 0
```

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. You may also include a *header* in the data file to avoid using line 5 above; type `?read.table` for further information.

It can't hurt to learn a little about programming in R because you will see some of it along the way. Consider a simple program that we will call `crazy` to produce a graph of a sequence of sample means of increasing sample sizes from a Cauchy distribution with location parameter zero. The code is:

```
1 crazy <- function(num) {
2   x <- rep(NA, num)
3   for (n in 1:num) x[n] <- mean(rcauchy(n))
4   plot(x, type="l", xlab="sample size", ylab="sample mean")
5   }
```

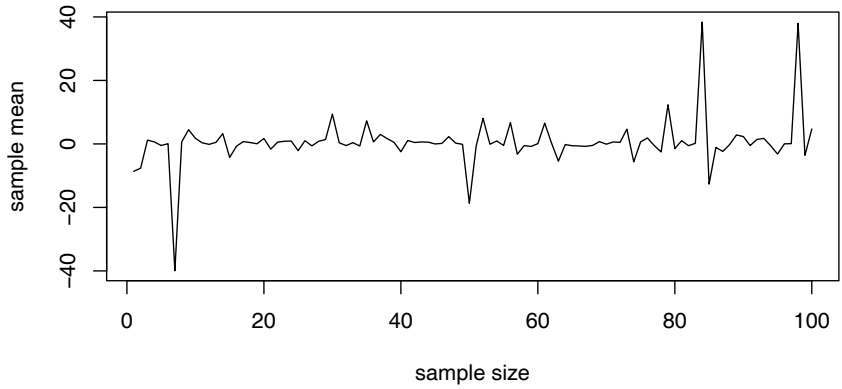

Fig. R.1. Crazy example.

The first line creates the function `crazy` and gives it one argument, `num`, that is the sample size that will end the sequence. Line 2 makes a vector, `x`, of `num` missing values `NA`, that will be used to store the sample means. Line 3 generates `n` random Cauchy variates [`rcauchy(n)`], finds the mean of those values, and puts the result into `x[n]`, the n-th value of `x`. The process is repeated in a "do loop" `num` times so that `x[1]` is the sample mean from a sample of size one, `x[2]` is the sample mean from a sample of size two, and so on, until finally, `x[num]` is the sample mean from a sample of size `num`. After the do loop is complete, the fourth line generates a graphic (see Figure R.1). The fifth line closes the function. To use `crazy` with a limit sample size of 100, for example, type

```
6 crazy(100)
```

and you will get a graphic that looks like Figure R.1

You may want to use one of the R packages. In this case you have to first download the package and then install it. For example,

```
1 install.packages(c("wavethresh", "tseries"))
```

will download and install the packages `wavethresh` that we use in Chapter 4 and `tseries` that we use in Chapter 5; you will be asked to choose the closest mirror to you. To use a package, you have to load it at each start up of R, for example:

```
2 library(wavethresh)  # load the wavethresh package
```

A good way to get help for a package is to use html help

```
3 help.start()
```

and follow the *Packages* link.

Finally, a word of caution: `TRUE` and `FALSE` are reserved words, whereas `T` and `F` are initially set to these. Get in the habit of using the words rather than the letters `T` or `F` because you may get into trouble if you do something like `F = qf(p=.01, df1=3, df2=9)`, so that `F` is no longer `FALSE`, but a quantile of the specified F-distribution.

R.3 Time Series Primer

In this section, we give a brief introduction on using R for time series. We assume that `tsa3.rda` has been loaded. 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
```

Now make it a time series:

```
(mydata = as.ts(mydata))
Time Series:
Start = 1
End = 5
Frequency = 1
[1] 1 2 3 2 1
```

Make it an annual time series that starts in 1950:

```
(mydata = ts(mydata, start=1950))
Time Series:
Start = 1950
End = 1954
Frequency = 1
[1] 1 2 3 2 1
```

Now make it a quarterly time series that starts in 1950-III:

```
(mydata = ts(mydata, start=c(1950,3), frequency=4))
     Qtr1 Qtr2 Qtr3 Qtr4
1950              1    2
1951    3    2    1
time(mydata)  # view the sampled times
          Qtr1    Qtr2    Qtr3    Qtr4
1950                   1950.50 1950.75
1951  1951.00 1951.25 1951.50
```

To use part of a time series object, use `window()`:

```
(x = window(mydata, start=c(1951,1), end=c(1951,3)))
     Qtr1 Qtr2 Qtr3
1951    3    2    1
```

Next, we'll look at lagging and differencing. 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 (we display the time series attributes in a single row of the output to save space).

```
cbind(x, lag(x), lag(x,-1))
```

```
Time Series:  Start = 0  End = 6  Frequency = 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 difference a series, $\nabla x_t = x_t - x_{t-1}$, use

```
diff(x)
```

but note that

```
diff(x, 2)
```

is *not* second order differencing, it is $x_t - x_{t-2}$. For second order differencing, that is, $\nabla^2 x_t$, do this:

```
diff(diff(x))
```

and so on for higher order differencing.

For graphing time series, there a few standard plotting mechanisms that we use repeatedly. If `x` is a time series, then `plot(x)` will produce a time plot. If `x` is not a time series object, then `plot.ts(x)` will coerce it into a time plot as will `ts.plot(x)`. There are differences, which we explore in the following. It would be a good idea to skim the graphical parameters help file (`?par`) while you are here.[2] See Figure R.2 for the resulting graphic.

```
x = -5:5      # x is NOT a time series object
y = 5*cos(x)  # neither is y
op = par(mfrow=c(3,2))  # multifigure setup: 3 rows, 2 cols
plot(x, main="plot(x)")
plot(x, y, main="plot(x,y)")
plot.ts(x, main="plot.ts(x)")
plot.ts(x, y, main="plot.ts(x,y)")
ts.plot(x, main="ts.plot(x)")
ts.plot(ts(x), ts(y), col=1:2, main="ts.plot(x,y)")
par(op)  # reset the graphics parameters  [see footnote]
```

[2] In the plot example, the parameter set up uses `op = par(...)` and ends with `par(op)`; these lines are used to reset the graphic parameters to their previous settings. Please make a note of this because we do not display these commands ad nauseam in the text.

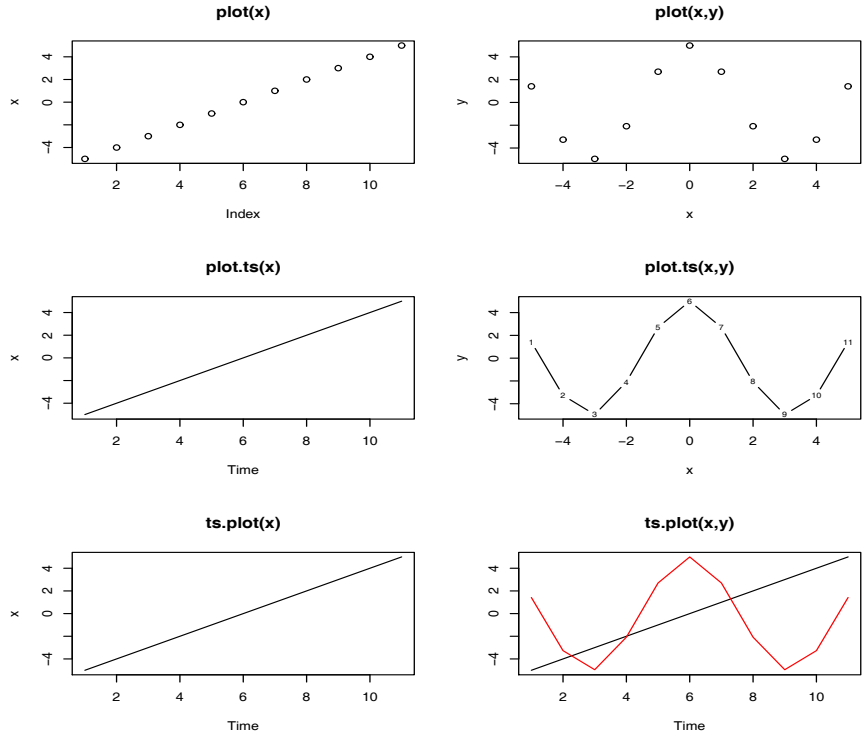

Fig. R.2. Demonstration of different R graphic tools for plotting time series.

We will also make use of regression via `lm()`. First, suppose we want to fit a simple linear regression, $y = \alpha + \beta x + \epsilon$. In R, the formula is written as `y~x`:

```
set.seed(1999)       # so you can reproduce the result
x = rnorm(10,0,1)
y = x + rnorm(10,0,1)
summary(fit <- lm(y~x))
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.8851 -0.3867  0.1325  0.3896  0.6561

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2576     0.1892   1.362   0.2104
x             0.4577     0.2016   2.270   0.0529

Residual standard error: 0.58 on 8 degrees of freedom
Multiple R-squared: 0.3918,     Adjusted R-squared: 0.3157
F-statistic: 5.153 on 1 and 8 DF,  p-value: 0.05289
```

```
plot(x, y)   # draw a scatterplot of the data (not shown)
abline(fit)  # add the fitted line to the plot (not shown)
```

All sorts of information can be extracted from the `lm` object, which we called `fit`. For example,

```
resid(fit)      # will display the residuals (not shown)
fitted(fit)     # will display the fitted values (not shown)
lm(y ~ 0 + x)   # will exclude the intercept  (not shown)
```

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 "tie" the series together using `ts.intersect`. If you do not tie the series together, they will not be aligned properly. Please read the warning *Using time series* in the `lm()` help file [`help(lm)`]. Here is an example regressing Chapter 2 data, weekly cardiovascular mortality (`cmort`) on particulate pollution (`part`) at the present value and lagged four weeks (`part4`). First, we create a data frame called `ded` that consists of the three series:

```
ded = ts.intersect(cmort, part, part4=lag(part,-4), dframe=TRUE)
```

Now the series are all aligned and the regression will work.

```
fit = lm(mort~part+part4, data=ded, na.action=NULL)
summary(fit)
```

```
Call: lm(formula=mort~part+part4,data=ded,na.action=NULL)
Residuals:
     Min       1Q   Median       3Q      Max
-22.7429  -5.3677  -0.4136   5.2694  37.8539

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. There is a package called `dynlm` that makes it easy to fit lagged regressions. The basic advantage of `dynlm` is that it avoids having to make a data frame; that is, line 1 would be avoided.

In Problem 2.1, 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(rep(1:4, 21))  # make (Q)uarter factors
reg = lm(log(jj)~0 + trend + Q, na.action=NULL)  # no intercept
model.matrix(reg)  # view the model 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
.     .    .  .  .  .
.     .    .  .  .  .
83  10.50  0  0  1  0
84  10.75  0  0  0  1
```

```
summary(reg)  # view the results (not shown)
```

The workhorse for ARIMA simulations is `arima.sim`. Here are some examples; no output is shown here so you're on your own.

```
x = arima.sim(list(order=c(1,0,0),ar=.9),n=100)+50  # AR(1) w/mean 50
x = arima.sim(list(order=c(2,0,0),ar=c(1,-.9)),n=100)  # AR(2)
x = arima.sim(list(order=c(1,1,1),ar=.9,ma=-.5),n=200) # ARIMA(1,1,1)
```

Next, we'll discuss ARIMA estimation. This gets a bit tricky because R is not useR friendly when it comes to fitting ARIMA models. Much of the story is spelled out in the "R Issues" page of the website for the text. In Chapter 3, we use the scripts `acf2`, `sarima`, and `sarima.for` that are included with `tsa3.Rda`. But we will also show you how to use the scripts included with R.

First, we'll fit an ARMA(1,1) model to some simulated data (with diagnostics and forecasting):

```
set.seed(666)
x = 50 + arima.sim(list(order=c(1,0,1), ar=.9, ma=-.5), n=200)
acf(x); pacf(x) # display sample ACF and PACF ... or ...
acf2(x)         # use our script (no output shown)
(x.fit = arima(x, order = c(1, 0, 1)))  # fit the model
```

```
Call:  arima(x = x, order = c(1, 0, 1))
Coefficients:
         ar1     ma1  intercept
      0.8340  -0.432    49.8960
s.e.  0.0645   0.111     0.2452
sigma^2 estimated as 1.070: log likelihood = -290.79, aic = 589.58
```

Note that the reported `intercept` estimate is an estimate of the mean and *not* the constant. That is, the fitted model is

$$\widehat{x}_t - 49.896 = .834(x_{t-1} - 49.896) + \widehat{w}_t$$

where $\widehat{\sigma}_w^2 = 1.070$. Diagnostics can be accomplished as follows,

```
tsdiag(x.fit, gof.lag=20) # ?tsdiag for details (don't use this!!)
```

but the Ljung-Box-Pierce test is not correct because it does not take into account the fact that the residuals are from a fitted model. If the analysis is repeated using the `sarima` script, a partial output would look like the following (`sarima` will also display the correct diagnostics as a graphic; e.g., see Figure 3.17 on page 151):

```
sarima(x, 1, 0, 1)
```

```
Coefficients:
          ar1     ma1    xmean
       0.8340  -0.432  49.8960
s.e.   0.0645   0.111   0.2452
sigma^2 estimated as 1.070: log likelihood = -290.79, aic = 589.58
$AIC [1] 1.097494   $AICc  [1] 1.108519   $BIC  [1] 0.1469684
```

Finally, to obtain and plot the forecasts, you can use the following R code:

```
x.fore = predict(x.fit, n.ahead=10)
U = x.fore$pred + 2*x.fore$se   # x.fore$pred holds predicted values
L = x.fore$pred - 2*x.fore$se   # x.fore$se holds stnd errors
miny = min(x,L);  maxy = max(x,U)
ts.plot(x, x.fore$pred, col=1:2, ylim=c(miny, maxy))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
```

Using the script `sarima.for`, you can accomplish the same task in one line.

```
sarima.for(x, 10, 1, 0, 1)
```

Example 3.46 on page 159 uses this script.

We close this appendix with a quick spectral analysis. This material is covered in detail in Chapter 4, so we will not discuss this example in much detail here. We will simulate an AR(2) and then estimate the spectrum via nonparametric and parametric methods. No graphics are shown, but we have confidence that you are proficient enough in R to display them yourself.

```
x = arima.sim(list(order=c(2,0,0), ar=c(1,-.9)), n=2^8) # some data
(u = polyroot(c(1,-1,.9)))  # x is AR(2) w/complex roots
  [1] 0.5555556+0.8958064i 0.5555556-0.8958064i
Arg(u[1])/(2*pi)  # dominant frequency around .16
  [1] 0.1616497
par(mfcol=c(4,1))
plot.ts(x)
spec.pgram(x,spans=c(3,3),log="no") # nonparametric spectral estimate
spec.ar(x, log="no")                # parametric spectral estimate
spec.arma(ar=c(1,-.9), log="no")    # true spectral density
```

The script `spec.arma` is included in `tsa3.rda`. Also, see `spectrum` as an alternative to `spec.pgram`. Finally, note that R tapers and logs by default, so if you simply want the periodogram of a series, the command is `spec.pgram(x, taper=0, fast=FALSE, detrend=FALSE, log="no")`. If you just asked for `spec.pgram(x)`, you would not get the RAW periodogram because the data are detrended, possibly padded, and tapered, even though the title of the resulting graphic would say *Raw Periodogram*. An easier way to get a raw periodogram is:

```
per = abs(fft(x))^2/length(x)
```

This final example points out the importance of knowing the defaults for the R scripts you use.