Time Series Analysis

Using the R Statistical Package

tsaEZ®

http://www.stat.pitt.edu/stoffer/tsa4/

PUBLISHED BY FREE DOG PUBLISHING

LIVE FREE OR BARK

VERSION: 2017.12.31

Preface

These notes can be used for an introductory time series course where the prerequisites are an understanding of linear regression and some basic probability skills (primarily expectation). It also assumes general math skills at the high school level (trigonometry, complex numbers, polynomials, calculus, and so on).

- Various topics depend heavily on techniques from nonlinear regression. Consequently, the reader should have a solid knowledge of linear regression analysis, including multiple regression and weighted least squares. Some of this material is reviewed briefly in Chapters 2 and 3.
- A calculus based course on probability is essential. Readers should be familiar with most of the content of basic probability facts: http://www.stat.pitt.edu/stoffer/tsa4/intro_prob.pdf.
- For readers who are a bit rusty on high school math skills, the WikiBook: http://en.wikibooks.org/wiki/Subject:K-12_mathematics may be useful. In particular, we mention the book covering calculus: http://en.wikibooks.org/wiki/Calculus.
  We occasionally use matrix notation. For readers lacking this skill, see the high school page on matrices: https://en.wikibooks.org/wiki/High_School_Mathematics_Extensions/Matrices.
  For Chapter 4, this primer on complex numbers: http://tutorial.math.lamar.edu/pdf/Complex/ComplexNumbers.pdf may be helpful.

All of the numerical examples were done using the R statistical package, and the code is typically listed at the end of an example. Appendix R has information regarding the use of R and how to obtain the most recent version of the package used throughout, `astsa`. In addition, there are several exercises that may help first time users get more comfortable with the software.

The references are not listed here, but may be found in Shumway & Stoffer (2017) or earlier versions. Internal links are `dark red`, external links are `magenta`, R code is in `blue`, output is `purple` and comments are `# green`.

Contents

Chapter 1

Time Series Characteristics

1.1 Introduction

The analysis of data observed at different time points leads to unique problems. The obvious dependence introduced by the sampling data over time restricts the applicability of many conventional statistical methods that require random samples. The analysis of such data is commonly referred to as *time series analysis*.

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

Historically, time series methods were applied to problems in the physical and environmental sciences. This fact accounts for the basic engineering flavor permeating the language of time series analysis. In our view, the first step in any time series investigation always involves careful scrutiny of the recorded data plotted over time. 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* (Chapter 3) and the *frequency domain approach* (Chapter 4).

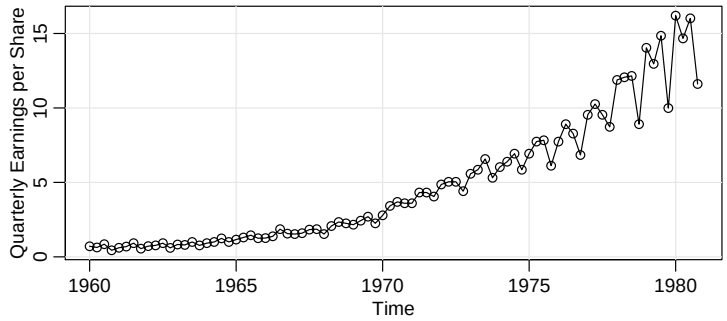


Fig. 1.1. *Johnson & Johnson quarterly earnings per share, 1960-I to 1980-IV.*

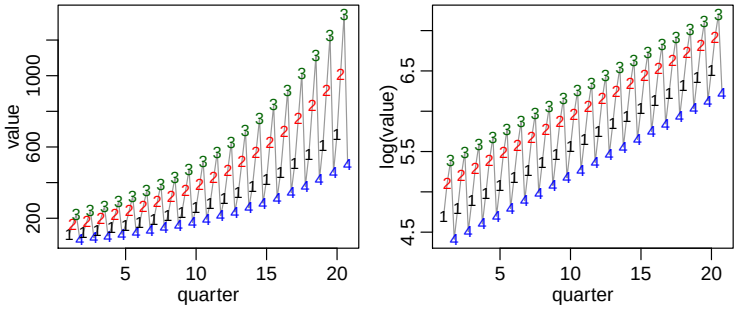


Fig. 1.2. Left: *Initial deposits of $100, $150, $200, then $75, in quarter 1, 2, 3, then 4, over 20 years, with an annual growth rate of 10%;* $x_t = (1+.10)x_{t-4}$. Right: *Logs of the quarterly values;* $\log(x_t) = \log(1+.10) + \log(x_{t-4})$.

1.2 Some Time Series Data

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

Example 1.1 Johnson & Johnson Quarterly Earnings

Figure 1.1 shows quarterly earnings per share for the U.S. company Johnson & Johnson. There are 84 quarters (21 years) measured from the first quarter of 1960 to the last quarter of 1980. Modeling such series begins by observing the primary patterns in the time history. In this case, note the increasing underlying trend and variability, and a somewhat regular oscillation superimposed on the trend that seems to repeat over quarters. Methods for analyzing data such as these are explored in Chapter 2 (see Problem 2.1) using regression techniques. Also, compare Figure 1.1 with Figure 1.2.

To use package `astsa`, and then plot the data for this example using R, type the following (try plotting the logged data yourself).

```
library(astsa)       # ** SEE FOOTNOTE
tsplot(jj, type="o", ylab="Quarterly Earnings per Share")
tsplot(log(jj))      # not shown
```

** We assume that the R package `astsa` has been downloaded and installed. See Appendix R (Section R.2.1) for further details.

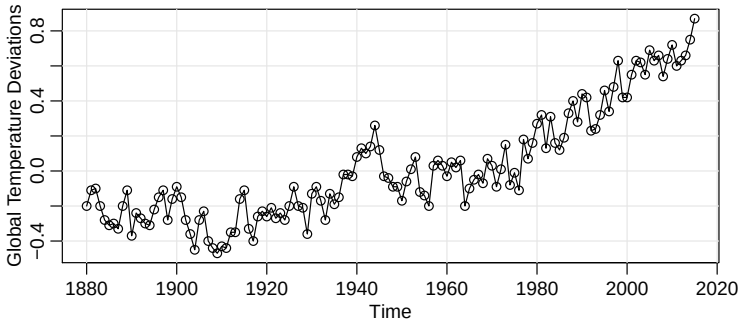


Fig. 1.3. *Yearly average global temperature deviations (1880–2015) in °C.*

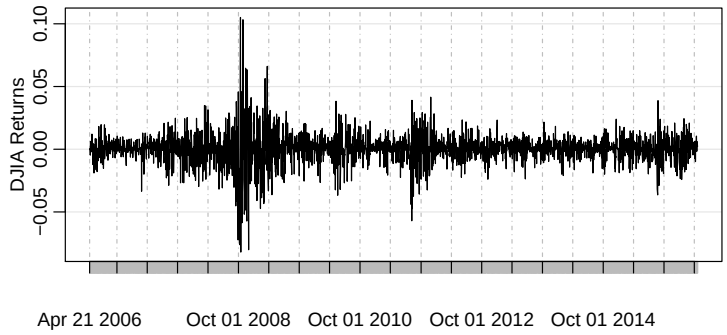


Fig. 1.4. *The daily returns of the Dow Jones Industrial Average (DJIA) from April 20, 2006 to April 20, 2016.*

Example 1.2 Global Warming

Consider the global temperature series record shown in Figure 1.3. The data are the global mean land–ocean temperature index from 1880 to 2015, with the base period 1951-1980. The values are deviations (°C) from the 1951-1980 average, updated from Hansen et al. (2006). The upward trend in the series during the latter part of the twentieth century has been used as an argument for the climate change hypothesis. Note that the trend is not linear, with periods of leveling off and then sharp upward trends. The question of interest is whether the overall trend is natural or caused by some human-induced interface. The R code for this example is:

```
tsplot(globtemp, type="o", ylab="Global Temperature Deviations")
```

Example 1.3 Dow Jones Industrial Average

As an example of financial time series data, Figure 1.4 shows the daily returns (or percent change) of the Dow Jones Industrial Average (DJIA) from 2006 to 2016. It is easy to spot the financial crisis of 2008 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 exhibits 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 have been developed to handle these problems; see Chapter 5. We then used the fact that if x_t is the actual value

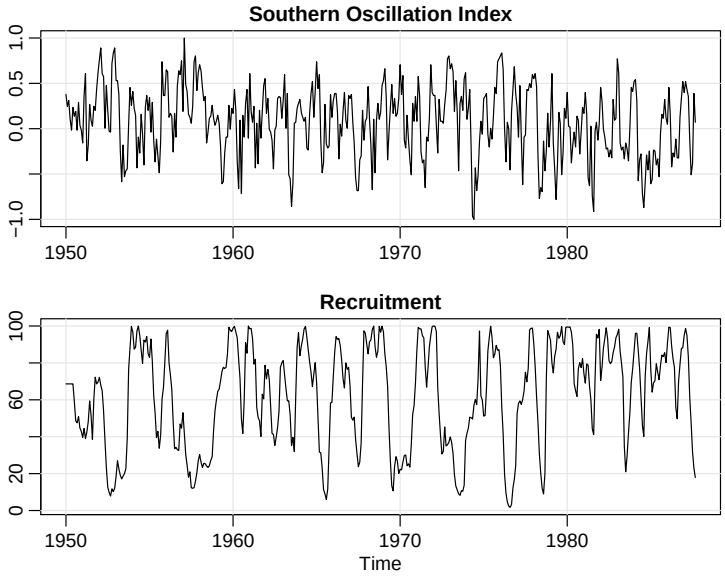


Fig. 1.5. *Monthly SOI and Recruitment (estimated new fish), 1950-1987.*

Looking at both curves, it is very difficult to conclude whether there exists and underlying relationship between them. This can be made very easy later on, when we define the crosscovariance between the two time series.

of the DJIA and $r_t = (x_t - x_{t-1})/x_{t-1}$ is the return, then $1 + r_t = x_t/x_{t-1}$ and $\log(1 + r_t) = \log(x_t/x_{t-1}) = \log(x_t) - \log(x_{t-1}) \approx r_t$.[1] The data set is provided in `astsa` but `xts` must be loaded.

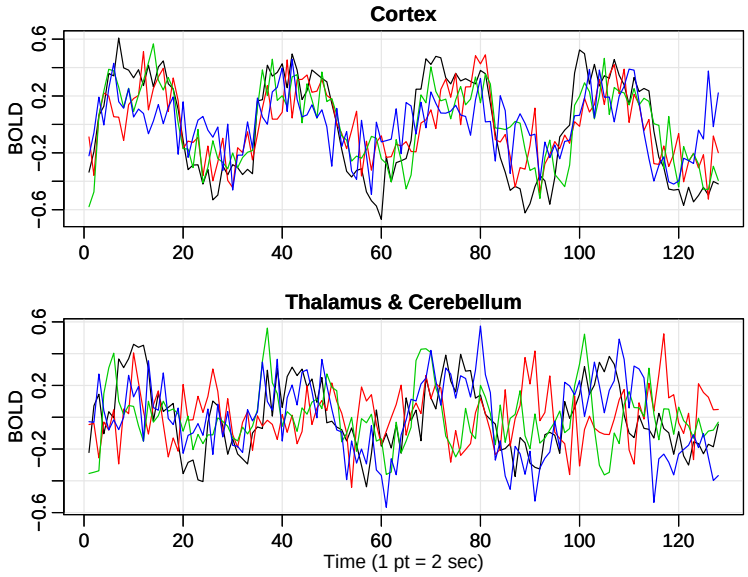
```
library(xts)
djiar = diff(log(djia$Close))[-1]          # approximate returns
tsplot(djiar, main="DJIA Returns", xlab='', margins=.5)
```

Example 1.4 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 (an index of the number of new fish). Both series are for a period of 453 months ranging over the years 1950–1987. 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 for various global extreme weather events. The series show two basic oscillations types, an obvious annual cycle (hot in the summer, cold in the winter), and a slower frequency that seems to repeat about every 4 years. The study of the kinds of cycles and their strengths is the subject of Chapter 4. The two series are also related; it is easy to imagine the fish population is dependent on the ocean temperature. The following R code will reproduce Figure 1.5:

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

[1] $\log(1+p) = p - \frac{p^2}{2} + \frac{p^3}{3} - \cdots$ for $-1 < p \leq 1$. If p is near zero, the higher-order terms in the expansion are negligible.



Each curve corrrsponds to the activity of different neurons.

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

Example 1.5 fMRI Imaging

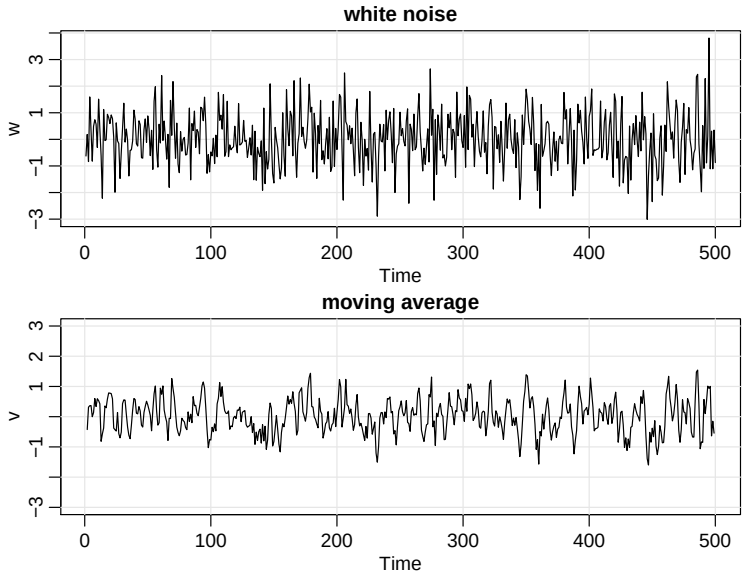
Often, time series are observed under varying experimental conditions or treatment configurations. Such a set of series is shown in Figure 1.6, where data are collected from various locations in the brain via functional magnetic resonance imaging (fMRI). In this example, a stimulus was applied for 32 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 ($n = 128$). The series 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. Use the following R commands to plot the data:

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

1.3 Time Series Models

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

The fundamental visual characteristic distinguishing the different series shown in Example 1.1 – Example 1.5 is their differing degrees of smoothness. A parsimonious explanation for this smoothness is that adjacent points in time are



Observe that the moving average is smoother than the white noise. This is natural, since the moving average is obtained by AVERAGING out successive samples of the white noise series. The more samples we average, the smoother series the series that will result. Try it.

Fig. 1.7. *Gaussian white noise series (top) and three-point moving average of the Gaussian white noise series (bottom).*

correlated, so the value of the series at time t, say, x_t, depends in some way on the past values $x_{t-1}, x_{t-2}, \ldots$. This idea expresses a fundamental way in which we might think about generating realistic looking time series.

Example 1.6 White Noise

A simple kind of generated series might be a collection of uncorrelated random variables, w_t, with mean 0 and finite variance σ_w^2. The time series generated from uncorrelated variables is used as a model for noise in engineering applications where it is called *white noise*; we shall sometimes denote this process as $w_t \sim wn(0, \sigma_w^2)$. The designation white originates from the analogy with white light (details in Chapter 4).

We often require stronger conditions and need the noise to be Gaussian white noise, wherein the w_t are independent and identically distributed (iid) normal random variables, with mean 0 and variance σ_w^2; or more succinctly, $w_t \sim \text{iid } \mathrm{N}(0, \sigma_w^2)$. Figure 1.7 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 resemblance to portions of the DJIA returns in Figure 1.4.

Although both cases require zero mean and constant variance, the difference is that generically, the term white noise means the time series is uncorrelated. Gaussian white noise implies normality (which implies independence).

If the stochastic behavior of all time series could be explained in terms of the white noise model, classical statistical methods would suffice. Two ways of introducing serial correlation and more smoothness into time series models are given in Example 1.7 and Example 1.8.

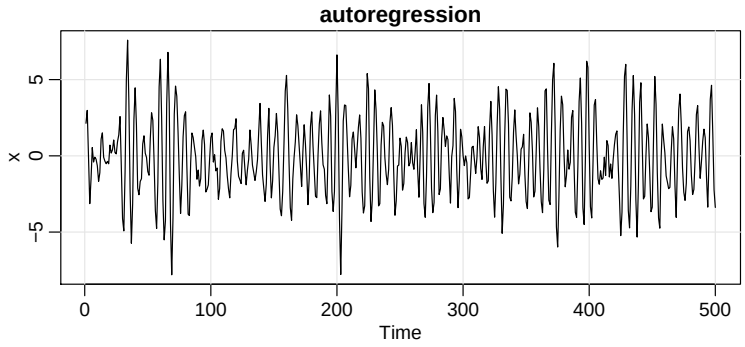


Fig. 1.8. *Autoregressive series generated from model* (1.2).

Example 1.7 Moving Averages and Filtering

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 1.6 by an average of its current value and its immediate neighbors in the past and future. That is, let

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

which leads to the series shown in the lower panel of Figure 1.7. This series is much smoother than the white noise series, and it is apparent that averaging removes some of the high frequency behavior of the noise. We begin to notice a similarity to some of the non-cyclic fMRI series in Figure 1.6.

To reproduce Figure 1.7 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`.

```
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))
tsplot(w, main="white noise")
tsplot(v, ylim=c(-3,3), main="moving average")
```

The SOI and Recruitment series in Figure 1.5, as well as some of the fMRI series in Figure 1.6, differ from the moving average series because they are dominated by an oscillatory behavior. A number of methods exist for generating series with this quasi-periodic behavior; we illustrate a popular one based on the autoregressive model considered in Chapter 3.

Example 1.8 Autoregressions

Suppose we consider the white noise series w_t of Example 1.6 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 $t = 1, 2, \ldots, 500$. The resulting output series is shown in Figure 1.8. 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 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

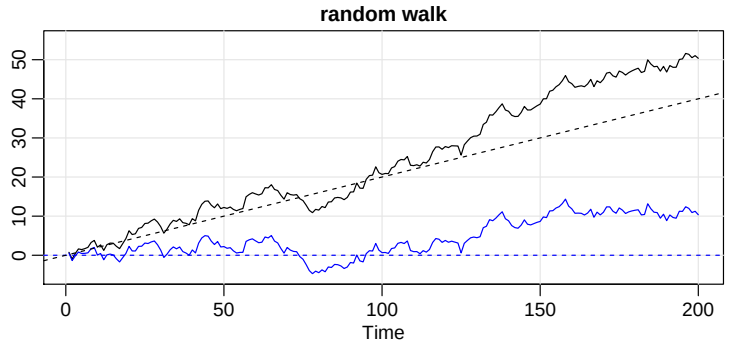


Fig. 1.9. *Random walk, $\sigma_w = 1$, with drift $\delta = .2$ (upper jagged line), without drift, $\delta = 0$ (lower jagged line), and dashed lines showing the drifts.*

the succeeding values by substituting into (1.2). That is, given $w_1, w_2, \ldots, w_{500}$, and x_0, x_{-1}, we start with $x_1 = x_0 - .9x_{-1} + w_1$, then recursively compute $x_2 = x_1 - .9x_0 + w_2$, then $x_3 = x_2 - .9x_1 + w_3$, and so on. We note the approximate periodic behavior of the series, which is similar to that displayed by the SOI and Recruitment in Figure 1.5 and some fMRI series in Figure 1.6. 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`). The initial conditions are set equal to zero, so we let the filter run an extra 50 values to avoid startup problems.

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

Example 1.9 Random Walk with Drift

A model for analyzing trend such as seen in the global temperature data in Figure 1.3, 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 $\delta = 0$, the model is called simply a random walk because the value of the time series at time t is the value of the series at time $t-1$ plus a completely random movement determined by w_t. Note that we may rewrite (1.3) as a cumulative sum of white noise variates. That is,

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

for $t = 1, 2, \ldots$; either use induction, or plug (1.4) into (1.3) to verify this statement. Figure 1.9 shows 200 observations generated from the model with $\delta = 0$ and .2, and with standard normal nose. For comparison, we also superimposed the straight lines δt on the graph.

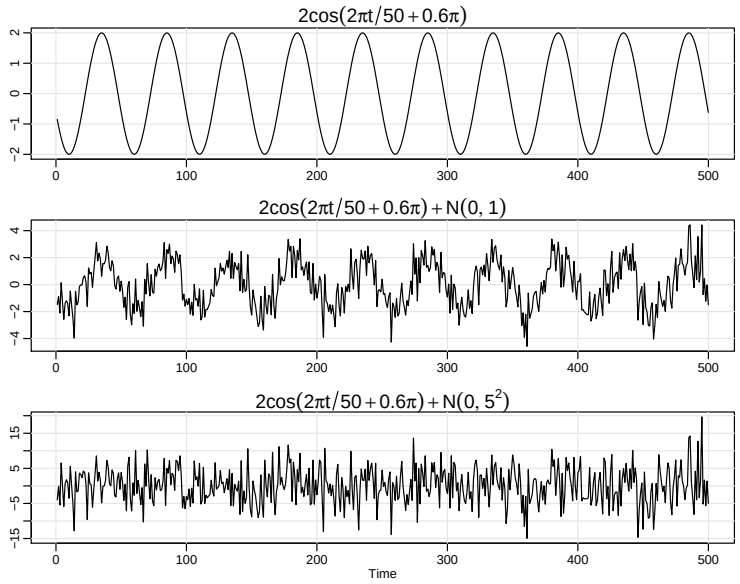


Fig. 1.10. *Cosine wave with period 50 points (top panel) compared with the cosine wave contaminated with additive white Gaussian noise, $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel); see (1.5).*

To reproduce Figure 1.9 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);  x = cumsum(w)    # two commands in one line
wd = w +.2;       xd = cumsum(wd)
tsplot(xd, ylim=c(-5,55), main="random walk", ylab='')
abline(a=0, b=.2, lty=2)           # drift
lines(x, col=4)
abline(h=0, col=4, lty=2)
```

Example 1.10 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

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

for $t = 1, 2, \ldots, 500$, where the first term is regarded as the signal, shown in the upper panel of Figure 1.10. We note that a sinusoidal waveform can be written as

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

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

An additive noise term was taken to be white noise with $\sigma_w = 1$ (middle panel) and $\sigma_w = 5$ (bottom panel), drawn from a normal distribution. Adding the two together obscures the signal, as shown in the lower panels of Figure 1.10. Of course, the degree to which the signal is obscured depends on

the amplitude of the signal relative to the size of σ_w. The ratio of the amplitude of the signal to σ_w (or some function of the ratio) is sometimes called the *signal-to-noise ratio (SNR)*; the larger the SNR, the easier it is to detect the signal. Note that the signal is easily discernible in the middle panel, whereas the signal is obscured in the bottom panel. Typically, we will not observe the signal but the signal obscured by noise.

To reproduce Figure 1.10 in R, use the following commands:

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

1.4 Measures of Dependence

We now discuss various measures that describe the general behavior of a process as it evolves over time. A rather simple descriptive measure is the mean function.

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

$$\mu_{xt} = E(x_t) \tag{1.7}$$

provided it exists, where E denotes the usual expected value operator.[2] *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 1.11 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 1.7 reflects this, as the series clearly fluctuates around a mean value of zero. Smoothing the series as in Example 1.7 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 1.12 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

[2] Expectation is discussed in the third chapter of the basic probability facts pdf mentioned in the preface. For continuous-valued finite variance processes, the mean is $\mu_t = E(x_t) = \int_{-\infty}^{\infty} x f_t(x)\, dx$ and the variance is $\sigma_t^2 = E(x_t - \mu_t)^2 = \int_{-\infty}^{\infty} (x - \mu_t)^2 f_t(x)\, dx$, where f_t is the density of x_t. 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 $f_t(x) = \frac{1}{\sigma_t\sqrt{2\pi}} \exp\{-\frac{1}{2\sigma_t^2}(x - \mu_t)^2\}$ for $x \in \mathbb{R}$.

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

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 1.9.

Example 1.13 Mean Function of Signal Plus Noise

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

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

and the mean function is just the cosine wave.

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

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

$$\gamma_x(s,t) = \mathrm{cov}(x_s, x_t) = E[(x_s - \mu_s)(x_t - \mu_t)], \tag{1.8}$$

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

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

$$\gamma_x(t,t) = E[(x_t - \mu_t)^2] = \mathrm{var}(x_t). \tag{1.9}$$

Note that the autocovariance function depends, IN GENERAL, on both indices, s and t.

If two jointly Gaussian variables are uncorrrelated they are also independent. This is not true in general. Uncorelatedness does not imply independence. The opposite is always true. Independence always impies uncorrelatedness.

Example 1.14 Autocovariance of White Noise

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

$$\gamma_w(s,t) = \mathrm{cov}(w_s, w_t) = \begin{cases} \sigma_w^2 & s = t, \\ 0 & s \neq t. \end{cases} \tag{1.10}$$

A realization of white noise with $\sigma_w^2 = 1$ is shown in the top panel of Figure 1.7.

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

Property 1.1 *If the random variables*

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

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

$$\operatorname{cov}(U, V) = \sum_{j=1}^{m} \sum_{k=1}^{r} a_j b_k \operatorname{cov}(X_j, Y_k). \tag{1.11}$$

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

An easy way to remember (1.11) is to treat it like multiplication:

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

Example 1.15 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 1.7. In this case,

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

When $s = t$ we have

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

When $s = t + 1$,

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

using (1.10). 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

$$\gamma_v(s,t) = \begin{cases} \frac{3}{9}\sigma_w^2 & s = t, \\ \frac{2}{9}\sigma_w^2 & |s-t| = 1, \\ \frac{1}{9}\sigma_w^2 & |s-t| = 2, \\ 0 & |s-t| > 2. \end{cases} \tag{1.12}$$

Example 1.15 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 1.16 Autocovariance of a Random Walk

For the random walk model, $x_t = \sum_{j=1}^{t} w_j$, we have

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

because the w_t are uncorrelated random variables. For example, with $s = 1$ and $t = 2$,

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

Note that, as opposed to the previous examples, the autocovariance function of a random walk depends on the particular time values s and t, and not on the time separation or lag. Also, notice that the variance of the random walk, $\text{var}(x_t) = \gamma_x(t,t) = t\,\sigma_w^2$, increases without bound as time t increases. The effect of this variance increase can be seen in Figure 1.9 where the processes start to move away from their mean functions $\delta\,t$ (note that $\delta = 0$ and .2 in that example).

Once more note that the authocavariance here depends on the values of both s and t, and selects the minimum of the two. It does NOT depend, for example, on their difference.

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*

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}}. \tag{1.13}$$

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.[3] 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.

Due to the Cauchy-Schwartz Inequality.

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*

$$\gamma_{xy}(s,t) = \text{cov}(x_s, y_t) = E[(x_s - \mu_{xs})(y_t - \mu_{yt})]. \tag{1.14}$$

The cross-covariance function can be scaled to live in $[-1, 1]$:

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

$$\rho_{xy}(s,t) = \frac{\gamma_{xy}(s,t)}{\sqrt{\gamma_x(s,s)\gamma_y(t,t)}}. \tag{1.15}$$

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

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.

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

$$\{x_{t_1}, x_{t_2}, \ldots, x_{t_k}\} \quad \text{and} \quad \{x_{t_1+h}, x_{t_2+h}, \ldots, x_{t_k+h}\},$$

are identical, for all $k = 1, 2, \ldots$, *all time points* $t_1, t_2, \ldots, t_k$, *and all time shifts* $h = 0, \pm 1, \pm 2, \ldots$.

It is difficult to assess strict stationarity from data. 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.

Definition 1.7 *A* **weakly stationary** *time series is a finite variance process where*

(i) the mean value function, μ_t, *defined in* (1.7) *is constant and does not depend on time* t, *and*
(ii) the autocovariance function, $\gamma(s,t)$, *defined in* (1.8) *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.*

Stationarity requires regularity in the mean and autocorrelation functions so that these quantities (at least) may be estimated by averaging. It should be clear that a strictly stationary, finite variance, time series is also stationary. The converse is not true in general. One important case where stationarity implies strict stationarity is if the time series is Gaussian [meaning all finite collections of the series are Gaussian].

Example 1.17 A Random Walk is Not Stationary

A random walk is not stationary because its autocovariance function, $\gamma(s,t) = \min\{s,t\}\sigma_w^2$, depends on time; see Example 1.16 and Problem 1.7. Also, the random walk with drift violates both conditions of Definition 1.7 because, as shown in Example 1.12, the mean function, $\mu_{xt} = \delta t$, is also a function of time t.

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

$$\mu_t = \mu. \tag{1.16}$$

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

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

Observe that the autocovariance function is SYMMETRIC, and its maximum value occurs at lag=0. This is natural, since the correlation of each sample with ITSELF can only be larger with the correlation between different samples.

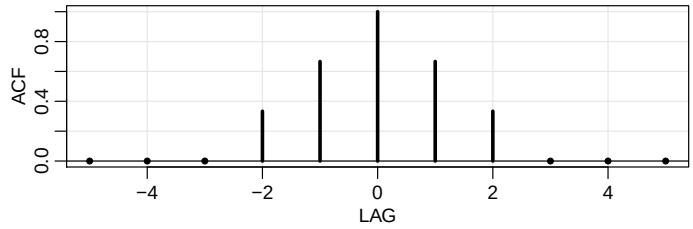


Fig. 1.11. *Autocovariance function of a three-point moving average.*

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 $\gamma(h,0)$.

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

$$\gamma(h) = \operatorname{cov}(x_{t+h}, x_t) = E[(x_{t+h}-\mu)(x_t-\mu)]. \tag{1.17}$$

Definition 1.9 *The* **autocorrelation function (ACF) of a stationary time series** *will be written using* (1.13) *as*

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}. \tag{1.18}$$

The Cauchy–Schwarz inequality implies that $-1 \le \rho(h) \le 1$ 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 1.18 Stationarity of White Noise

The mean and autocovariance functions of the white noise series discussed in Example 1.6 and Example 1.14 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.

Example 1.19 Stationarity of a Moving Average

The three-point moving average process of Example 1.7 is stationary because, from Example 1.11 and Example 1.15, 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.

Note that the ACF is given by

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

Figure 1.11 shows a plot of the autocorrelation as a function of lag h. Note that the autocorrelation function is symmetric about lag zero.

Example 1.20 Trend Stationarity
For example, if $x_t = \alpha + \beta t + y_t$, where y_t is stationary, then the mean function is $\mu_{x,t} = E(x_t) = \alpha + \beta t + \mu_y$, which is not independent of time. Therefore, the process is not stationary. The autocovariance function, however, is independent of time, because

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

Thus, the model may be considered as having stationary behavior around a linear trend; this behavior is sometimes called *trend stationarity*. An example of such a process is the price of chicken series displayed in Figure 2.1.

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

$$\gamma(0) = E[(x_t - \mu)^2] = \operatorname{var}(x_t). \tag{1.19}$$

Also, the Cauchy–Schwarz inequality implies $|\gamma(h)| \le \gamma(0)$. Another useful property is that the autocovariance function of a stationary series is symmetric around the origin,

$$\gamma(h) = \gamma(-h) \tag{1.20}$$

The autocovariance funcion is SYMMETRIC around the origin

for all h. This property follows because

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

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

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

Property 1.2 Wold Decomposition *Any stationary time series, x_t, can be written as linear combination (filter) of white noise terms; that is,*

$$x_t = \mu + \sum_{j=0}^{\infty} \psi_j w_{t-j}, \qquad \sum_{j=0}^{\infty} \psi_j^2 < \infty, \tag{1.21}$$

with $\psi_0 = 1$. We call these **linear processes**.

The models we will encounter in Chapter 3 are linear processes. For the linear process, we may show that the mean function is $E(x_t) = \mu$, and the autocovariance function is given by

$$\gamma(h) = \sigma_w^2 \sum_{j=0}^{\infty} \psi_{j+h}\psi_j \tag{1.22}$$

for $h \geq 0$; recall that $\gamma(-h) = \gamma(h)$. To see (1.22), note that

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

Observe that after the products, only terms corresponding to the same time instants survive (the rest result in zero, since the noise is white). For all the surviving terms, the multiplying coefficients differ by h.

Note that, for the moving average in Example 1.7, we have $\psi_0 = \psi_{-1} = \psi_1 = 1/3$ and the result in Example 1.19 comes out immediately. The autoregressive series in Example 1.8 can also be put in this form as we will see in Chapter 3.

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* **jointly stationary** *if they are each stationary, and the cross-covariance function*

$$\gamma_{xy}(h) = \operatorname{cov}(x_{t+h}, y_t) = E[(x_{t+h} - \mu_x)(y_t - \mu_y)] \tag{1.23}$$

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*

$$\rho_{xy}(h) = \frac{\gamma_{xy}(h)}{\sqrt{\gamma_x(0)\gamma_y(0)}}. \tag{1.24}$$

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

$$\rho_{xy}(h) = \rho_{yx}(-h), \tag{1.25}$$

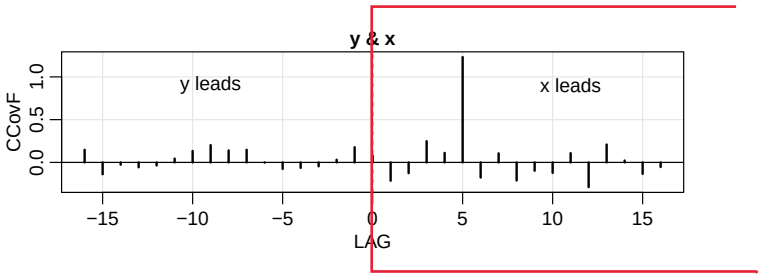
The crosscovariance function is NOT symmnetric

which can be shown by manipulations similar to those used to show (1.20).

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,

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



In the right are x leads. For the case of the figure, this means that x at time t-5 is positively correlated with the the random variable y at time t, and this is true for all time instants t.

Fig. 1.12. *Demonstration of the results of Example 1.22 when $\ell = 5$. The title indicates which series is leading.*

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,

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

because only one term is nonzero (recall Property 1.1). Similarly, $\gamma_{xy}(0) = 0, \gamma_{xy}(-1) = -\sigma_w^2$. We obtain, using (1.24),

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

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

Example 1.22 Prediction Using Cross-Correlation

Consider the problem of determining possible leading or lagging relations between two series x_t and y_t. If for some integer ℓ, the model

If l >0, we say that x leads y. Indeed, the values of y depends of the value of x l time instants EARLIER. This also explains physicaly, why the crosscovariance is NOT symmetric

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

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 that the noise w_t is uncorrelated with the x_t series, the cross-covariance function can be computed as

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

Since the largest value of $|\gamma_x(h-\ell)|$ is $\gamma_x(0)$, i.e., when $h = \ell$, the cross-covariance function will look like the autocovariance of the input series x_t, and it will have 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. Below is the R code of an example with $\ell = 5$ and $\hat{\gamma}_{yx}(h)$ is shown in Figure 1.12.

```
x = rnorm(100);  y = lag(x,-5) + rnorm(100)
ccf(y, x, ylab='CCovF', type='covariance')
```

1.6 Estimation of Correlation

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

Accordingly, if a time series is stationary, the mean function (1.16) $\mu_t = \mu$ is constant so that we can estimate it by the sample mean,

$$\bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t. \tag{1.26}$$

The estimate is unbiased, $E(\bar{x}) = \mu$, and its standard error is the square root of $\mathrm{var}(\bar{x})$, which can be computed using first principles (recall Property 1.1), and is given by

$$\mathrm{var}(\bar{x}) = \frac{1}{n}\sum_{h=-n}^{n}\Big(1 - \frac{|h|}{n}\Big)\gamma_x(h)\,. \tag{1.27}$$

If the process is white noise, (1.27) 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 1.14).

The theoretical autocorrelation function, (1.18), is estimated by the sample ACF as follows.

Definition 1.12 *The* **sample autocorrelation function (ACF)** *is defined as*

$$\widehat{\rho}(h) = \frac{\widehat{\gamma}(h)}{\widehat{\gamma}(0)} = \frac{\sum_{t=1}^{n-h}(x_{t+h} - \bar{x})(x_t - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \tag{1.28}$$

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

The sum in the numerator of (1.28) runs over a restricted range because x_{t+h} is not available for $t + h > n$. Note that we are in fact estimating the autocovariance function by

$$\widehat{\gamma}(h) = n^{-1}\sum_{t=1}^{n-h}(x_{t+h} - \bar{x})(x_t - \bar{x}), \tag{1.29}$$

Note that we divide by n, that guarantees that the absolute value ot the aucorrelation function remains less that one.

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

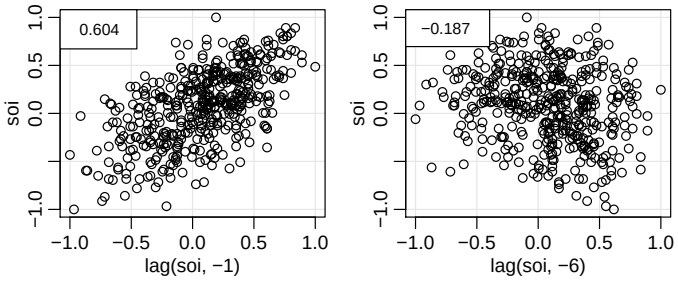
$$\{(x_{t+h}, x_t);\ t = 1, \ldots, n-h\}\,. \tag{1.30}$$

There are n-h possible pairs of samples in a time series observation/realization.

This assures that the sample ACF will behave as a true autocorrelation function, and for example, will not give values bigger than one in absolute value.

Example 1.23 Sample ACF and Scatterplots

Estimating autocorrelation is similar to estimating of correlation in the classical case, but now we have the $n - h$ pairs of observations displayed in (1.30). Figure 1.13 shows an example using the SOI series where $\widehat{\rho}(1) = .604$ and $\widehat{\rho}(6) = -.187$. The following code was used for Figure 1.13.



The scatterplot to the left reveals a structure. Pairs of data points that are one month away, tend to be scattered around a line. This means that, on average, when one value is, say positive, we expect the other one to be positive, too. The same for negative values. The scatter plot to the right hardly reveals a structure. If one of the values is, say, positive, the other one may be positive or negative. That is why the correlation is low.

Fig. 1.13. *Display for Example 1.23. For the SOI series, we have a scatterplot of pairs of values one month apart (left) and six months apart (right). The estimated correlation is displayed in the box.*

```
(r = round(acf(soi, 6, plot=FALSE)$acf[-1], 3))  # sample acf values
  [1]  0.604  0.374  0.214  0.050 -0.107 -0.187
par(mfrow=c(1,2), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
plot(lag(soi,-1), soi);  legend('topleft', legend=r[1])
plot(lag(soi,-6), soi);  legend('topleft', legend=r[6])
```

Remark 1.1 It is important to note that this approach to estimating correlation ***makes sense only if the data are stationary***. If the data were not stationary, each point in the graph could be an observation from a different correlation structure.

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

Property 1.3 Large-Sample Distribution of the ACF

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

Based on this, we can assess whether a series is white or not.

Based on Property 1.3, we obtain a rough method for assessing whether a series is white noise by determining how many values of $\widehat{\rho}(h)$ are outside the interval $\pm 2/\sqrt{n}$ (two standard errors); for white noise, approximately 95% of the sample ACFs should be within these limits.[4] The bounds do not hold in general and can be ignored if the interest is other than assessing whiteness. 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. Afterwards the plotted ACF of the residuals behave as stated.

Example 1.24 A Simulated Time Series

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

$$y_t = 5 + x_t - .5x_{t-1}\,. \tag{1.31}$$

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

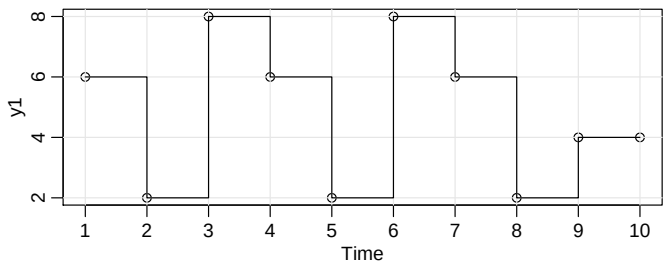


Fig. 1.14. *Realization of* (1.31)*,* $n = 10$*.*

We consider two cases, one with a small sample size ($n = 10$; see Figure 1.14) and another with a moderate sample size ($n = 100$).

```
set.seed(101011)
x1 = 2*(2*rbinom(11, 1, .5) - 1)    # simulated sequence of coin tosses
x2 = 2*(2*rbinom(101, 1, .5) - 1)
y1 = 5 + filter(x1, sides=1, filter=c(1,-.5))[-1]
y2 = 5 + filter(x2, sides=1, filter=c(1,-.5))[-1]
tsplot(y1, type='s'); points(y1)    # y2 not shown
acf(y1, lag.max=4, plot=FALSE)  # 1/sqrt(10) =.32
 Autocorrelations of series 'y1', by lag
      0      1      2      3      4
  1.000 -0.352 -0.316  0.510 -0.245
acf(y2, lag.max=4, plot=FALSE)  # 1/sqrt(100) =.1
 Autocorrelations of series 'y2', by lag
      0      1      2      3      4
  1.000 -0.496  0.067  0.087  0.063
```

The theoretical ACF can be obtained from the model (1.31) using first principles so that

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

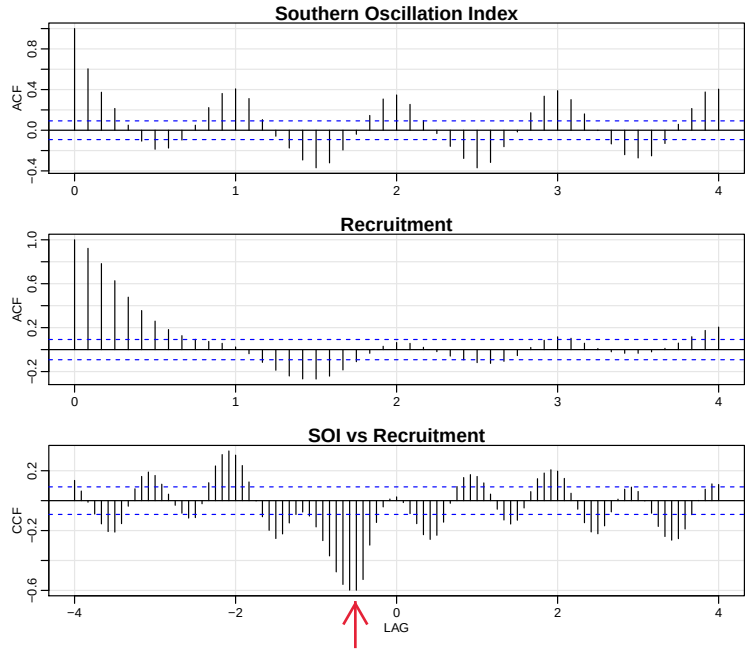
and $\rho_y(h) = 0$ for $|h| > 1$ (do Problem 1.19 now). It is interesting to compare the theoretical ACF with sample ACFs for the realization where $n = 10$ and the other realization where $n = 100$; note the increased variability in the smaller size sample.

Definition 1.13 *The estimators for the cross-covariance function,* $\gamma_{xy}(h)$*, as given in* (1.23) *and the cross-correlation,* $\rho_{xy}(h)$*, in* (1.24) *are given, respectively, by the* **sample cross-covariance function**

$$\widehat{\gamma}_{xy}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x})(y_t - \bar{y}), \tag{1.32}$$

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

$$\widehat{\rho}_{xy}(h) = \frac{\widehat{\gamma}_{xy}(h)}{\sqrt{\widehat{\gamma}_x(0)\widehat{\gamma}_y(0)}}. \tag{1.33}$$



SOI leads by sixth months but in a negative direction. When SOI increases Recruitment decreases and vice versa.

Fig. 1.15. *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).*

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.

Property 1.4 Large-Sample Distribution of Cross-Correlation

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

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 1.4 (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 1.15 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; see

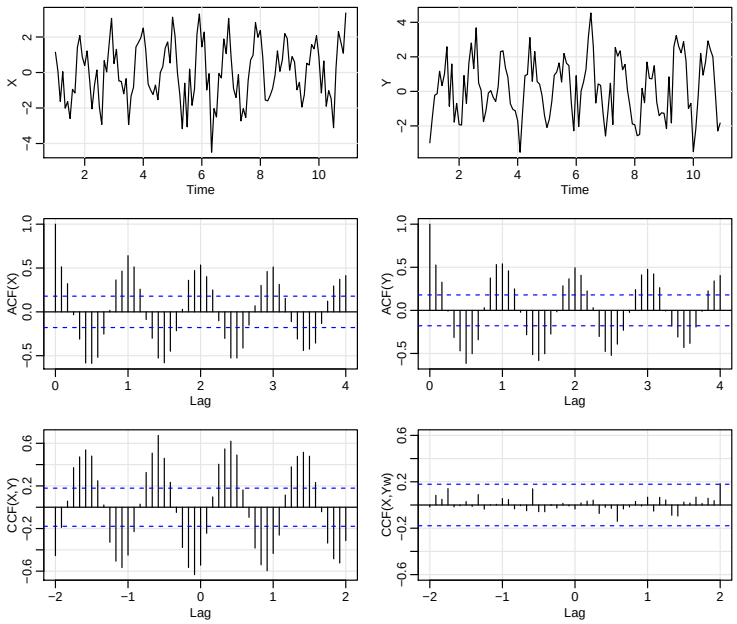


Fig. 1.16. *Display for Example 1.26*

Example 1.26. The cross-correlation function peaks at $h = -6$, showing that the SOI measured at time $t - 6$ months is associated with the Recruitment series at time t. We could say the SOI leads the Recruitment series by six months. The sign of the CCF at $h = -6$ is negative, leading to the conclusion that the two series move in different directions; that is, increases in SOI lead to decreases in Recruitment and vice versa. Again, note the periodicity of 12 months in the CCF.

The flat lines shown on the plots indicate $\pm 2/\sqrt{453}$, so that upper values would be exceeded about 2.5% of the time if the noise were white as specified in Property 1.3 and Property 1.4. Of course, neither series is noise, so we can ignore these lines. To reproduce Figure 1.15 in R, use the following commands:

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

Example 1.26 Prewhitening and Cross Correlation Analysis

Although we do not have all the tools necessary yet, it is worthwhile discussing the idea of prewhitening a series prior to a cross-correlation analysis. The basic idea is simple; in order to use Property 1.4, at least one of the series must be white noise. If this is not the case, there is no simple way to tell if a cross-correlation estimate is significantly different from zero. Hence, in Example 1.25, we were only guessing at the linear dependence relationship between SOI and Recruitment.

For example, in Figure 1.16 we generated two series, x_t and y_t, for $t = 1, \ldots, 120$ independently as

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

where $\{w_{t1}, w_{t2}; t = 1, \ldots, 120\}$ are all independent standard normals. The series are made to resemble SOI and Recruitment. The generated data are shown in the top row of the figure. The middle row of Figure 1.16 show the sample ACF of each series, each of which exhibits the cyclic nature of each series. The bottom row (left) of Figure 1.16 shows the sample CCF between x_t and y_t, which appears to show cross-correlation even though the series are independent. The bottom row (right) also displays the sample CCF between x_t and the prewhitened y_t, which shows that the two sequences are uncorrelated. By prewhtiening y_t, we mean that the signal has been removed from the data by running a regression of y_t on $\cos(2\pi t)$ and $\sin(2\pi t)$ [see Example 2.10] and then putting $\tilde{y}_t = y_t - \hat{y}_t$, where $\hat{y}_t$ are the predicted values from the regression.

The following code will reproduce Figure 1.16.

```
set.seed(1492); num = 120; t = 1:num
X = ts(2*cos(2*pi*t/12) + rnorm(num), freq=12 )
Y = ts(2*cos(2*pi*(t+5)/12) + rnorm(num), freq=12 )
Yw = resid( lm(Y~ cos(2*pi*t/12) + sin(2*pi*t/12), na.action=NULL) )
par(mfrow=c(3,2), mgp=c(1.6,.6,0), mar=c(3,3,1,1) )
plot(X);  plot(Y)
acf1(X, 48);  acf1(Y, 48)
ccf2(X, Y, 24);  ccf2(X, Yw, 24, ylim=c(-.6,.6))
```

Problems

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

1.2 (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 1.8. 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. Note: `v = filter(x, rep(1/4, 4), sides = 1)`

(b) Repeat (a) but with

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

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

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

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

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

1.3 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 $E[x_t] = \mu_t$.

1.4 Consider the time series

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

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

(a) Determine whether x_t is stationary.
(b) Show that the process $y_t = x_t - x_{t-1}$ is stationary.
(c) Show that the mean of the moving average

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

is $\beta_0 + \beta_1 t$.

1.5 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 and sketch the ACF as a function of h.

1.6 We have not discussed the stationarity of autoregressive models, and we will do that in Chapter 3. But for now, let $x_t = \phi x_{t-1} + w_t$ where $w_t \sim wn(0,1)$ and ϕ is a constant. Assume x_t is stationary and x_{t-1} is uncorrelated with the noise term w_t.

(a) Show that mean function of x_t is $\mu_{xt} = 0$.
(b) Show $\gamma_x(0) = \text{var}(x_t) = 1/(1-\phi^2)$.
(c) For which values of ϕ does the solution to part (b) make sense?
(d) Find the lag-one autocorrelation, $\rho_x(1)$.

1.7 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 1.4b.)

1.8 Would you treat the global temperature data discussed in Example 1.2 and shown in Figure 1.3 as stationary or non-stationary? Support your answer.

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, $t+m$, for $m > 0$.

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

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

is minimized by the value

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

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

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

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

1.11 For two jointly stationary series x_t and y_t, verify (1.25).

1.12 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 $h = 0, \pm1, \pm2, \ldots$ 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.

1.13 Let w_t, for $t = 0, \pm1, \pm2, \ldots$ 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.

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

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

(b) Show that the autocovariance function of x_t is given by $\gamma_x(0) = \sigma_w^2(1+\theta^2)$, $\gamma_x(\pm 1) = \sigma_w^2\theta$, and $\gamma_x(h) = 0$ otherwise.
(c) Show that x_t is stationary for all values of $\theta \in \mathbb{R}$.
(d) Use (1.27) to calculate $\mathrm{var}(\bar{x})$ for estimating μ when (i) $\theta = 1$, (ii) $\theta = 0$, and (iii) $\theta = -1$
(e) In time series, the sample size n is typically large, so that $\frac{(n-1)}{n} \approx 1$. With this as a consideration, comment on the results of part (d); in particular, how does the accuracy in the estimate of the mean μ change for the three different cases?

1.15 (a) Simulate a series of $n = 500$ Gaussian white noise observations as in Example 1.6 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 1.18.]
(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

1.16 (a) Simulate a series of $n = 500$ moving average observations as in Example 1.7 and compute the sample ACF, $\widehat{\rho}(h)$, to lag 20. Compare the sample ACF you obtain to the actual ACF, $\rho(h)$. [Recall Example 1.19.]
(b) Repeat part (a) using only $n = 50$. How does changing n affect the results?

1.17 Simulate 500 observations from the AR model specified in Example 1.8 and then plot the sample ACF to lag 50. What does the sample ACF tell you about the approximate cyclic behavior of the data? Hint: Recall Example 1.25.

1.18 Simulate a series of $n = 500$ observations from the signal-plus-noise model presented in Example 1.10 with (a) $\sigma_w = 0$, (b) $\sigma_w = 1$ and (c) $\sigma_w = 5$. Compute the sample ACF to lag 100 of the three series you generated and comment.

1.19 For the time series y_t described in Example 1.24, verify the stated result that $\rho_y(1) = -.4$ and $\rho_y(h) = 0$ for $h > 1$.

Chapter 2

Time Series Regression and EDA

2.1 Classical Regression for Time Series

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

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq} + w_t, \tag{2.1}$$

where $\beta_0, \beta_1, \ldots, \beta_q$ 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.

Example 2.1 Estimating a Linear Trend

Consider the monthly price (per pound) of a chicken in the US from mid-2001 to mid-2016 (180 months), say x_t, shown in Figure 2.1. There is an obvious upward trend in the series, and we might use simple linear regression to estimate that trend by fitting the model

$$x_t = \beta_0 + \beta_1 z_t + w_t, \quad z_t = 2001\tfrac{7}{12}, 2001\tfrac{8}{12}, \ldots, 2016\tfrac{6}{12}.$$

This is in the form of the regression model (2.1) with $q = 1$. Note that we are making the assumption that the errors, w_t, are an iid normal sequence, which may not be true; the problem of autocorrelated errors is discussed in detail in Chapter 3.

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

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

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

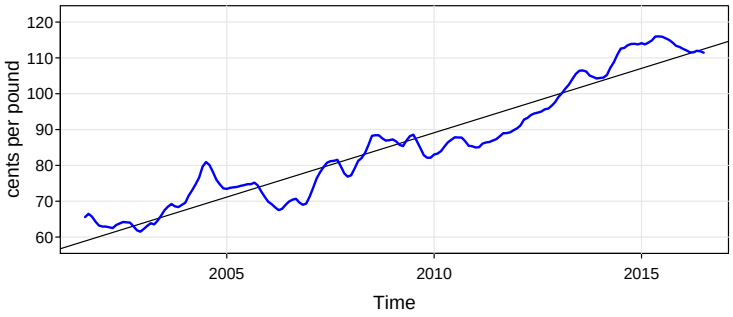


Fig. 2.1. *The price of chicken: monthly whole bird spot price, Georgia docks, US cents per pound, August 2001 to July 2016, with fitted linear trend line.*

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

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

Using R, we obtained the estimated slope coefficient of $\hat{\beta}_1 = 3.59$ (with a standard error of .08) yielding a highly significant estimated increase of about 3.6 cents *per year*.[1] Finally, Figure 2.1 shows the data with the estimated trend line superimposed. To perform this analysis in R, use the following commands:

```
summary(fit <- lm(chicken~time(chicken)))  # regress price on time
tsplot(chicken, ylab="cents per pound")
abline(fit)           # add the fitted regression line to the plot
```

The multiple linear regression model described by (2.1) can be conveniently written in a more general notation by defining the column vectors $z_t = (1, z_{t1}, z_{t2}, \ldots, z_{tq})'$ and $\beta = (\beta_0, \beta_1, \ldots, \beta_q)'$, where $'$ denotes transpose, so (2.1) can be written in the alternate form

$$x_t = \beta_0 + \beta_1 z_{t1} + \cdots + \beta_q z_{tq} + w_t = \beta' z_t + w_t. \tag{2.2}$$

where $w_t \sim \text{iid } \mathrm{N}(0, \sigma_w^2)$. As in the previous example, OLS estimation minimizes the error sum of squares

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

with respect to $\beta_0, \beta_1, \ldots, \beta_q$. This minimization can be accomplished by solving $\partial Q/\partial \beta_i = 0$ for $i = 0, 1, \ldots, q$, which yields $q+1$ equations with $q+1$ unknowns. In vector notation, this procedure gives the *normal equations*

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

If $\sum_{t=1}^{n} z_t z_t'$ is non-singular, the least squares estimate of β is

$$\widehat{\beta} = \left(\sum_{t=1}^{n} z_t z_t'\right)^{-1} \sum_{t=1}^{n} z_t x_t.$$

The ordinary least squares estimator is UNBIASED and it achieves the smallest variance among all the **linear** unbiased estimators, provided that the noise is WHITE.

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

The minimized error sum of squares (2.3), denoted SSE, can be written as

$$SSE = \sum_{t=1}^{n} (x_t - \widehat{x}_t)^2 = \sum_{t=1}^{n} (x_t - \widehat{\beta}' z_t)^2. \tag{2.5}$$

The ordinary least squares estimators are unbiased, i.e., $E(\widehat{\beta}) = \beta$, and have the smallest variance within the class of linear unbiased estimators.

This is true under the assumption of white noise. Best Linear Unbiased Estimator (BLUE)

If the errors w_t are normally distributed, $\widehat{\beta}$ is normally distributed with

The Gaussion assumption is useful in order to use the subsequent tests

$$\operatorname{cov}(\widehat{\beta}) = \sigma_w^2 C, \tag{2.6}$$

where

$$C = \left(\sum_{t=1}^{n} z_t z_t' \right)^{-1} \tag{2.7}$$

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

$$s_w^2 = MSE = \frac{SSE}{n - (q+1)}, \tag{2.8}$$

Note that we do not divide by n but by n-(q+1). If we divide by n, the estimator turns out to be BIASED.

where MSE denotes the *mean squared error*. Under the normal assumption,

$$t = \frac{(\widehat{\beta}_i - \beta_i)}{s_w \sqrt{c_{ii}}} \tag{2.9}$$

has the t-distribution with $n - (q+1)$ degrees of freedom; c_{ii} denotes the i-th diagonal element of C, as defined in (2.7). This result is often used for individual tests of the null hypothesis $H_0 : \beta_i = 0$ for $i = 1, \ldots, q$.

Such tests are important in order to test whether a weight is significant or not

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

$$x_t = \beta_0 + \beta_1 z_{t1} + \cdots + \beta_r z_{tr} + w_t \tag{2.10}$$

Model determination techniques, in a step wise fashion

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

The null hypothesis in this case is $H_0 : \beta_{r+1} = \cdots = \beta_q = 0$. We can test the reduced model (2.10) against the full model (2.2) by comparing the error sums of squares under the two models using the F-statistic

$$F = \frac{(SSE_r - SSE)/(q-r)}{SSE/(n-q-1)} = \frac{MSR}{MSE}, \tag{2.11}$$

where SSE_r is the error sum of squares under the reduced model (2.10). Note that $SSE_r \geq SSE$ because the reduced model has fewer parameters. If $H_0 : \beta_{r+1} = \cdots = \beta_q = 0$ is true, then $SSE_r \approx SSE$ because the estimates of those βs will be close to 0. Hence, we do not believe H_0 if $SSR = SSE_r - SSE$ is big. Under the null hypothesis, (2.11) has a central F-distribution with $q - r$ and $n - q - 1$ degrees of freedom when (2.10) is the correct model.

These results are often summarized in an 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 (SSR). The null hypothesis is rejected at level α if

ANOVA Analysis of variance

Table 2.1. *Analysis of Variance for Regression*

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

SSEr is always larger then SSE, since we use less parameters. If SSR is small, this means that the eliminated parameters are NOT important, since they do not contribute much in the error.

$F > F^{q-r}_{n-q-1}(\alpha)$, the $1-\alpha$ percentile of the F distribution with $q-r$ numerator and $n-q-1$ denominator degrees of freedom.

A special case of interest is $H_0: \beta_1 = \cdots = \beta_q = 0$. In this case $r = 0$, and the model in (2.10) becomes

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

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

$$R^2 = \frac{SSE_0 - SSE}{SSE_0}\,, \tag{2.12}$$

The smaller the SSE the larger the R

where the residual sum of squares under the reduced model is

$$SSE_0 = \sum_{t=1}^{n}(x_t - \bar{x})^2\,. \tag{2.13}$$

In this case SSE_0 is the sum of squared deviations from the mean $\bar{x}$ and is otherwise known as the adjusted total sum of squares (SST). The measure $R^2 = SSR/SST$ is called the *coefficient of determination.*

The techniques discussed in the previous paragraph can be used for model selection; e.g., stepwise regression. Another approach is based on *parsimony* where we try to find the model that has the best fit with the fewest number of parameters. Suppose we consider a normal regression model with k coefficients and denote the (maximum likelihood) estimator for the variance as

Model PARSIMONY

$$\widehat{\sigma}^2_k = \frac{SSE(k)}{n}, \tag{2.14}$$

where $SSE(k)$ denotes the residual sum of squares under the model with k regression coefficients. Then, Akaike (1969, 1973, 1974) suggested balancing the error of the fit against the number of parameters in the model as follows.

Definition 2.1 Akaike's Information Criterion (AIC)

$$\text{AIC} = \log\,\widehat{\sigma}^2_k + \frac{n+2k}{n}, \tag{2.15}$$

where $\widehat{\sigma}^2_k$ is given by (2.14) and k is the number of parameters in the model.

The more parameters we use the smaller the error. So, where to stop? The idea here is that we want the error to be small, but at the same time NOT to use excessively large number of parameters. So, this criterion gives a TRADE-OFF. Small ENOUGH error and small ENOUGH number of parameters.

The value of k yielding the minimum AIC specifies the best model.[2] The idea is roughly that minimizing $\widehat{\sigma}^2_k$ would be a reasonable objective, except that it decreases monotonically as k increases. Therefore, we ought to penalize the error

[2] Formally, AIC is defined as $-2\log L_k + 2k$ where L_k is the maximum value of the likelihood and k is the number of parameters in the model. For the normal regression problem, AIC can be reduced to the form given by (2.15).

variance by a term proportional to the number of parameters. The choice for the penalty term given by (2.15) is not the only one, and a considerable literature is available advocating different penalty terms. A corrected form, suggested by Sugiura (1978), and expanded by Hurvich and Tsai (1989), can be based on small-sample distributional results for the linear regression model. The corrected form is defined as follows.

Definition 2.2 AIC, Bias Corrected (AICc)

$$\text{AICc} = \log \widehat{\sigma}_k^2 + \frac{n+k}{n-k-2}, \tag{2.16}$$

where $\widehat{\sigma}_k^2$ is given by (2.14), 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)

$$\text{BIC} = \log \widehat{\sigma}_k^2 + \frac{k \log n}{n}, \tag{2.17}$$

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

$$M_t = \beta_0 + \beta_1 t + w_t \tag{2.18}$$

$$M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T.) + w_t \tag{2.19}$$

$$M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T.) + \beta_3 (T_t - T.)^2 + w_t \tag{2.20}$$

$$M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T.) + \beta_3 (T_t - T.)^2 + \beta_4 P_t + w_t \tag{2.21}$$

trend

linearity w.r. to temperature

curvlinearity w.r. to temperature

curvlinear w.r. to temperature, linear w.r. to particulates

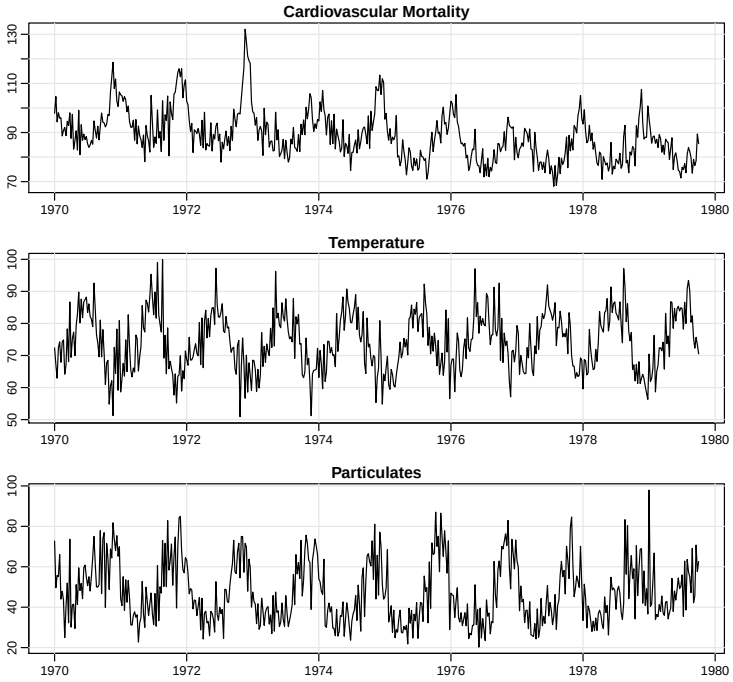


Fig. 2.2. *Average weekly cardiovascular mortality (top), temperature (middle) and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.*

Table 2.2. *Summary Statistics for Mortality Models*

Model	k	SSE	df	MSE	R^2	AIC	BIC
(2.18)	2	40,020	506	79.0	.21	5.38	5.40
(2.19)	3	31,413	505	62.2	.38	5.14	5.17
(2.20)	4	27,985	504	55.5	.45	5.03	5.07
(2.21)	5	20,508	503	40.8	.60	4.72	4.77

where we adjust temperature for its mean, $T_{\cdot} = 74.26$, to avoid collinearity problems. It is clear that (2.18) is a trend only model, (2.19) is linear temperature, (2.20) is curvilinear temperature and (2.21) is curvilinear temperature and pollution. We summarize some of the statistics given for this particular case in Table 2.2.

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 and AICc are nearly the same). Note that one can compare any two models using the residual sums of squares and (2.11). Hence, a model with only trend could be compared to the full model using $q = 4, r = 1, n = 508$, so

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

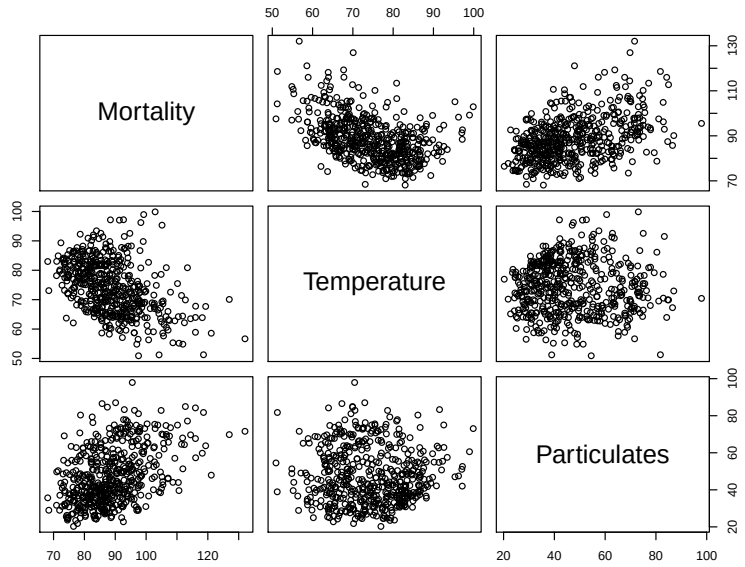


Fig. 2.3. *Scatterplot matrix showing relations between mortality, temperature, and pollution.*

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

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

Observe a) the negative trend with time, b) the negative dependence with temperature and c) the postive dependence with pollutants

for mortality, where the standard errors, computed from (2.6)–(2.8), 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 $\hat{w}_t = M_t - \hat{M}_t$ for autocorrelation (of which there is a substantial amount), but we defer this question to Section 3.8 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 (2.21), and compute the corresponding values of AIC, AICc and BIC.[3] 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 the data
tsplot(cmort, main="Cardiovascular Mortality",  ylab="")
tsplot(tempr, main="Temperature", ylab="")
tsplot(part, main="Particulates",   ylab="")
dev.new()          # open a new graphic device
ts.plot(cmort,tempr,part, col=1:3)  # all on same plot (not shown)
legend('topright', legend=c('Mortality', 'Temperature', 'Pollution'),
           lty=1, col=1:3)
```

[3] The easiest way to extract AIC and BIC from an `lm()` run in R is to use the command `AIC()` or `BIC()`. Our definitions differ from R by terms that do not change from model to model. In the example, we show how to obtain (2.15) and (2.17) from the R output. It is more difficult to obtain AICc.

```
dev.new()
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
BIC(fit)/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 Problem 2.2. The following is a simple example of lagged regression.

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 strong evidence that the relationship is NOT linear (this is discussed further in Example 2.8), *for demonstration purposes only*, we consider the following regression,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t, \tag{2.22}$$

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

$$\widehat{R}_t = 65.79 - 44.28_{(2.78)} S_{t-6} \tag{2.23}$$

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 we defer this discussion 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 an object that we call `fish` using `ts.intersect`, which aligns the lagged series.

```
fish = ts.intersect( rec, soiL6=lag(soi,-6) )
summary(fit1 <- lm(rec~ soiL6, data=fish, na.action=NULL))
```

The headache of aligning the lagged series can be avoided by using the R package `dynlm`, which must be downloaded and installed.

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

In the `dynlm` example, `fit2` is similar to a `lm` object, but the time series attributes are retained without any additional commands.

2.2 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 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 1.3 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

This model that uses trend in the mean plus a staionary time series allows for "stationarity AROUND the trend"

$$x_t = \mu_t + y_t \tag{2.24}$$

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

If the estimate of the trend is reasonable, the residual is close enough to be stationary

$$\widehat{y}_t = x_t - \widehat{\mu}_t. \tag{2.25}$$

Consider the following example.

Example 2.4 Detrending Chicken Prices

Here we suppose the model is of the form of (2.24),

$$x_t = \mu_t + y_t,$$

where, as we suggested in the analysis of the chicken price data presented in Example 2.1, a straight line might be useful for detrending the data; i.e.,

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

we model the trend as a straight line

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

$$\widehat{\mu}_t = -7131 + 3.59\, t.$$

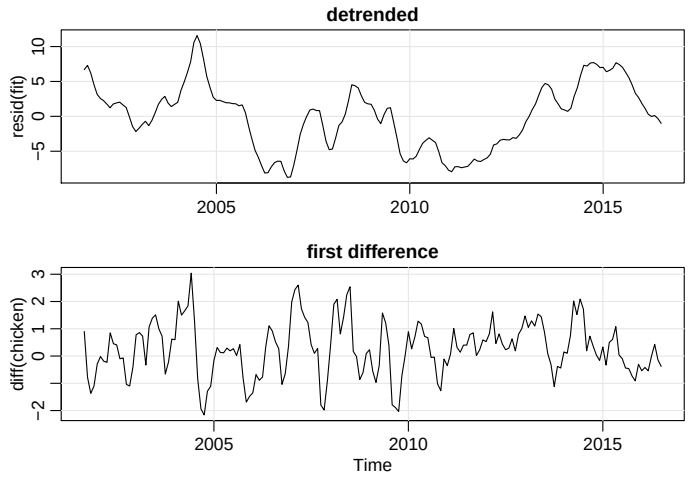


Fig. 2.4. *Detrended (top) and differenced (bottom) chicken price series. The original data are shown in Figure 2.1.*

Figure 2.1 shows the data with the estimated trend line superimposed. To obtain the detrended series we simply subtract $\hat{\mu}_t$ from the observations, x_t, to obtain the detrended series[4]

$$\hat{y}_t = x_t + 7131 - 3.59\,t.$$

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).

Note that to obtain this, the time series y was considered as a noise (Example 2.1). Of course, now, y is NOT white noise, in general, and the use of LS is not the best criterion choice.

In Example 1.9 and the corresponding Figure 1.9 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,

linear trend with time

$$\mu_t = \delta + \mu_{t-1} + w_t, \tag{2.26}$$

we model the TREND as a random walk

where w_t is white noise and is independent of y_t. If the appropriate model is (2.24), then differencing the data, x_t, yields a stationary process; that is,

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + w_t + y_t - y_{t-1}. \end{aligned} \tag{2.27}$$

δ can be obtained as the mean of the differenced series.

It is easy to show $z_t = y_t - y_{t-1}$ is stationary using Property 1.1. That is, because y_t is stationary,

$$\begin{aligned} \gamma_z(h) &= \operatorname{cov}(z_{t+h}, z_t) = \operatorname{cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\ &= 2\gamma_y(h) - \gamma_y(h+1) - \gamma_y(h-1) \end{aligned} \tag{2.28}$$

[4] 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 with regard to efficiency.

is independent of time; we leave it as an exercise (Problem 2.5) to show that $x_t - x_{t-1}$ in (2.27) 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 (2.27). 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 $\mu_t = \beta_0 + \beta_1 t$ in the model (2.24), differencing the data produces stationarity (see Problem 2.4):

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

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

$$\nabla x_t = x_t - x_{t-1}. \tag{2.29}$$

First Difference Operator

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of (2.29), 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,*

$$B^k x_t = x_{t-k}. \tag{2.30}$$

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

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

That is, B^{-1} is the *forward-shift operator*. In addition, it is clear that we may rewrite (2.29) as

$$\nabla x_t = (1 - B)x_t, \tag{2.31}$$

Note that this is not a polynomial. B does NOT take values. It is a SYMBOL.

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

$$\nabla^2 x_t = (1 - B)^2 x_t = (1 - 2B + B^2)x_t = x_t - 2x_{t-1} + x_{t-2} \tag{2.32}$$

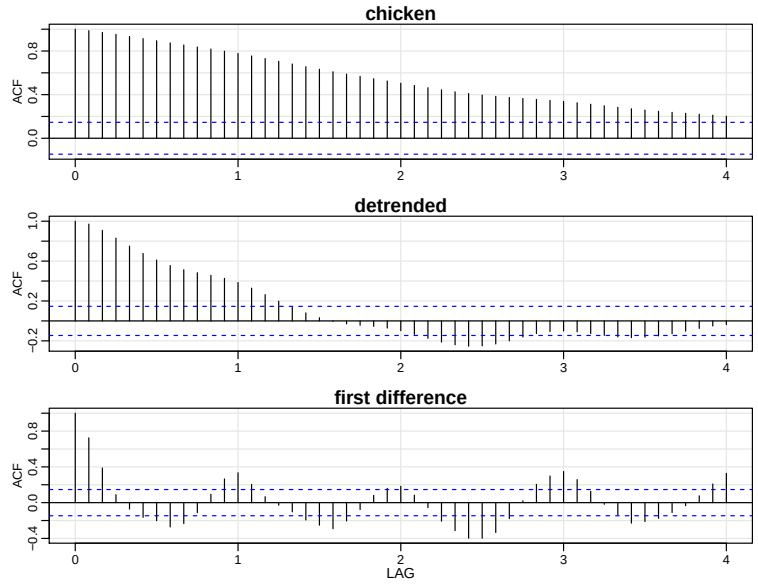
by the linearity of the operator. To check, just take the difference of the first difference $\nabla(\nabla x_t) = \nabla(x_t - x_{t-1}) = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2})$.

Definition 2.5 Differences of order d *are defined as*

$$\nabla^d = (1 - B)^d,$$

Use Algebra to perform difference

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.*



Observe that the ACFs of the detrended and the differenced time series reveal different information that underlies the original time series.

Fig. 2.5. *Sample ACFs of chicken prices (top), and of the detrended (middle) and the differenced (bottom) series. Compare the top plot with the sample ACF of a straight line:* `acf(1:100)`.

The first difference (2.29) 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 4. The differencing technique is an important component of the ARIMA model discussed in Chapter 3.

Example 2.5 Differencing Chicken Prices

The first difference of the chicken prices series, also shown in Figure 2.4, produces different results than removing trend by detrending via regression. For example, the five-year business cycle we observed in the detrended series is not obvious in the differenced series (although it is still there, which can be verified using Chapter 4 techniques).

The ACF of this series is also shown in Figure 2.5. In this case, the difference series exhibits an annual cycle that was not seen in the original or detrended data.

The R code to reproduce Figure 2.4 and Figure 2.5 is as follows.

```
fit = lm(chicken~time(chicken), na.action=NULL) # regress chicken on time
par(mfrow=c(2,1))
tsplot(resid(fit), main="detrended")
tsplot(diff(chicken), main="first difference")
par(mfrow=c(3,1))  # plot ACFs
acf1(chicken, 48, main="chicken")
acf1(resid(fit), 48, main="detrended")
acf1(diff(chicken), 48, main="first difference")
```

Example 2.6 Differencing Global Temperature

The global temperature series shown in Figure 1.3 appears to behave more as a random walk than a trend stationary series. Hence, rather than detrend the data, it would be more appropriate to use differencing to coerce it into stationarity.

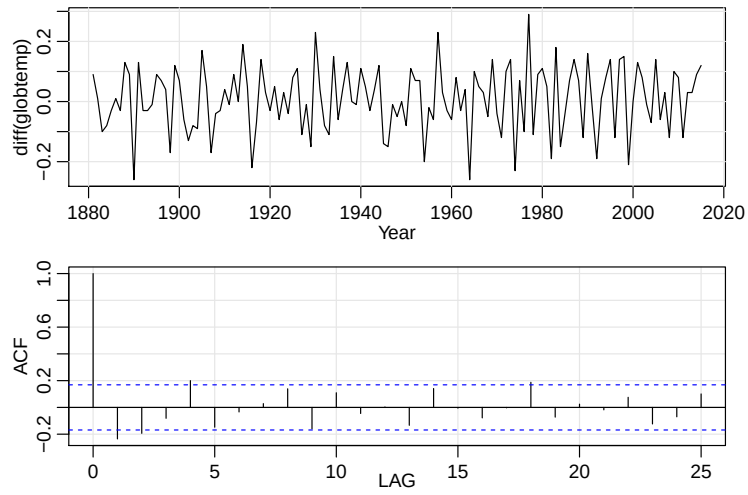


Fig. 2.6. *Differenced global temperature series and its sample ACF.*

The detreded data are shown in Figure 2.6 along with the corresponding sample ACF. 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 .008, or an increase of about one degree centigrade per 100 years.

The R code to reproduce ~~Figure 2.4 and Figure 2.5 is as follows.~~

```
par(mfrow=c(2,1))
tsplot(diff(globtemp), type="o")
 mean(diff(globtemp))  # drift estimate = .008
acf1(diff(gtemp), 48)
```

Figure 2.6

NON LINEAR TRANSFORMATIONS

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

$$y_t = \log x_t, \tag{2.34}$$

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

Nonlinear transformations are used to EQUALIZE the variablity over time

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log x_t & \lambda = 0. \end{cases} \tag{2.35}$$

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.

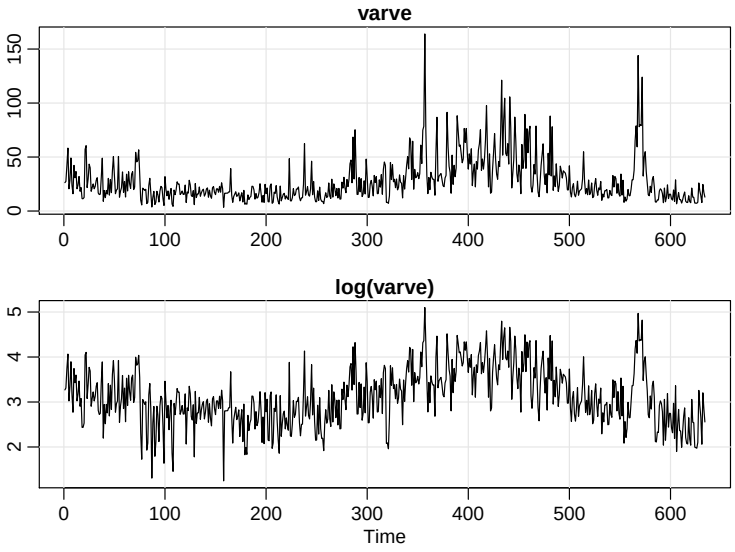


Fig. 2.7. *Glacial varve thicknesses (top) from Massachusetts for* $n = 634$ *years compared with* log *transformed thicknesses (bottom).*

Example 2.7 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 2.7 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 2.7 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 2.6, to argue that the approximation to normality is improved. The ordinary first differences (2.31) are also computed in Problem 2.6, 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 2.7 was generated in R as follows:

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

SCATTERPLOT MATRICES

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

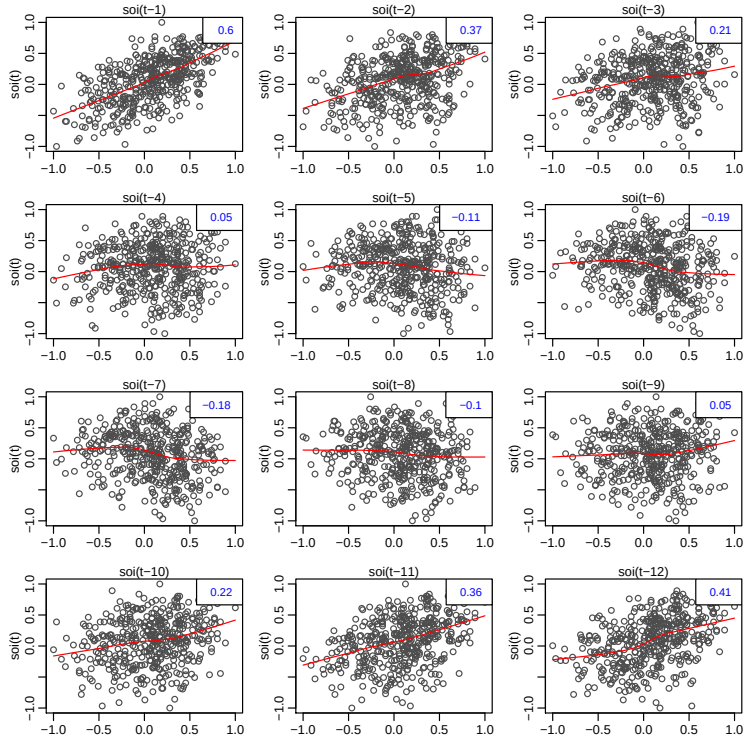


Fig. 2.8. *Scatterplot matrix relating current SOI values, S_t, to past SOI values, S_{t-h}, at lags $h = 1, 2, ..., 12$. The values in the upper right corner are the sample autocorrelations and the lines are a lowess fit.*

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}.

Shows wich values of h lead to the best predictability

Example 2.8 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 2.8, 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 method for fitting local regression.

Locally Weighted Scatterplot Smoothing (LOWESS) or (LOESS)

In Figure 2.8, 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 $h = 6, 7$.

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

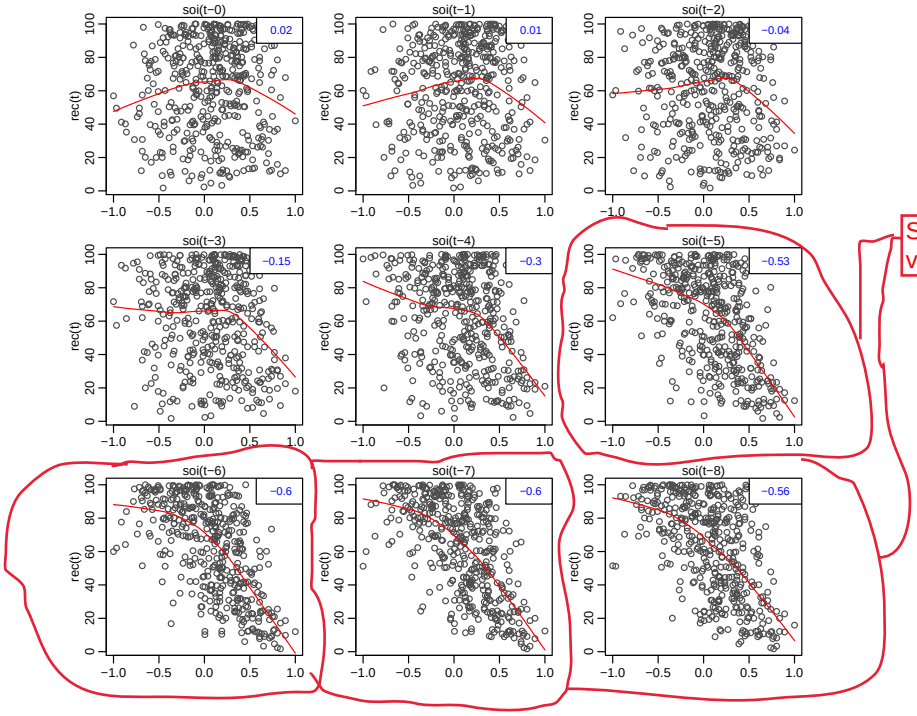


Fig. 2.9. *Scatterplot matrix of the Recruitment series, R_t, on the vertical axis plotted against the SOI series, S_{t-h}, on the horizontal axis at lags $h = 0, 1, \ldots, 8$. The values in the upper right corner are the sample cross-correlations and the lines are a lowess fit.*

example, we might wish to predict the Recruitment series, R_t, from current or past values of the SOI series, S_{t-h}, for $h = 0, 1, 2, \ldots$ it would be worthwhile to examine the scatterplot matrix. Figure 2.9 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 2.9 shows a fairly strong nonlinear relationship between Recruitment, R_t, and the SOI series at $S_{t-5}, S_{t-6}, S_{t-7}, S_{t-8}$, indicating the SOI series tends to lead the Recruitment series and the coefficients are negative, implying that increases in the SOI lead to decreases in the Recruitment. The nonlinearity observed in the scatterplots (with the help of the superimposed lowess fits) indicates that the behavior between Recruitment and the SOI is different for positive values of SOI than for negative values of SOI.

The R code for this example is

```
lag1.plot(soi, 12)       # Figure 2.8
lag2.plot(soi, rec, 8)   # Figure 2.9
```

Example 2.9 Regression with Lagged Variables (cont)

In Example 2.3 we regressed Recruitment on lagged SOI,

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

However, in Example 2.8, we saw that the relationship is nonlinear and different when SOI is positive or negative. In this case, we may consider adding a

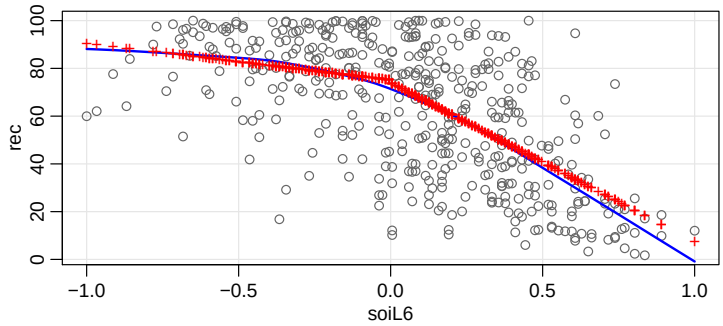


Fig. 2.10. *Display for Example 2.9: Plot of Recruitment (R_t) vs SOI lagged 6 months (S_{t-6}) with the fitted values of the regression as points (+) and a lowess fit (—).*

dummy variable to account for this change. In particular, we fit the model

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

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

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

The result of the fit is given in the R code below. Figure 2.10 shows R_t vs S_{t-6} with the fitted values of the regression and a lowess fit superimposed. The piecewise regression fit is similar to the lowess fit, but we note that the residuals are not white noise (see the code below). This is followed up in Example 3.33.

```
dummy = ifelse(soi<0, 0, 1)
fish = ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6), dframe=TRUE)
summary(fit <- lm(rec~ soiL6*dL6, data=fish, na.action=NULL))
  Coefficients:
              Estimate Std. Error t value Pr(>|t|)
  (Intercept)   74.479      2.865  25.998  < 2e-16
  soiL6        -15.358      7.401  -2.075   0.0386
  dL6           -1.139      3.711  -0.307   0.7590
  soiL6:dL6    -51.244      9.523  -5.381  1.2e-07
  ---
  Residual standard error: 21.84 on 443 degrees of freedom
  F-statistic: 99.43 on 3 and 443 DF,  p-value: < 2.2e-16
attach(fish)     #  so we can use the names of the variables in fish
plot(soiL6, rec)
lines(lowess(soiL6, rec), col=4, lwd=2)
points(soiL6, fitted(fit), pch='+', col=2)
tsplot(resid(fit))  # not shown ...
acf(resid(fit))     # ... but obviously not noise
```

ASSESING PERIODIC BEHAVIOR

As a final exploratory tool, we discuss assessing periodic behavior in time series data using regression analysis; this material may be thought of as an introduction to spectral analysis, which we discuss in detail in Chapter 4. In Example 1.10, 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

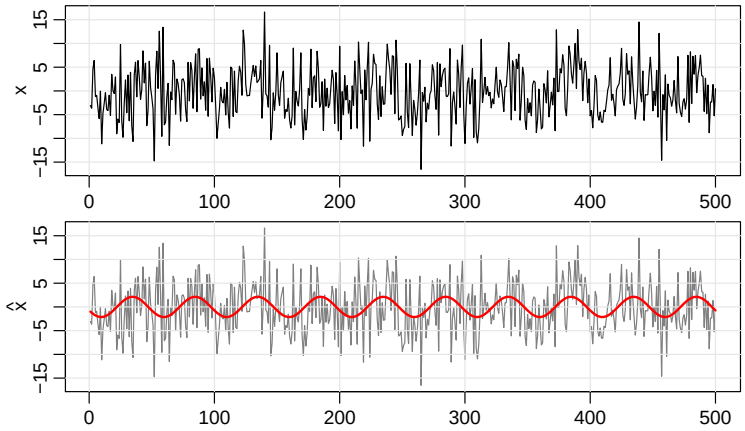


Fig. 2.11. *Data generated by (2.36) [top] and the fitted line superimposed on the data [bottom].*

shown in Figure 1.1 make one cycle every year (four quarters) on top of an increasing trend and the speech data in Figure 1.3 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 2.10 Using Regression to Discover a Signal in Noise

In Example 1.10, we generated $n = 500$ observations from the model

$$x_t = A\cos(2\pi\omega t + \phi) + w_t, \tag{2.36}$$

where $\omega = 1/50$, $A = 2$, $\phi = .6\pi$, and $\sigma_w = 5$; the data are shown on the bottom panel of Figure 1.10. 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 (2.36) in a nonlinear way, so we use a trigonometric identity[5] and write

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

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

Now the model (2.36) can be written in the usual linear regression form given by (no intercept term is needed here)

$$x_t = \beta_1\cos(2\pi t/50) + \beta_2\sin(2\pi t/50) + w_t. \tag{2.37}$$

Note, however, that this is an easy problem, since we know the number of

Using linear regression, we find $\hat{\beta}_1 = -.74_{(.33)}$, $\hat{\beta}_2 = -1.99_{(.33)}$ with $\hat{\sigma}_w = 5.18$; the values in parentheses are the standard errors. We note the actual values of the coefficients for this example are $\beta_1 = 2\cos(.6\pi) = -.62$, and $\beta_2 = -2\sin(.6\pi) = -1.90$. It is clear that we are able to detect the signal in the noise using regression, even though the signal-to-noise ratio is small. Figure 2.11 shows data generated by (2.36) with the fitted line superimposed.

To reproduce the analysis and Figure 2.11 in R, use the following:

```
set.seed(90210)              # so you can reproduce these results
x  = 2*cos(2*pi*1:500/50 + .6*pi) + rnorm(500,0,5)
```

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

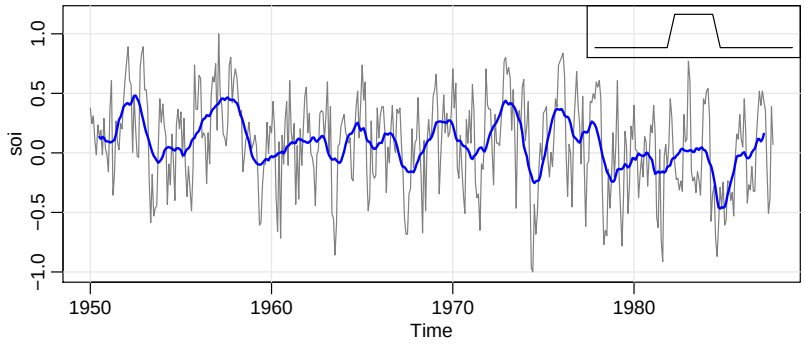


Fig. 2.12. *The SOI series smoothed using (2.38) with $k = 6$ (and half-weights at the ends). The insert shows the shape of the moving average ("boxcar") kernel [not drawn to scale] described in (2.40).*

```
z1 = cos(2*pi*1:500/50)
z2 = sin(2*pi*1:500/50)
summary(fit <- lm(x~ 0 + z1 + z2))  # zero to exclude the intercept
  Coefficients:
     Estimate Std. Error t value Pr(>|t|)
  z1  -0.7442     0.3274  -2.273   0.0235
  z2  -1.9949     0.3274  -6.093 2.23e-09
  Residual standard error: 5.177 on 498 degrees of freedom
par(mfrow=c(2,1))
tsplot(x, margins=.25)
tsplot(x, col=8, margins=.25, ylab=expression(hat(x)))
lines(fitted(fit), col=2)
```

We will discuss this and related approaches in more detail in Chapter 4.

2.3 Smoothing Time Series

In Section 1.4, we introduced the concept of smoothing a time series, and in Example 1.7, we discussed using a moving average to smooth white noise. This method is useful for discovering certain traits in a time series, such as long-term trend and seasonal components (see Section 4.7 for details). In particular, if x_t represents the observations, then

Smoothing is useful to learn long term trends

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j}, \tag{2.38}$$

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

Example 2.11 Moving Average Smoother

For example, Figure 2.12 shows the monthly SOI series discussed in Example 1.4 smoothed using (2.38) with weights $a_0 = a_{\pm1} = \cdots = a_{\pm5} = 1/12$, and $a_{\pm6} = 1/24$; $k = 6$. This particular method removes (filters out) the obvious annual temperature cycle and helps emphasize the El Niño cycle. To reproduce Figure 2.12 in R:

```
wgts = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=wgts)
```

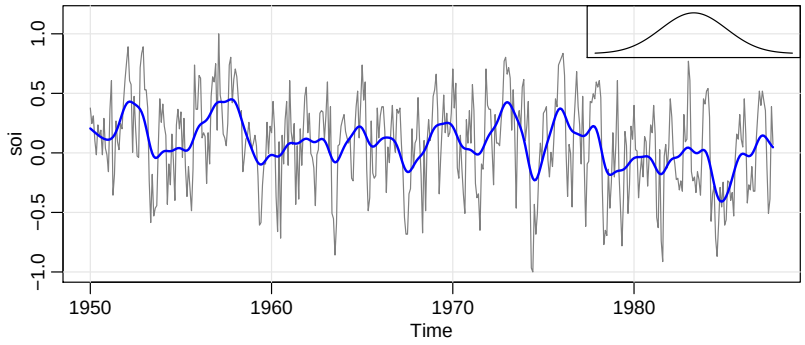


Fig. 2.13. *Kernel smoother of the SOI. The insert shows the shape of the normal kernel [not drawn to scale].*

```
tsplot(soi)
lines(soif, lwd=2, col=4)
```

Although the moving average smoother does a good job in highlighting the El Niño effect, it might be considered too choppy. We can obtain a smoother fit using the normal distribution for the weights, instead of boxcar-type weights of (2.38).

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 SOI series, where m_t is now

$$m_t = \sum_{i=1}^{n} w_i(t) x_{t_i}\,, \tag{2.39}$$

where

$$w_i(t) = K\left(\tfrac{t-t_i}{b}\right) \Big/ \sum_{j=1}^{n} K\left(\tfrac{t-t_j}{b}\right) \tag{2.40}$$

Note that the choice of b is VERY critical.

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

To implement this in R, use the `ksmooth` function where a bandwidth can be chosen. The wider the bandwidth, b, the smoother the result. In our case, we are smoothing over time, which is of the form $t/12$ for `soi`. In Figure 2.13, we used the value of $b = 1$ to correspond to approximately smoothing over about a year The R code for this example is

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

Example 2.13 Lowess

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

Lowess is a method of smoothing that is rather complex, but the basic idea is close to nearest neighbor regression. Figure 2.14 shows smoothing of SOI

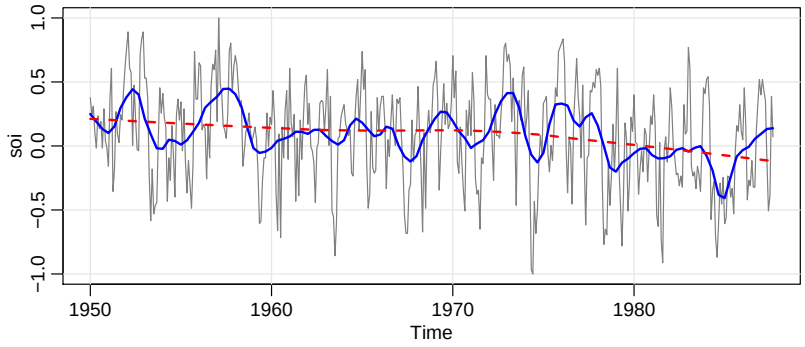


Fig. 2.14. *Locally weighted scatterplot smoothers (`lowess`) of the SOI series.*

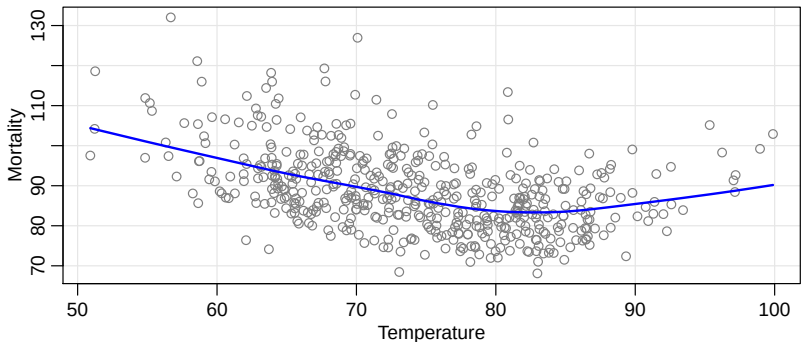


Fig. 2.15. *Smooth of mortality as a function of temperature using lowess.*

using the R function `lowess` (see Cleveland, 1979). First, a certain proportion of nearest neighbors to x_t are included in a weighting scheme; values closer to x_t in time get more weight. Then, a robust weighted regression is used to predict x_t and obtain the smoothed values m_t. The larger the fraction of nearest neighbors included, the smoother the fit will be. In Figure 2.14, one smoother uses 5% of the data to obtain an estimate of the El Niño cycle of the data. In addition, a (negative) trend in SOI would indicate the long-term warming of the Pacific Ocean. To investigate this, we used a lowess with the default smoother span of `f=2/3` of the data. Figure 2.14 can be reproduced in R as follows.

```
tsplot(soi)
lines(lowess(soi, f=.05), lwd=2, col=4)    # El Nino cycle
lines(lowess(soi), lty=2, lwd=2, col=2)    # trend (using default span)
```

Play with this code by chanigng the value of f

Example 2.14 Smoothing One Series as a Function of Another

Smoothing techniques can also be applied to smoothing a time series as a function of another time series. In Example 2.2, we discovered a nonlinear relationship between mortality and temperature. Continuing along these lines, Figure 2.15 shows a scatterplot of mortality, M_t, and temperature, T_t, along with M_t smoothed as a function of T_t using lowess Note that mortality increases at extreme 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 83° F. Figure 2.15 can be reproduced in R as follows using the defaults.

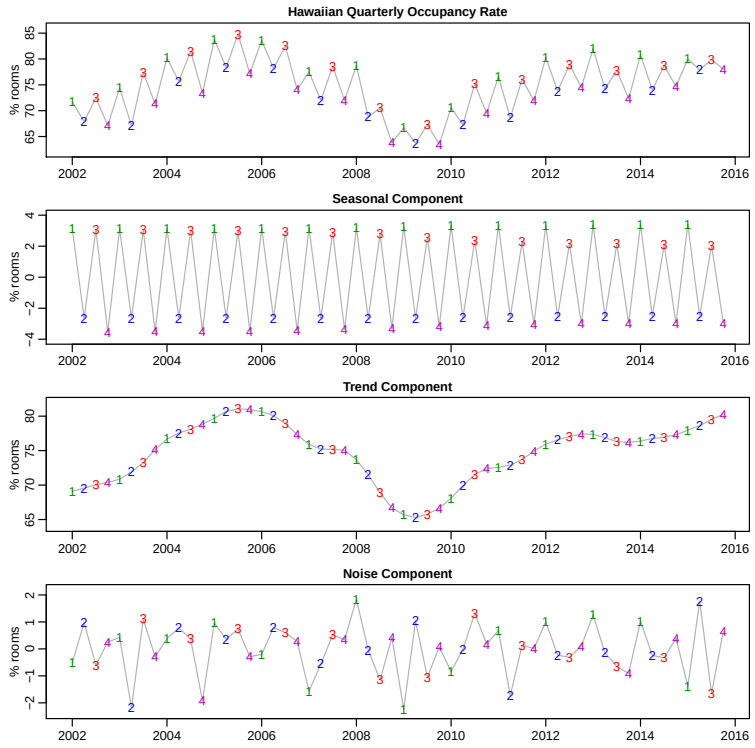


Fig. 2.16. *Structural model of the Hawaiian quarterly haole occupancy rate.*

```
plot(tempr, cmort, xlab="Temperature", ylab="Mortality")
lines(lowess(tempr, cmort))
```

Example 2.15 Classical Structural Modeling

A classical approach to time series analysis is to decompose data into components labeled trend (T_t), seasonal (S_t), irregular or noise (N_t). If we let x_t denote the data (or perhaps the data transformed to induce homoscedasticity), we can then sometimes write

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

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

Figure 2.16 shows the result of the decomposition using loess on the quarterly occupancy rate of Hawaiian hotels from 2002 to 2016. The data are in `astsa`, but only in versions higher than 1.7.1 (see the website for the text for instructions on obtaining the latest version of the package).

R provides a few scripts to fit the decomposition. The script `decompose` uses moving averages as in Example 2.11. Another script, `stl`, uses loess to obtain each component and is similar to the approach used in Example 2.13. To use `stl`, the seasonal smoothing method must be specified. That is, specify either the character string `"periodic"` or the span of the loess window for seasonal extraction. The span should be odd and at least 7 (there is no default). By using

a seasonal window, we are allowing $S_t \approx S_{t-4}$ rather than $S_t = S_{t-4}$, which is forced by specifying a periodic seasonal component; see the code below for more details.

The code for producing various graphs similar to Figure 2.16 is given below. Note that the seasonal component is very regular showing a 2% to 4% gain in the first and third quarters, while showing a 2% to 4% loss in the second and fourth quarters. The trend component perhaps more like a business cycle than what may be considered a trend. As previously implied, the components are not well defined and the decomposition is not unique; one person's trend may be another person's business cycle.

```
x = window(hor, start=2002)  # data set in astsa version 1.7.1+
plot(decompose(x))           # not shown
plot(stl(x, s.window='per')) # not shown
plot(stl(x, s.window=15))
```

Problems

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

(a) Fit the regression model

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

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

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

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

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

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

2.2 For the mortality data examined in Example 2.2:

(a) Add another component to the regression in (2.21) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression in (2.21). State your conclusion.

(b) Using AIC and BIC, is the model in (a) an improvement over the final model in Example 2.2?

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

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

```
par(mfrow=c(2,2), mar=c(2.5,2.5,0,0)+.5, mgp=c(1.6,.6,0))  # set up
 for (i in 1:4){
  x = ts(cumsum(rnorm(500,.01,1)))        # data
  regx = lm(x~0+time(x), na.action=NULL)  # regression
  tsplot(x, ylab='Random Walk w Drift', col='darkgray')   # plots
   abline(a=0, b=.01, col=2, lty=2)     # true mean (red - dashed)
   abline(regx, col=4)   }              # fitted line (blue - straight)
```

(b) Generate *four* series of length $n = 500$ that are linear trend plus noise, say $y_t = .01\,t + w_t$, where t and w_t are as in part (a). Fit the regression $y_t = \beta t + w_t$ using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01\,t$) and the fitted line, $\hat{y}_t = \hat{\beta}\,t$, on the same graph.

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

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

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

where β_0, β_1 are fixed constants.

(a) Prove x_t is nonstationary.

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

(c) Repeat part (b) if w_t is replaced by a general stationary process, say y_t, with mean function μ_y and autocovariance function $\gamma_y(h)$. [*Hint:* See (2.28).]

2.5 Show (2.27) is stationary.

2.6 The glacial varve record plotted in Figure 2.7 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 1.3?

(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 $p = (x_t - x_{t-1})/x_{t-1}$.

2.7 Use the three different smoothing techniques described in Example 2.11, Example 2.12, and Example 2.13, to estimate the trend in the global temperature series displayed in Figure 1.3. Comment.

2.8 In Section 2.3, we saw that the El Niño / La Niña cycle was approximately 4 years. To investigate whether there is a strong 4-year cycle, compare a sinusoidal (one cycle every four years) fit to the Southern Oscillation Index to a lowess fit (as in Example 2.13). In the sinusoidal fit, include a term for the trend. Discuss the results.

To get started, you can form the regressors for the sinusoidal fit as follows:

```
trnd = time(soi)
C4   = cos(2*pi*trnd/4)
S4   = sin(2*pi*trnd/4)
```

2.9 As in Problem 2.1, let y_t be the raw Johnson & Johnson series shown in Figure 1.1, and let $x_t = \log(y_t)$. Use each of the techniques mentioned in Example 2.15 to decompose the logged data as $x_t = T_t + S_t + N_t$ and describe the results. If you did Problem 2.1, compare the results of that problem with those found in this problem.

Chapter 3

ARIMA Models

3.1 Introduction

Classical regression is often insufficient for explaining all of the interesting dynamics of a time series. Instead, the introduction of correlation through lagged linear relationships leads to proposing the autoregressive (AR) and moving average (MA) models. Often, these models are combined to form the autoregressive moving average (ARMA) model. 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). Seasonal data, such as the data discussed in Example 1.1 and Example 1.4 lead to seasonal autoregressive integrated moving average (SARIMA) models. The Box–Jenkins method for identifying a plausible models is given in this chapter along with techniques for parameter estimation and forecasting.

3.2 Autoregressive Moving Average Models

First, we'll investigate autoregressive models, which are an obvious extension of linear regression models.

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

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

Autoregressive time series with mean value μ

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

We note that (3.2) is similar to the regression model of Section 2.1, and hence the term auto (or self) regression. Some technical difficulties develop from applying that model because the regressors, $x_{t-1}, \ldots, x_{t-p}$, are random components, whereas in regression, the regressors are assumed to be fixed. A useful form follows by using the backshift operator (2.30) 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

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

Example 3.1 The AR(1) Model

Consider the first-order model, AR(1), given by

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

Using the backshift operator as in (3.3) or (3.4), the model is

$$\underbrace{(1 - \phi B)}_{\phi(B)} x_t = w_t\,.$$

To get a feeling for the model, first suppose that x_t is stationary. We can rule out the case $\phi = 1$ because this would make x_t a random walk which we know is not stationary. Similarly, we can rule out $\phi = -1$.

If x_t is stationary, the mean function μ_t is constant so

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

implies $\mu_t = \phi\mu_t + 0$; thus

$$\mu_t = 0\,.$$

Consequently, if we replace x_t by $x_t - \mu$, then $E(x_t - \mu) = 0$, so the mean function is μ. In addition, assuming x_{t-1} and w_t are uncorrelated,

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

If x_t is stationary, the variance is constant, so

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

Thus

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

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

Similarly, we can show

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

Thus,

$$\rho(1) = \frac{\gamma(1)}{\gamma(0)} = \phi\,,$$

and we see that ϕ is in fact a correlation (again implying that $|\phi| < 1$).

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

$$x_t = \sum_{j=0}^{\infty} \phi^j w_{t-j}. \tag{3.5}$$

(3.5) See, also, footnote 1, below

Representation (3.5) is called the *causal stationary solution* of the model. The term causal refers to the fact that x_t does not depend on the future. In fact, by simple substitution,

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

Using (3.5), it is easy to see that the AR(1) process is stationary with mean

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

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

$$\begin{aligned}
\gamma(h) &= \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}\left(\sum_{j=0}^{\infty} \phi^j w_{t+h-j}, \sum_{k=0}^{\infty} \phi^k w_{t-k} \right) \\
&= \operatorname{cov}\left[\left(w_{t+h} + \cdots + \phi^h w_t + \phi^{h+1} w_{t-1} + \cdots \right), \left(w_t + \phi w_{t-1} + \cdots \right) \right] \\
&= \sigma_w^2 \sum_{j=0}^{\infty} \phi^{h+j} \phi^j = \sigma_w^2 \phi^h \sum_{j=0}^{\infty} \phi^{2j} = \frac{\sigma_w^2 \phi^h}{1 - \phi^2}, \quad h \geq 0.
\end{aligned} \tag{3.6}$$

Recall that $\gamma(h) = \gamma(-h)$, so we will only exhibit the autocovariance function for $h \geq 0$. From (3.6), the ACF of an AR(1) is

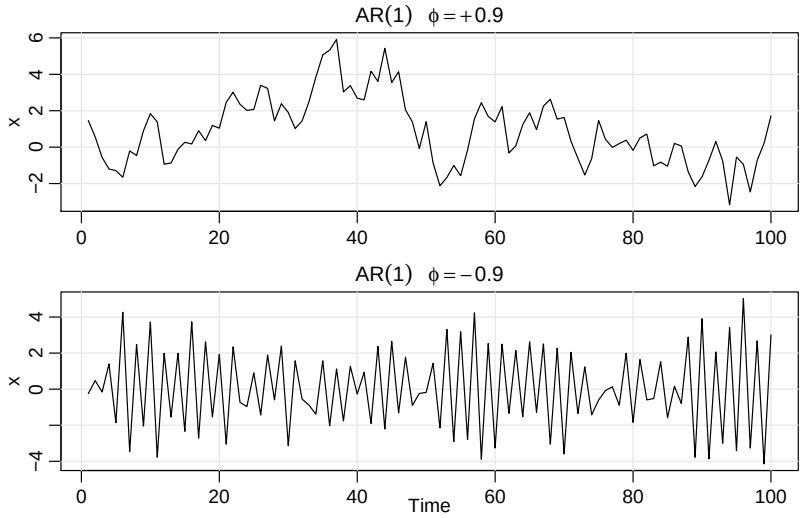
$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \phi^h, \quad h \geq 0. \tag{3.7}$$

In addition, from the causal form (3.5), we see that our assumption that x_{t-1} and w_t are uncorrelated are immediate because $x_{t-1} = \sum_{j=0}^{\infty} \phi^j w_{t-1-j}$ is a linear filter of past shocks, $w_{t-1}, w_{t-2}, \ldots$, which are uncorrelated with w_t, the present shock.

Example 3.2 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 $\phi = -.9$,

[1] Iterate backward, $x_t = \phi x_{t-1} + w_t = \phi(\phi x_{t-2} + w_{t-1}) + w_t = \phi^2 x_{t-2} + \phi w_{t-1} + w_t = \cdots = \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j}$. If $|\phi| < 1$ and $\sup_t E(x_t^2) < \infty$, then $\lim_{k\to\infty} E\left(x_t - \sum_{j=0}^{k-1} \phi^j w_{t-j} \right)^2 = \lim_{k\to\infty} \phi^{2k} E\left(x_{t-k}^2\right) = 0$.



Smooth

Choppy

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

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:

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

Play with this code for different values of ϕ, let us say: ϕ= +0.5, -0.5, +0.1, -0.1. What to you observe?

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

Definition 3.2 *The* **moving average model** *of order q, or* **MA(*q*)** *model, is defined to be*

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \cdots + \theta_q w_{t-q}, \tag{3.8}$$

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.*

As in the AR(p) case, the MA(q) model may be written as

$$x_t = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) w_t, \tag{3.9}$$

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

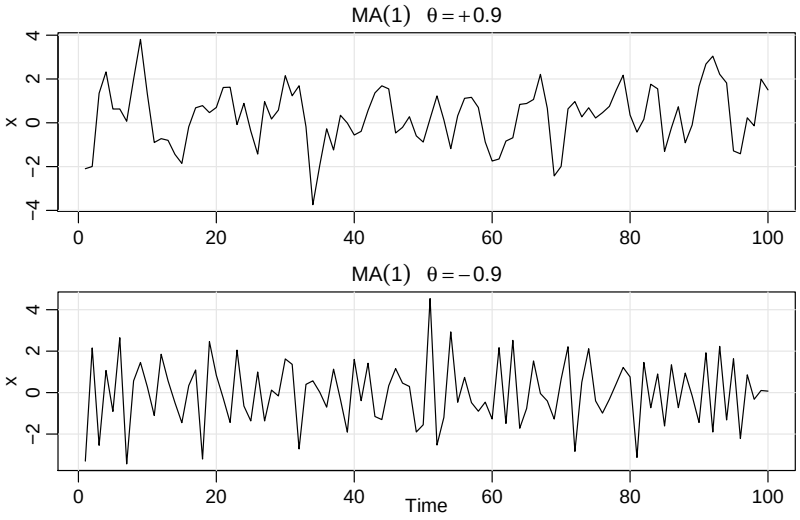


Fig. 3.2. *Simulated MA(1) models: $\theta = .9$ (top); $\theta = -.9$ (bottom).*

or more concisely as

$$x_t = \theta(B)w_t, \tag{3.10}$$

Unlike the autoregressive process, the moving average process is stationary for any values of the parameters $\theta_1, \ldots, \theta_q$.

Example 3.3 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) model in which the correlation between x_t and x_{t-k} is never zero. When $\theta = .9$, for example, x_t and x_{t-1} are positively correlated, and $\rho(1) = .497$. When $\theta = -.9$, x_t and x_{t-1} are negatively correlated, $\rho(1) = -.497$. Figure 3.2 shows a time plot of these two processes with $\sigma_w^2 = 1$. The series for which $\theta = .9$ is smoother than the series for which $\theta = -.9$. A figure similar to Figure 3.2 can be created in R as follows:

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

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

Definition 3.3 *A time series* $\{x_t;\ t = 0, \pm 1, \pm 2, \ldots\}$ *is* **ARMA**(p, q) *if it is stationary and*

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}, \tag{3.11}$$

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*

$$x_t = \alpha + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}. \tag{3.12}$$

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

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

where $\epsilon_t = w_t + \theta_1 w_{t-1} + \cdots + \theta_q w_{t-q}$, although we call the regression parameters ϕ instead of β.

As 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). Using (3.3) and (3.9), the ARMA(p, q) model in (3.11) may be written in concise form as

$$\phi(B)x_t = \theta(B)w_t. \tag{3.13}$$

The concise form of an ARMA model points to a potential problem in that we can unnecessarily complicate the model by multiplying both sides by another operator, say

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

without changing the dynamics. Consider the following example.

Example 3.4 Parameter Redundancy

Consider a white noise process $x_t = w_t$; in this case $\phi(B) = \theta(B) = 1$. Now, pick an arbitrary operator, say $\eta(B) = 1 - .5B$, and apply it on both sides:

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

Simplifying, we get

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

or

$$x_t = .5x_{t-1} - .5w_{t-1} + w_t, \tag{3.14}$$

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 (3.14)], but we have hidden the fact that x_t is white noise because of the parameter redundancy or over-parameterization.

Check that this is a "disguised" from of white noise. A case of parameter redundancy.

Example 3.4 points out the need to be careful when fitting ARMA models to data. Unfortunately, *it is easy to fit an overly complex ARMA model to data*. For example, if a process is truly white noise, it is possible to fit a significant ARMA(k, k) model to the data. Consider the following example.

Example 3.5 Parameter Redundancy (cont)

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

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

Thus, letting $y_t = x_t - 5.05$ be the mean centered process, the fitted model looks like

$$(1 + .96B)y_t = (1 + .95B)w_t\,,$$

which we should recognize as an over-parametrized model.

Henceforth, we will require an ARMA model to have no common factors, so that it is reduced to its simplest form. In addition, for the purposes of estimation and forecasting, we will require an ARMA model to be causal (or non-anticipative) and invertible as defined below.

We will require the ARMA model to be CAUSAL and INVERTIBLE

Definition 3.4 Causality and Invertibility *Consider an ARMA(p, q) model,*

$$\phi(B)x_t = \theta(B)w_t\,.$$

The **causal form** *of the model is given by*

$$x_t = \phi(B)^{-1}\theta(B)w_t = \psi(B)w_t = \sum_{j=0}^{\infty}\psi_j w_{t-j}, \qquad (3.15)$$

where $\psi(B) = \sum_{j=0}^{\infty}\psi_j B^j$ *(*$\psi_0 = 1$*) and assuming* $\phi(B)^{-1}$ *exists. When it does exist, then* $\phi(B)^{-1}\phi(B) = 1$*; also the parameters* ψ_j *may be obtained by matching coefficients of B in*

$$\phi(B)\psi(B) = \theta(B)\,.$$

Note that θ(B) and ϕ(B) are OPERATORS nor polynomials. B is a backshift operator

The **invertible form** *of the model is given by*

$$w_t = \theta(B)^{-1}\phi(B)x_t = \pi(B)x_t = \sum_{j=0}^{\infty}\pi_j x_{t-j}. \qquad (3.16)$$

Note from here that a MA time series is equivalent with an AR of infinite order.

where $\pi(B) = \sum_{j=0}^{\infty}\pi_j B^j$ *(*$\pi_0 = 1$*) assuming* $\theta(B)^{-1}$ *exists. Likewise, the parameters* π_j *may be obtained by matching coefficients of B in*

$$\phi(B) = \pi(B)\theta(B)\,.$$

Remark 3.1 Why are Causality and Invertibility Important? Causality requires that the present value of the time series, x_t, does not depend on the future (otherwise, forecasting would be futile). Invertibility requires that the present shock, w_t, does not depend on the future.

Not all models meet the requirements of causality and invertibility.

Property 3.1 Causality and Invertibility (existence)

Let

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \quad \text{and} \quad \theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$$

These are polynomials. z gets values in the complex plane.

be the AR and MA polynomials obtained by replacing the backshift operator B in (3.3) and (3.9) by a complex number z.

An ARMA(p,q) model is **causal** if and only if $\phi(z) \neq 0$ for $|z| \leq 1$. *The coefficients of the linear process given in (3.15) can be determined by solving* $(\psi_0 = 1)$

Causal and Stable

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

The poles that is the roots of the denominators are OUTSIDE the unit circle

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 (3.16) can be determined by solving* $(\pi_0 = 1)$

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

The poles that is the roots of denominator are outside the unit circle

We demonstrate the property in the following examples.

Example 3.6 AR(1) Redux

In Example 3.1 we saw that the AR(1) model $x_t = \phi x_{t-1} + w_t$, or

$$(1 - \phi B)x_t = w_t$$

has the causal representation

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

provided that $|\phi| < 1$. And if $|\phi| < 1$, the AR polynomial

$$\phi(z) = 1 - \phi z$$

The AR polynomial has an inverse

has an inverse

$$\frac{1}{\phi(z)} = \sum_{j=0}^{\infty} \phi^j z^j \qquad |z| \leq 1.$$

We see immediately that $\psi_j = \phi^j$. In addition, the root of $\phi(z) = 1 - \phi z$ is $z_0 = 1/\phi$ and we see that $|z_0| > 1$ if and only if $|\phi| < 1$.

Clearly, the use of the polynomial relationships stated in Property 3.1 is much easier to work with than trying to iterate backward as was done in footnote 1 (considering that the bookkeeping would be unbearable for more complicated models).

* $\phi(z)$ can't be zero in here... you wouldn't want to divide by zero, would you?

† $\theta(z)$ can't be zero in here.

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 notice that

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

and

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

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

$$x_t = .9x_{t-1} + .5w_{t-1} + w_t. \tag{3.17}$$

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

$$(1 - .9z)(1 + \psi_1 z + \psi_2 z^2 + \cdots + \psi_j z^j + \cdots) = 1 + .5z.$$

Rearranging, we get

$$1 + (\psi_1 - .9)z + (\psi_2 - .9\psi_1)z^2 + \cdots + (\psi_j - .9\psi_{j-1})z^j + \cdots = 1 + .5z.$$

The coefficients of z on the left and right sides must be the same, so we get $\psi_1 - .9 = .5$ or $\psi_1 = 1.4$, and $\psi_j - .9\psi_{j-1} = 0$ for $j > 1$. Thus, $\psi_j = 1.4(.9)^{j-1}$ for $j \geq 1$ and (3.17) can be written as

$$x_t = w_t + 1.4\sum_{j=1}^{\infty} .9^{j-1} w_{t-j}. \qquad \text{MA of infinite order}$$

The values of ψ_j may be calculated in R as follows:

```
ARMAtoMA(ar = .9, ma = .5, 10)  # first 10 psi-weights
 [1] 1.40 1.26 1.13 1.02 0.92 0.83 0.74 0.67 0.60 0.54
```

The invertible representation using Property 3.1 is obtained by matching coefficients in $\theta(z)\pi(z) = \phi(z)$,

$$(1 + .5z)(1 + \pi_1 z + \pi_2 z^2 + \pi_3 z^3 + \cdots) = 1 - .9z.$$

In this case, the π-weights are given by $\pi_j = (-1)^j 1.4\,(.5)^{j-1}$, for $j \geq 1$, and hence, we can also write (3.17) as

$$x_t = 1.4\sum_{j=1}^{\infty} (-.5)^{j-1} x_{t-j} + w_t. \qquad \text{AR of infinite order}$$

The values of π_j may be calculated using `astsa` as follows:

```
ARMAtoAR(ar = .9, ma = .5, 10)   # first 10 pi-weights
 [1] -1.400  0.700 -0.350  0.175 -0.088  0.044 -0.022  0.011 -0.005  0.003
```

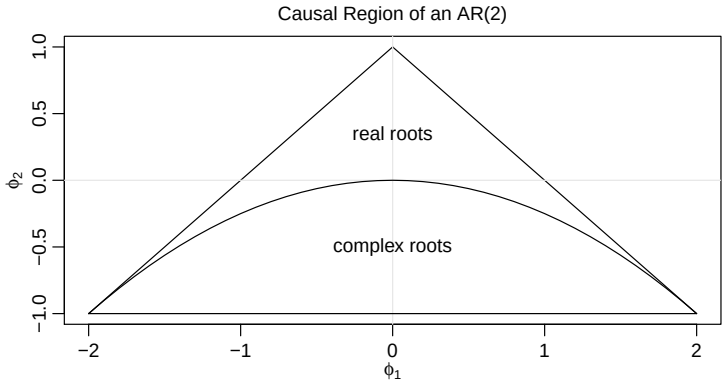


Fig. 3.3. *Causal region for an AR(2) in terms of the parameters.*

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, we must have $\phi(z) \neq 0$ for $|z| \leq 1$. If we solve $\phi(z) = 1-\phi z = 0$, we find that the root (or zero) occurs at $z_0 = 1/\phi$, so that $|z_0| > 1$ is equivalent to $|\phi| < 1$. In this case it's easy to relate parameter conditions to root conditions.

The AR(2) model, $(1-\phi_1 B - \phi_2 B^2)x_t = w_t$, is causal when the two roots of $\phi(z) = 1-\phi_1 z - \phi_2 z^2$ lie outside of the unit circle. That is, if z_1 and z_2 are the roots, then $|z_1| > 1$ and $|z_2| > 1$. Using the quadratic formula, this requirement can be written as

$$\left| \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2} \right| > 1.$$

The roots of $\phi(z)$ may be real and distinct, real and equal, or a complex conjugate pair. In terms of the coefficients, the equivalent condition is

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1, \quad \text{and} \quad |\phi_2| < 1\,, \tag{3.18}$$

which is not all that easy to show! This causality condition specifies a triangular region in the parameter space; see Figure 3.3.

3.3 Autocorrelation and Partial Autocorrelation

We begin by exhibiting the ACF of an MA(q) process.

Example 3.9 ACF of an MA(q)

The model is $x_t = \theta(B)w_t$, 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

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

$$= \begin{cases} \sigma_w^2 \sum_{j=0}^{q-h}\theta_j\theta_{j+h}, & 0 \le h \le q \\ 0 & h > q. \end{cases} \tag{3.19}$$

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 (3.19) by $\gamma(0)$ yields the ACF of an MA(q):

$$\rho(h) = \begin{cases} \dfrac{\sum_{j=0}^{q-h}\theta_j\theta_{j+h}}{1+\theta_1^2+\cdots+\theta_q^2} & 1 \le h \le q \\ 0 & h > q. \end{cases} \tag{3.20}$$

Example 3.10 ACF of an AR(p) and ARMA(p, q)

For an AR(p) or ARMA(p, q) model, $\phi(B)x_t = \theta(B)w_t$, write it as

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

or

$$x_t = \sum_{j=0}^{\infty}\psi_j w_{t-j}. \tag{3.21}$$

Note that an AR model is equivalent to a MA of an INFINITE order

It follows immediately that $E(x_t) = 0$. Also, the autocovariance function of x_t can be written as

$$\gamma(h) = \text{cov}(x_{t+h}, x_t) = \sigma_w^2\sum_{j=0}^{\infty}\psi_j\psi_{j+h}, \quad h \ge 0, \tag{3.22}$$

which is similar to the calculation in (3.6). The ACF is given by

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \frac{\sum_{j=0}^{\infty}\psi_j\psi_{j+h}}{\sum_{j=0}^{\infty}\psi_j^2}, \quad h \ge 0. \tag{3.23}$$

Unlike the MA(q), the ACF of an AR(p) or an ARMA(p, q) does not cut off at any lag, so using the ACF to help identify the order of an AR or ARMA is difficult. Also, (3.23) is not appealing in that it provides little information about the appearance of the ACF of various models.

Example 3.11 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:

$$E(x_t x_{t-h}) = \phi_1 E(x_{t-1}x_{t-h}) + \phi_2 E(x_{t-2}x_{t-h}) + E(w_t x_{t-h}).$$

Considering h>0, at this stage, makes the last term on the right hand side zero

The result is

$$\gamma(h) = \phi_1\gamma(h-1) + \phi_2\gamma(h-2), \quad h = 1, 2, \ldots . \tag{3.24}$$

In (3.24), we used the fact that $E(x_t) = 0$ and for $h > 0$, $E(w_t\, x_{t-h}) = 0$ because, by causality, x_{t-h} does not depend on future errors. Divide (3.24) through by $\gamma(0)$ to obtain a recursion for the ACF:

$$\rho(h) - \phi_1\rho(h-1) - \phi_2\rho(h-2) = 0, \quad h = 1, 2, \ldots . \tag{3.25}$$

This is a difference equation and it is solved via its CHARACTERISTIC polynomial. Since it is a second order, we need two initial conditions, ρ(0)=1 and ρ(1).

The initial conditions are $\rho(0) = 1$ and $\rho(-1) = \phi_1/(1-\phi_2)$, which is obtained by evaluating (3.25) for $h = 1$ and noting that $\rho(1) = \rho(-1)$.

Equations such as (3.25) are called difference equations, and the solutions are fairly simple expressions. First, the polynomial associated with (3.25) is

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

where the power of z is the power of the backshift, B; i.e., (3.25) is $(1 - \phi_1 B - \phi_2 B^2)\rho(h) = 0$. In general, z is a complex number. Let z_1 and z_2 be the roots (or zeros) of the associated polynomial, i.e., $\phi(z_1) = \phi(z_2) = 0$. For a causal model, the roots are outside the unit circle: $|z_1| > 1$ and $|z_2| > 1$. Now, consider the solutions:

(i) When z_1 and z_2 are distinct, then

$$\rho(h) = c_1 z_1^{-h} + c_2 z_2^{-h},$$

Distinct and real roots

so $\rho(h) \to 0$ exponentially fast as $h \to \infty$. The constants c_1 and c_2 are obtained by solving for them using the initial conditions given above. For example, when $h = 0$, we have $1 = c_1 + c_2$, and so on.

(ii) When $z_1 = z_2\ (= z_0)$ are equal (and hence real), then

$$\rho(h) = z_0^{-h}(c_1 + c_2 h),$$

Single (real) root

so $\rho(h) \to 0$ exponentially fast as $h \to \infty$.

In case (i) with complex roots, $z_2 = \bar{z}_1$ are a complex conjugate pair, and $c_2 = \bar{c}_1$ [because $\rho(h)$ is real], and

$$\rho(h) = c_1 z_1^{-h} + \bar{c}_1 \bar{z}_1^{-h}.$$

Two complex congugate roots

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

$$\rho(h) = a|z_1|^{-h}\cos(h\theta + b),$$

This is most interesting. Although we build up a random process by adding every time, as input, a sample from a white noise, the resulting time series has an in-built sinusoidal structure, albeit it is exponentially decreasing.

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 Example 3.12.

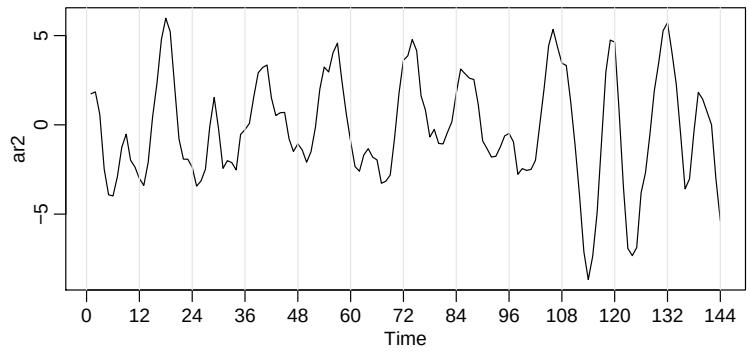


Fig. 3.4. *Simulated AR(2) model, $n = 144$ with $\phi_1 = 1.5$ and $\phi_2 = -.75$.*

Example 3.12 An AR(2) with Complex Roots

Figure 3.4 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 Figure 3.5. To calculate the roots of the polynomial and solve for arg:

```
z = c(1,-1.5,.75)      # coefficients of the polynomial
(a = polyroot(z)[1])   # print one root: 1+0.57735i = 1 + i/sqrt(3)
arg = Arg(a)/(2*pi)    # arg in cycles/pt
1/arg                  # = 12,  the pseudo period
```

To reproduce Figure 3.4:

```
set.seed(8675309)
ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
plot(ar2, axes=FALSE, xlab="Time")
axis(2);  axis(1, at=seq(0,144,by=12));  box()
abline(v=seq(0,144,by=12), lty=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)
```

In general, the behavior of the ACF of an AR(p) or an ARMA(p,q) when $p \geq 2$ will be similar to the AR(2) case. When $p = 1$, the behavior is like the AR(1) case.

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 $|\phi| < 1$. Using the theory of difference equations, we can show that the ACF is given by

$$\rho(h) = \frac{(1+\theta\phi)(\phi+\theta)}{1+2\theta\phi+\theta^2}\,\phi^{h-1}, \quad h \geq 1. \tag{3.26}$$

Notice that the general pattern of $\rho(h)$ in (3.26) is not different from that of an AR(1) given in (3.7). 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)

In (3.20), we saw that for MA(q) models, the ACF will be zero for lags greater than q. Moreover, because $\theta_q \neq 0$, the ACF will not be zero at lag q. Thus, the ACF provides a considerable amount of information about the order of the dependence when the process is a moving average process.

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

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

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

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

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

$$\begin{aligned}\gamma_x(2) = \text{cov}(x_t, x_{t-2}) &= \text{cov}(\phi x_{t-1} + w_t, x_{t-2}) \\ &= \text{cov}(\phi^2 x_{t-2} + \phi w_{t-1} + w_t, x_{t-2}) = \phi^2 \gamma_x(0).\end{aligned}$$

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 partialling out) the effect of x_{t-1}. That is, we consider the correlation between $x_t - \phi x_{t-1}$ and $x_{t-2} - \phi x_{t-1}$, because it is the correlation between x_t and x_{t-2} with the linear dependence of each on x_{t-1} removed. In this way, we have broken the dependence chain between x_t and x_{t-2}. In fact,

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

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.

Note that this dependence is through to x_{t-1}

Definition 3.5 *The* **partial autocorrelation function (PCF)** *of a stationary process, x_t, denoted ϕ_{hh}, for $h = 1, 2, \ldots$, is*

$$\phi_{11} = \text{corr}(x_1, x_0) = \rho(1) \tag{3.27}$$

and

$$\phi_{hh} = \text{corr}(x_h - \hat{x}_h,\ x_0 - \hat{x}_0), \quad h \geq 2, \tag{3.28}$$

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

Note that due to stationarity, we can replace the time index h with t +h and the the index 0 with t.

Note that the samples between x_h and x_0 are, x_1, x_2, ..., x_{h-1}.

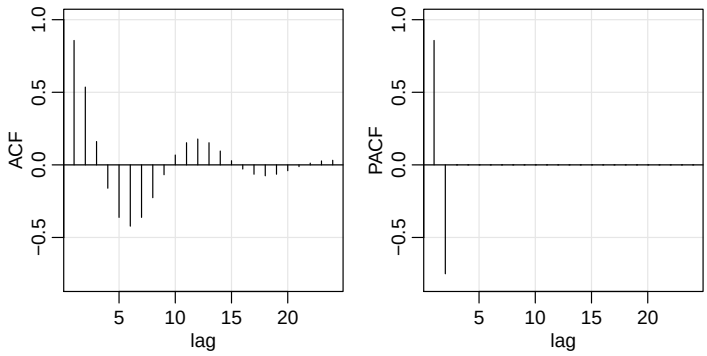


Fig. 3.5. *The ACF and PACF of an AR(2) model with* $\phi_1 = 1.5$ *and* $\phi_2 = -.75$.

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

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

Example 3.14 The PACF of an AR(p)

The model can be written as

$$x_{t+h} = \sum_{j=1}^{p} \phi_j x_{t+h-j} + w_{t+h},$$

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

$$\hat{x}_{t+h} = \sum_{j=1}^{p} \phi_j x_{t+h-j}.$$

Note that the error we commit in using this predictor is equal to the noise that generates the time series. Thus, the error variance is equal to the noise variance. However, this is the best we can hope to do in predicting from past samples, if the noise samples are i.i.d.

Although we have not proved this result, it should be obvious that it is so. Thus, when $h > p$,

$$\phi_{hh} = \text{corr}(x_{t+h} - \hat{x}_{t+h},\, x_t - \hat{x}_t) = \text{corr}(w_{t+h},\, x_t - \hat{x}_t) = 0,$$

because, by causality, $x_t - \hat{x}_t$ depends only on $\{w_{t+h-1}, w_{t+h-2}, \ldots\}$. 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 3.5 shows the ACF and the PACF of the AR(2) model presented in Example 3.12. To reproduce Figure 3.5 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))
tsplot(ACF, type="h", xlab="lag", ylim=c(-.8,1))
abline(h=0)
tsplot(PACF, type="h", xlab="lag", ylim=c(-.8,1))
abline(h=0)
```

Note that when regressing x_t on FUTURE samples, x_{t+1},..., x_{t+h-1}, the noise samples that are involved in the regression are w_{t+h-1}, w_{t+h-2}, ... This is by the equivalent causal model, where a sample at any time index, t, i.e., x_t, depends only on noise samples at the current and previous time indices ONLY.

We also have the following large sample result for the PACF, which may be compared to the similar result for the ACF given in Property 1.3.

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

Property 3.2 Large Sample Distribution of the PACF

If the time series is a causal AR(p) process and the sample size n is large, then $\sqrt{n}\,\hat{\phi}_{hh}$ is approximately $N(0,1)$, for $h > p$. This result also holds for $p = 0$, wherein the process is white noise.

Example 3.15 The PACF of an MA(q)

A MA model is equivalent to an AR of INFINITE order

For an 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 $|\theta| < 1$, it can be shown that

$$\phi_{hh} = -\frac{(-\theta)^h(1-\theta^2)}{1-\theta^{2(h+1)}}, \quad h \geq 1.$$

We do not have to compute the PACF by performing numerous regressions first. The computations are done via a recursive formula called the Durbin–Levinson algorithm.

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.

Example 3.16 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 3.6 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 Section 3.4, we ran a regression (see Section 2.1) using the data triplets $\{(x; z_1, z_2) : (x_3; x_2, x_1), (x_4; x_3, x_2), \ldots, (x_{453}; x_{452}, x_{451})\}$ to fit the model

$$x_t = \phi_0 + \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

for $t = 3, 4, \ldots, 453$. The values of the estimates were $\hat{\phi}_0 = 6.74_{(1.11)}$, $\hat{\phi}_1 = 1.35_{(.04)}$, $\hat{\phi}_2 = -.46_{(.04)}$, and $\hat{\sigma}_w^2 = 89.72$, where the estimated standard errors are in parentheses.

The following R code can be used for this analysis. We use the script `acf2` from `astsa` to print and plot the ACF and PACF.

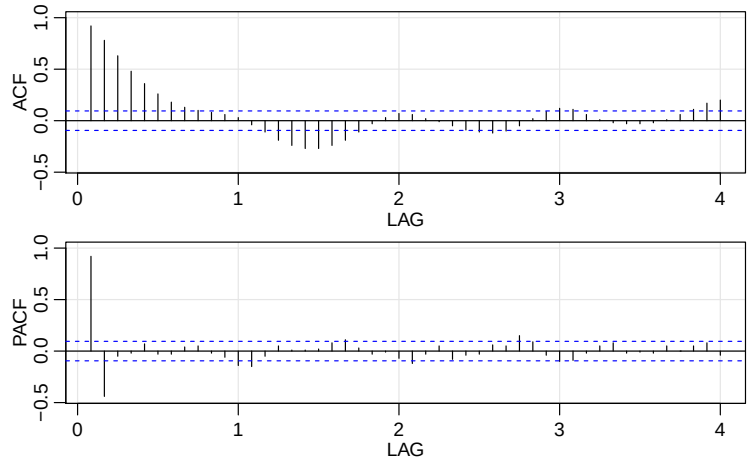


Fig. 3.6. *ACF and PACF of the Recruitment series. Note that the lag axes are in terms of season (12 months in this case).*

```
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
```

3.4 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, $E(x_t^k)$, to sample moments, $\frac{1}{n}\sum_{t=1}^n x_t^k$, for $k = 1, 2, \ldots$, and then solving for the parameters in terms of the sample moments. We immediately see that, if $E(x_t) = \mu$, 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),

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

similar to Example 3.11, we have the following result:

Definition 3.6 *The* **Yule–Walker equations** *are given by*

$$\rho(h) = \phi_1 \rho(h-1) + \cdots + \phi_p \rho(h-p), \quad h = 1, 2, \ldots, p, \tag{3.29}$$

$$\sigma_w^2 = \gamma(0)\,[1 - \phi_1 \rho(1) - \cdots - \phi_p \rho(p)]. \tag{3.30}$$

The estimators obtained by replacing $\gamma(0)$ with its estimate, $\hat{\gamma}(0)$ and $\rho(h)$ with its estimate, $\hat{\rho}(h)$, are called the *Yule–Walker estimators*. For AR(p) models, if the sample size is large, the Yule–Walker estimators are approximately normally distributed, and $\hat{\sigma}_w^2$ is close to the true value of σ_w^2.

Example 3.17 Yule–Walker Estimation for an AR(1)

For an AR(1), $(x_t - \mu) = \phi(x_{t-1} - \mu) + w_t$, the mean estimate is $\hat{\mu} = \bar{x}$, and (3.29) is

$$\rho(1) = \phi\rho(0) = \phi\,,$$

so

$$\hat{\phi} = \hat{\rho}(1) = \frac{\sum_{t=1}^{n-1}(x_{t+1} - \bar{x})(x_t - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}\,,$$

as expected. The estimate of the error variance is then

$$\hat{\sigma}_w^2 = \hat{\gamma}(0)\,[1 - \hat{\phi}^2]\,;$$

recall $\gamma(0) = \sigma_w^2/(1-\phi^2)$ from (3.6).

Example 3.18 Yule–Walker Estimation of the Recruitment Series

In Example 3.16 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 close to the regression values in Example 3.16.

```
rec.yw = ar.yw(rec, order=2)
rec.yw$x.mean     # mean estimate
 [1] 62.26278
rec.yw$ar         # parameter estimates
 [1]  1.3315874 -0.4445447
sqrt(diag(rec.yw$asy.var.coef))  # their standard errors
 [1] 0.04222637 0.04222637
rec.yw$var.pred  # error variance estimate
 [1] 94.79912
```

In the case of AR(p) models, the Yule–Walker estimators are optimal estimators, but this is not true for MA(q) or ARMA(p,q) models. AR(p) models are linear models, and the Yule–Walker estimators are essentially least squares estimators. MA or ARMA models are nonlinear models, so this technique does not give optimal estimators.

Example 3.19 Method of Moments Estimation for an MA(1)

Consider the MA(1) model, $x_t = w_t + \theta w_{t-1}$, where $|\theta| < 1$. The model can then be written as

$$x_t = -\sum_{j=1}^{\infty}(-\theta)^j x_{t-j} + w_t,$$

which is nonlinear in θ. The first two population autocovariances are $\gamma(0) = \sigma_w^2(1+\theta^2)$ and $\gamma(1) = \sigma_w^2\theta$, so the estimate of θ is found by solving:

$$\hat{\rho}(1) = \frac{\hat{\gamma}(1)}{\hat{\gamma}(0)} = \frac{\hat{\theta}}{1+\hat{\theta}^2}.$$

This is a quadratic equations with two roots.

Two solutions exist, so we would pick the invertible one. If $|\hat{\rho}(1)| \le \frac{1}{2}$, the solutions are real, otherwise, a real solution does not exist. Even though

$|\rho(1)| < \frac{1}{2}$ for an invertible MA(1), it may happen that $|\hat{\rho}(1)| \geq \frac{1}{2}$ because it is an estimator. For example, the following simulation in R produces a value of $\hat{\rho}(1) = .507$ 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)
```

The preferred method of estimation is maximum likelihood estimation (MLE), which determines the values of the parameters that are most *likely* to have produced the observations. For normal models, this is the same as weighted least squares. For ease, we first discuss conditional least squares.

3.4.1 Conditional Least Squares

Recall from Chapter 2, in simple linear regression, $x_t = \beta_0 + \beta_1 z_t + w_t$, we minimize

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

with respect to the βs. This is a simple problem because we have all the data pairs, (z_t, x_t) for $t = 1, \ldots, n$. For ARMA models, we do not have this luxury.

First, we focus on conditional least squares for ARMA(p,q) models via Gauss–Newton. Write the model parameters as $\beta = (\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q)$, and for the ease of discussion, we will put $\mu = 0$.

Write the ARMA model in terms of the errors

$$w_t(\beta) = x_t - \sum_{j=1}^{p} \phi_j x_{t-j} - \sum_{k=1}^{q} \theta_k w_{t-k}(\beta), \tag{3.31}$$

Because the coefficiants are unknowns, we write the noise as a function of the unknown coefficents.

emphasizing the dependence of the errors on the parameters (recall that $w_t = \sum_{j=0}^{\infty} \pi_j x_{t-j}$ by invertibilty, and the π_j are complicated functions of β).

We don't observe the x_t for $t \leq 0$, nor the errors w_t, which is what makes this harder than simple linear regression. For *conditional least squares*, we condition on $x_1, \ldots, x_p$ (if $p > 0$) and set $w_p = w_{p-1} = w_{p-2} = \cdots = w_{p+1-q} = 0$ (if $q > 0$), in which case, given β, we may evaluate (3.31) for $t = p+1, \ldots, n$. For example, for an ARMA$(1,1)$, $x_t = \phi x_{t-1} + \theta w_{t-1} + w_t$, we would start at $p+1 = 2$ and set $w_1 = 0$ so that

$$\begin{aligned} w_2 &= x_2 - \phi x_1 - \theta w_1 = x_2 - \phi x_1 \\ w_3 &= x_3 - \phi x_2 - \theta w_2 \\ &\vdots \\ w_n &= x_n - \phi x_{n-1} - \theta w_{n-1} \end{aligned}$$

observe that w_3 depends nonlinearly on θ and ϕ, because w_2 depends on ϕ

Given data, we can evaluate these errors at any values of the parameters; e.g., $\phi = \theta = .5$. Using this conditioning argument, the conditional error sum of squares is

$$S_c(\beta) = \sum_{t=p+1}^{n} w_t^2(\beta). \tag{3.32}$$

Minimizing $S_c(\beta)$ with respect to β yields the conditional least squares estimates. We could use a brute force method where we evaluate $S_c(\beta)$ over a grid of possible values for the parameters and choose the values with the smallest error sum of squares, but this method becomes prohibitive if there are many parameters.

If $q = 0$, the problem is linear regression and no iterative technique is needed to minimize $S_c(\phi_1, \ldots, \phi_p)$. For example, for a zero-mean AR(1), $x_t = \phi x_{t-1} + w_t$, the conditional sum of squares is

$$S_c(\phi) = \sum_{t=2}^{n} w_t^2(\phi) = \sum_{t=2}^{n} (x_t - \phi x_{t-1})^2.$$

Note that we have to start at $t = 2$ because x_0 is not observed. The conditional least squares estimate of ϕ follows from simple linear regression wherein,

$$\hat{\phi} = \frac{\sum_{t=2}^{n} x_t x_{t-1}}{\sum_{t=2}^{n} x_{t-1}^2},$$

which is nearly $\hat{\rho}(1)$ [the denominator does not include x_n^2].

If $q > 0$, the problem becomes nonlinear regression and we will rely on numerical optimization. Gauss–Newton uses an iterative method for solving the problem of minimizing (3.32). We demonstrate the method for an MA(1).

Example 3.20 Gauss–Newton for an MA(1)

Consider an MA(1) process, $x_t = w_t + \theta w_{t-1}$. Write the truncated errors as

$$w_t(\theta) = x_t - \theta w_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{3.33}$$

where we condition on $w_0(\theta) = 0$. Our goal is to find the value of θ that minimizes $S_c(\theta) = \sum_{t=1}^{n} w_t^2(\theta)$, which is a nonlinear function of θ. Gauss–Newton is an iterative procedure based on local linear approximations of S_c.

Let $\theta_{(0)}$ be an initial estimate of θ. For example, we could use method of moments. The first-order Taylor expansion[3] of $w_t(\theta)$ at $\theta_{(0)}$ is

$$w_t(\theta) \approx w_t(\theta_{(0)}) - \left(\theta - \theta_{(0)}\right) z_t(\theta_{(0)}), \tag{3.34}$$

Since w_t(θ) is a non linear function,we LINEARIZE it unsing Taylor's expansion

where

$$z_t(\theta_{(0)}) = -\frac{\partial w_t(\theta)}{\partial \theta}\bigg|_{\theta=\theta_{(0)}}.$$

Taking derivatives in (3.33),

$$\frac{\partial w_t(\theta)}{\partial \theta} = -w_{t-1}(\theta) - \theta \frac{\partial w_{t-1}(\theta)}{\partial \theta}, \quad t = 1, \ldots, n, \tag{3.35}$$

where $\partial w_0(\theta)/\partial\theta = 0$. Using the notation of (3.34), we can also write (3.35) as

$$z_t(\theta) = w_{t-1}(\theta) - \theta z_{t-1}(\theta), \quad t = 1, \ldots, n, \tag{3.36}$$

Observe that the derivative sequence is an AR one

[3] Newton's method and Taylor expansion (links to WikiBooks K-12 calculus book).

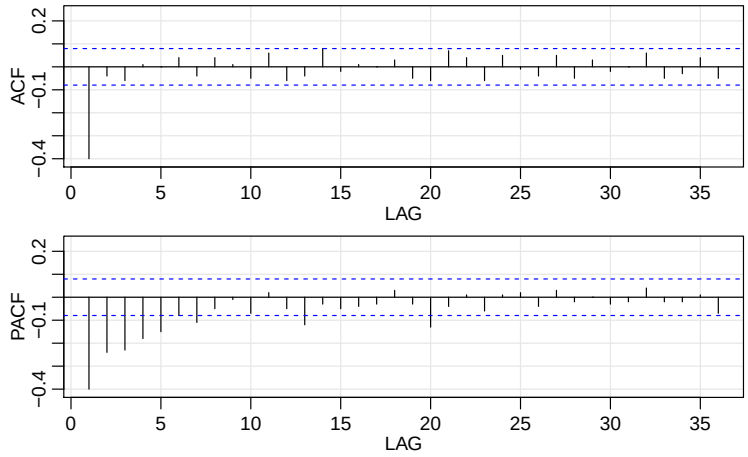


Fig. 3.7. *ACF and PACF of transformed glacial varves.*

where $z_0(\theta) = 0$. This implies that the derivative sequence is an AR process, which we may easily compute given a value of θ.

The linear approximation of $S_c(\theta)$ is found by replacing $w_t(\theta)$ by its linear approximation in (3.34),

$$Q(\theta) = \sum_{t=1}^{n} \Big[\underbrace{w_t(\theta_{(0)})}_{y_t} - \underbrace{(\theta - \theta_{(0)})}_{\beta} \underbrace{z_t(\theta_{(0)})}_{z_t} \Big]^2 \tag{3.37}$$

and this is the quantity that we will minimize. The problem is now simple linear regression ("$y_t = \beta z_t + \epsilon_t$"), so that

$$\widehat{(\theta - \theta_{(0)})} = \textstyle\sum_{t=1}^{n} z_t(\theta_{(0)}) w_t(\theta_{(0)}) \;/\; \sum_{t=1}^{n} z_t^2(\theta_{(0)}),$$

or

$$\hat{\theta} = \theta_{(0)} + \textstyle\sum_{t=1}^{n} z_t(\theta_{(0)}) w_t(\theta_{(0)}) \;/\; \sum_{t=1}^{n} z_t^2(\theta_{(0)}).$$

Consequently, the Gauss–Newton procedure in this case is, on iteration $j+1$, set

$$\theta_{(j+1)} = \theta_{(j)} + \frac{\sum_{t=1}^{n} z_t(\theta_{(j)}) w_t(\theta_{(j)})}{\sum_{t=1}^{n} z_t^2(\theta_{(j)})}, \quad j = 0, 1, 2, \ldots, \tag{3.38}$$

where the values in (3.38) are calculated recursively using (3.33) and (3.36). 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 3.21 Fitting the Glacial Varve Series

Consider the glacial varve series, for $n = 634$ years, analyzed in Example 2.7 and in Problem 2.6, where it was argued that a first-order moving average model might fit the logarithmically transformed and differenced varve series, say,

$$\nabla \log(x_t) = \log(x_t) - \log(x_{t-1}) = \log\left(\frac{x_t}{x_{t-1}}\right),$$

which can be interpreted as being approximately the percentage change in the thickness.

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.

Since $\hat{\rho}(1) = -.397$, our initial estimate is

$$\theta_{(0)} = \frac{1 - \sqrt{1 - 4\hat{\rho}(1)^2}}{2\hat{\rho}(1)} = -.495$$

using Example 3.19.

The results of eleven iterations of the Gauss–Newton procedure, (3.38), starting with $\theta_{(0)}$ are given in Table 3.2. The final estimate is

$$\hat{\theta} = \theta_{(11)} = -.773\,,$$

which results in

$$S_c(-.773) = 148.98\,.$$

Interim values and the corresponding value of the conditional sum of squares, $S_c(\theta)$ given in (3.32), are also displayed in the table. The final estimate of the error variance is

$$\hat{\sigma}_w^2 = \frac{148.98}{632} = .236$$

with 632 degrees of freedom (one is lost in differencing). The value of the sum of the squared derivatives at convergence is

$$\sum_{t=1}^{n} z_t^2(\theta_{(11)}) = 368.741\,,$$

and consequently, the estimated standard error of $\hat{\theta}$ is[4]

$$\mathrm{SE}(\hat{\theta}) = \sqrt{.236/368.741} = .025\,;$$

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.

The following code was used in this example.

```
x = diff(log(varve))                  # data
r = acf(x, lag=1, plot=FALSE)$acf[-1]  # acf(1)
c(0) -> w -> z                        # initialize
c() -> Sc -> Sz -> Szw -> SS -> para
```

[4] To estimate the standard error, we are using the standard regression results from (2.6) as an approximation

Table 3.2. *Gauss–Newton Results for Example 3.21*

j	$\theta_{(j)}$	$S_c(\theta_{(j)})$	$\sum_{t=1}^{n} z_t^2(\theta_{(j)})$
0	−0.495	158.739	171.240
1	−0.668	150.747	235.266
2	−0.733	149.264	300.562
3	−0.756	149.031	336.823
4	−0.766	148.990	354.173
5	−0.769	148.982	362.167
6	−0.771	148.980	365.801
7	−0.772	148.980	367.446
8	−0.772	148.980	368.188
9	−0.772	148.980	368.522
10	−0.773	148.980	368.673
11	−0.773	148.980	368.741

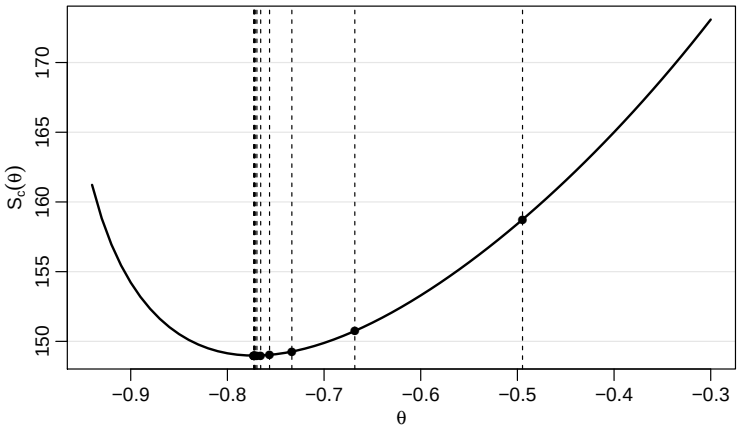


Fig. 3.8. *Conditional sum of squares versus values of the moving average parameter for the glacial varve example, Example 3.21. Vertical lines indicate the values of the parameter obtained via Gauss–Newton; see Table 3.2 for the actual values.*

```
num = length(x)
## Estimation
para[1] = (1-sqrt(1-4*(r^2)))/(2*r)      # MME
niter   = 12
for (p in 1:niter){
  for  (i in 2:num){ w[i] = x[i]  - para[p]*w[i-1]
                     z[i] = w[i-1]- para[p]*z[i-1] }
  Sc[p]     = sum(w^2)
  Sz[p]     = sum(z^2)
  Szw[p]    = sum(z*w)
  para[p+1] = para[p] + Szw[p]/Sz[p]  }
## Results
round(cbind(iteration=0:(niter-1), thetahat=para[1:niter], Sc, Sz), 3)
## Plot cond SS
th = seq(-.3,-.94,-.01)
for (p in 1:length(th)){
  for (i in 2:num){ w[i] = x[i]-th[p]*w[i-1] }
  SS[p] = sum(w^2)     }
plot(th, SS, type="l", ylab=expression(S[c](theta)),
           xlab=expression(theta))
abline(v=para[1:12], lty=2)       #  add results to plot
points(para[1:12], Sc[1:12], pch=16)
```

3.4.2 Unconditional Least Squares

Estimation of the parameters in an ARMA model is more like weighted least squares than ordinary least squares. Consider the normal regression model

$$x_t = \beta_0 + \beta_1 z_t + \epsilon_t\,,$$

where now, the errors have possibly different variances,

$$\epsilon_t \sim N(0, \sigma^2 h_t)\,.$$

In this case, we use weighted least squares to minimize

$$S(\beta) = \sum_{t=1}^{n} \frac{\epsilon_t^2(\beta)}{h_t} = \sum_{t=1}^{n} \frac{1}{h_t}\Big(x_t - [\beta_0 + \beta_1 z_t]\Big)^2$$

with respect to the βs. This problem is more difficult because the weights, $1/h_t$, are rarely known (the case $h_t = 1$ is ordinary least squares). For ARMA models, however, we do know the structure of these variances.

For ease, we'll concentrate on the AR(1) model,

$$x_t = \mu + \phi(x_{t-1} - \mu) + w_t \tag{3.39}$$

where $|\phi| < 1$ and $w_t \sim$ iid $\mathrm{N}(0, \sigma_w^2)$. Given data $x_1, x_2, \ldots, x_n$, the model (3.39) does not include a regression for the first observation because x_0 is not observed. However, we know from Example 3.1 that

$$x_1 = \mu + \epsilon_1 \qquad \epsilon_1 \sim N(0, \sigma_w^2/(1-\phi^2))\,.$$

In this case, we have $h_1 = 1/(1-\phi^2)$ and $h_t = 1$ for $t \geq 2$. Thus, the unconditional sum of squares is now

$$S(\mu, \phi) = (1-\phi^2)(x_1 - \mu)^2 + \sum_{t=2}^{n} [(x_t - \mu) - \phi(x_{t-1} - \mu)]^2\,. \tag{3.40}$$

The errors here have variance σ_w^2

In conditional least squares, we conditioned away the nasty part involving x_1 to make the problem easier. For unconditional least squares, we need to use numerical optimization even for the simple AR(1) case.

This problem generalizes in an obvious way to AR(p) models, and in a not so obvious way to ARMA models in general. At any rate, and unconditional least squares is equivalent to maximum likelihood estimation. In the general case of causal and invertible ARMA(p, q) models, maximum likelihood estimation, least squares estimation (conditional and unconditional), and Yule–Walker estimation in the case of AR models, all lead to optimal estimators.

Example 3.22 Some Specific Large Sample Distributions [5]

AR(1):

$$\hat{\phi} \sim \mathrm{AN}\left[\phi, n^{-1}(1-\phi^2)\right]. \tag{3.41}$$

AR(2):

$$\begin{pmatrix}\hat{\phi}_1\\ \hat{\phi}_2\end{pmatrix} \sim \mathrm{AN}\left[\begin{pmatrix}\phi_1\\ \phi_2\end{pmatrix}, n^{-1}\begin{pmatrix}1-\phi_2^2 & -\phi_1(1+\phi_2)\\ \text{sym} & 1-\phi_2^2\end{pmatrix}\right]. \tag{3.42}$$

[5] We write $X_n \sim \mathrm{AN}(\mu_n, \sigma_n^2)$ if, for large n, $Z_n = (X_n - \mu_n)/\sigma_n$ is approximately standard normal.

MA(1):

$$\hat{\theta} \sim \text{AN}\left[\theta, n^{-1}(1-\theta^2)\right]. \tag{3.43}$$

MA(2):

$$\begin{pmatrix}\hat{\theta}_1\\ \hat{\theta}_2\end{pmatrix} \sim \text{AN}\left[\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}, n^{-1}\begin{pmatrix}1-\theta_2^2 & \theta_1(1+\theta_2)\\ \text{sym} & 1-\theta_2^2\end{pmatrix}\right]. \tag{3.44}$$

Example 3.23 Overfitting Caveat

The large sample behavior of the parameter estimators gives us an additional insight into the problem of fitting ARMA models to data. For example, suppose a time series follows an AR(1) process and we decide to fit an AR(2) to the data. Do any problems occur in doing this? More generally, why not simply fit large-order AR models to make sure that we capture the dynamics of the process? After all, if the process is truly an AR(1), the other autoregressive parameters will not be significant. The answer is that if we overfit, we obtain less efficient, or less precise parameter estimates. For example, if we fit an AR(1) to an AR(1) process, for large n, $\text{var}(\hat{\phi}_1) \approx n^{-1}(1-\phi_1^2)$. But, if we fit an AR(2) to the AR(1) process, for large n, $\text{var}(\hat{\phi}_1) \approx n^{-1}(1-\phi_2^2) = n^{-1}$ because $\phi_2 = 0$. Thus, the variance of ϕ_1 has been inflated, making the estimator less precise.

We do want to mention, however, that overfitting can be used as a diagnostic tool. For example, if we fit an AR(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 Section 3.7.

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, $x_1, \ldots, x_n$, collected to the present. Throughout this section, we will assume that the model parameters are known. When the parameters are unknown, we replace them with their estimates.

To understand how to forecast an ARMA process, it is instructive to investigate forecasting an AR(1),

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

First, consider one-step-ahead prediction, that is, given data $x_1, \ldots, x_n$, we wish to forecast the value of the time series at the next time point, x_{n+1}. We will call the forecast x_{n+1}^n. In general, the notation x_t^n refers to what we can expect x_t to be given the data $x_1, \ldots, x_n$. Since

$$x_{n+1} = \phi x_n + w_{n+1}\,,$$

we should have

$$x_{n+1}^n = \phi x_n^n + w_{n+1}^n\,.$$

But since we know x_n (it is one of our observations), $x_n^n = x_n$, and since w_{n+1} is a future error and independent of $x_1, \ldots, x_n$, we have $w_{n+1}^n = E(w_{n+1}) = 0$. Consequently, the *one-step-ahead forecast* is

$$x_{n+1}^n = \phi x_n \,. \tag{3.45}$$

The one-step-ahead *mean squared prediction error* (MSPE) is given by

$$P_{n+1}^n = E[x_{n+1} - x_{n+1}^n]^2 = E[x_{n+1} - \phi x_n]^2 = E w_{n+1}^2 = \sigma_w^2.$$

The two-step-ahead forecast is obtained similarly. Since, by the model,

$$x_{n+2} = \phi x_{n+1} + w_{n+2}\,,$$

we should have

$$x_{n+2}^n = \phi x_{n+1}^n + w_{n+2}^n\,.$$

Again, w_{n+2} is a future error, so $w_{n+2}^n = 0$. Also, we already know $x_{n+1}^n = \phi x_n$, so the forecast is

$$x_{n+2}^n = \phi x_{n+1}^n = \phi^2 x_n\,. \tag{3.46}$$

The two-step-ahead *mean squared prediction error* (MSPE) is given by

$$\begin{aligned} P_{n+2}^n &= E[x_{n+2} - x_{n+2}^n]^2 = E[\phi x_{n+1} + w_{n+2} - \phi^2 x_n]^2 \\ &= E[w_{n+2} + \phi(x_{n+1} - \phi x_n)]^2 = E[w_{n+2} + \phi w_{n+1}]^2 = \sigma_w^2(1+\phi^2). \end{aligned}$$

Generalizing these results, it is easy to see that the m-step-ahead forecast is.

$$x_{n+m}^n = \phi^m x_n \qquad m = 1, 2, \ldots\,. \tag{3.47}$$

Also, the MSPE will is

$$P_{n+m}^n = E[x_{n+m} - x_{n+m}^n]^2 = \sigma_w^2(1 + \phi^2 + \cdots + \phi^{2(m-1)})\,. \tag{3.48}$$

Note that since $|\phi| < 1$, we will have $\phi^m \to 0$ fast as $m \to \infty$. Thus the forecasts in (3.47) will soon go to zero (or the mean) and become useless. In addition, the MSPE will converge to $\sigma_w^2 \sum_{j=0}^{\infty} \phi^{2j} = \sigma_w^2/(1-\phi^2)$, which is the variance of the process x_t; recall (3.6).

Forecasting an AR(p) model is basically the same as forecasting an AR(1) provided the sample size n is larger than the order p, which it is most of the time. Since MA(q) and ARMA(p,q) are AR(∞), the same basic techniques can be used. Because ARMA models are invertible; i.e., $w_t = x_t + \sum_{j=1}^{\infty} \pi_j x_{t-j}$, we may write

$$x_{n+m} = -\sum_{j=1}^{\infty} \pi_j x_{n+m-j} + w_{n+m}.$$

If we had the infinite history $\{x_n, x_{n-1}, \ldots, x_1, x_0, x_{-1}, \ldots\}$, of the data available, we would predict x_{n+m} by

$$x_{n+m}^n = -\sum_{j=1}^{\infty} \pi_j x_{n+m-j}^n$$

successively for $m = 1, 2, \ldots$. In this case, $x_t^n = x_t$ for $t = n, n-1, \ldots$. We only have the actual data $\{x_n, x_{n-1}, \ldots, x_1\}$ available, but a practical solution is to truncate the forecasts as

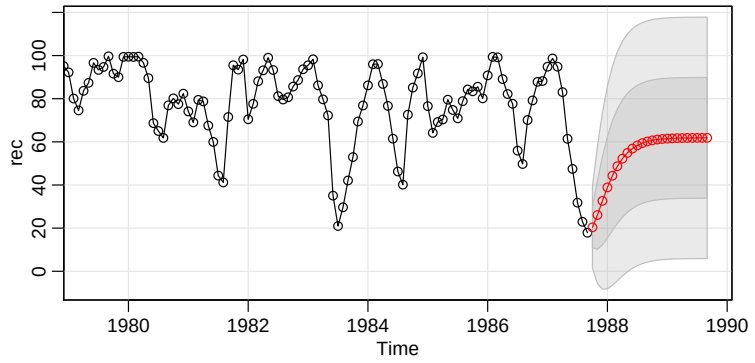


Fig. 3.9. *Twenty-four month forecasts for the Recruitment series. The actual data shown are from about January 1979 to September 1987, and then the forecasts plus and minus one and two standard error are displayed.*

$$x^n_{n+m} = -\sum_{j=1}^{n+m-1} \pi_j x^n_{n+m-j},$$

with $x^n_t = x_t$ for $1 \le t \le n$. For ARMA models in general, as long as n is large, the approximation works well because the π-weights are going to zero exponentially fast. For large n, it can be shown that the mean squared prediction error for ARMA(p,q) models is approximately (exact if $q = 0$)

$$P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^2 . \tag{3.49}$$

π^2_j

Example 3.24 Forecasting the Recruitment Series

In Example 3.16 we fit an AR(2) model to the Recruitment series using OLS. Here, we use MLE:

```
sarima(rec,2,0,0)          # fit model
 $ttable
       Estimate     SE  t.value p.value
 ar1     1.3512 0.0416  32.4933       0
 ar2    -0.4612 0.0417 -11.0687       0
 xmean  61.8585 4.0039  15.4494       0
61.8585*(1-1.3512+.4612)   # get constant (for fun)
 [1] 6.804435
```

The results are nearly the same as using OLS. Using the parameter estimates as the actual parameter values, the forecasts and root MSPEs can be calculated in a similar fashion to the introduction to this section.

Figure 3.9 shows the result of forecasting the Recruitment series over a 24-month horizon, $m = 1, 2, \ldots, 24$, obtained in R as

```
sarima.for(rec, 24, 2, 0, 0)
```

Note how the forecast levels off to the mean quickly and the prediction intervals are wide and become constant. That is, because of the short memory, the forecasts settle to the mean, 61.86, and the root MSPE becomes quite large (and eventually settles at the standard deviation of all the data).

3.6 Integrated Models

In previous chapters, we saw that if x_t is a random walk, $x_t = x_{t-1} + w_t$, then by differencing x_t, we find that $\nabla x_t = w_t$ is stationary. In many situations, time series can be thought of as being composed of two components, a nonstationary trend component and a zero-mean stationary component. For example, in Section 2.1 we considered the model

$$x_t = \mu_t + y_t, \tag{3.50}$$

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 (3.50) 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 (3.50) is quadratic, $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2$, then the differenced series $\nabla^2 y_t$ 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. The basic idea is that if differencing the data at some order d produces an ARMA process, then the original process is said to be ARIMA.

Definition 3.7 *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

$$\phi(B)(1-B)^d x_t = \theta(B) w_t. \tag{3.51}$$

Observe that the differencing applies on x_t and NOT on the noise(MA part)

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

It should be clear that, since $y_t = \nabla^d x_t$ is ARMA, we can use Section 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 Section 3.5, equation (3.49), works well. That is, the mean-squared prediction error can be approximated by

$$P^n_{n+m} = \sigma_w^2 \sum_{j=0}^{m-1} \psi_j^{*2}, \tag{3.52}$$

where ψ^*_j is the coefficient of z^j in $\psi^*(z) = \theta(z)/\phi(z)(1-z)^d$.

To better understand forecasting integrated models, we examine the properties of some simple cases.

Example 3.25 Random Walk with Drift

To fix ideas, we begin by considering the random walk with drift model first presented in Example 1.9, 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

$$x^n_{n+1} = \delta + x^n_n + w^n_{n+1} = \delta + x_n\,.$$

Zero

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

$$x^n_{n+m} = m\,\delta + x_n, \tag{3.53}$$

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

$$P^n_{n+m} = E(x_{n+m} - x^n_{n+m})^2 = E\Big(\sum_{j=n+1}^{n+m} w_j\Big)^2 = m\,\sigma_w^2. \tag{3.54}$$

Unlike the stationary case, as the forecast horizon grows, the prediction errors, (3.54), increase without bound and the forecasts follow a straight line with slope δ emanating from x_n.

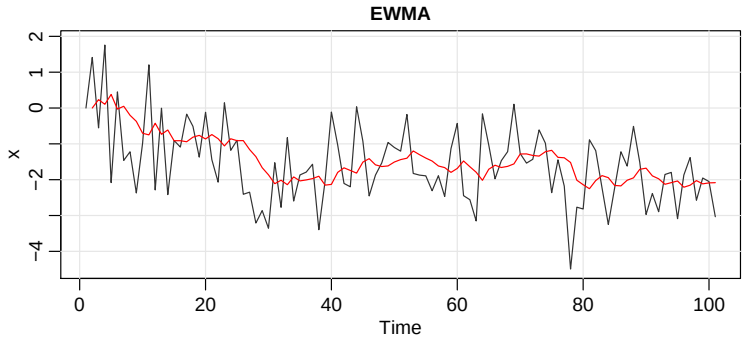


Fig. 3.10. *Output for Example 3.26: Simulated data with an EWMA superimposed.*

Example 3.26 IMA$(1,1)$ and EWMA

The ARIMA(0,1,1), or IMA(1,1) model is of interest because many economic time series can be successfully modeled this way. The model leads to a frequently used forecasting method called exponentially weighted moving averages (EWMA). We will write the model as

$$x_t = x_{t-1} + w_t - \lambda w_{t-1}, \tag{3.55}$$

Indeed, the difference is a MA(1,1)

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 (3.55), 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 (3.55) 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

$$x_t = \sum_{j=1}^{\infty} (1-\lambda)\lambda^{j-1} x_{t-j} + w_t. \tag{3.56}$$

as an approximation for large t (put $x_t = 0$ for $t \leq 0$). Verification of (3.56) is left to the reader (Problem 3.11). Using the approximation (3.56), we have that the approximate one-step-ahead predictor is

$$x_{n+1}^n = (1-\lambda)x_n + \lambda x_n^{n-1}, \tag{3.57}$$

Linear Combination of the OLD forcast and the NEW observation.

because $x_n^{n-1} = \sum_{j=1}^{\infty}(1-\lambda)\lambda^{j-1}x_{n-j}$ and $w_{n+1}^n = 0$. From (3.57), we see that the new forecast is a linear combination of the old forecast and the new observation. The mean-square prediction error can be approximated using (3.52) 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, (3.52) leads to

$$P_{n+m}^n \approx \sigma_w^2[1 + (m-1)(1-\lambda)^2].$$

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. 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). This and related techniques are generally called *exponential smoothing*; the ideas were made popular in the late 1950s and are still used today by old men who smell bad because they are simple and easy to use. To reproduce Figure 3.10, use the following.

```
set.seed(666)
x = arima.sim(list(order = c(0,1,1), ma = -0.8), n = 100)
(x.ima = HoltWinters(x, beta=FALSE, gamma=FALSE))  # α below is 1 − λ
  Smoothing parameter:  alpha:   0.1663072
plot(x.ima, main='EWMA')
```

3.7 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 (2.35), could be employed. Also, the particular application might suggest an appropriate transformation. For example, we have seen numerous examples where the data behave as $x_t = (1 + p_t)x_{t-1}$, where p_t is a small percentage change from period $t - 1$ to t, which may be negative. If p_t is a relatively stable process, then $\nabla \log(x_t) \approx p_t$ will be relatively stable. Frequently, $\nabla \log(x_t)$ is called the return or growth rate. This general idea was used in Example 3.21, and we will use it again in Example 3.27.

After suitably transforming the data, the next step is to identify preliminary values of the autoregressive order, p, the order of differencing, d, and the moving average order, q. A time plot of the data will typically suggest whether any differencing is needed. If differencing is called for, then difference the data once, $d = 1$, and inspect the time plot of ∇x_t. If additional differencing is necessary, then try differencing again and inspect a time plot of $\nabla^2 x_t$. 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

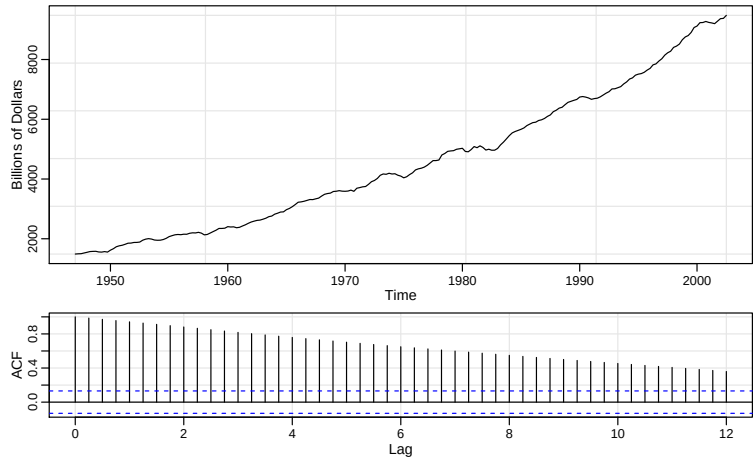


Fig. 3.11. Top: *Quarterly U.S. GNP from 1947(1) to 2002(3).* Bottom: *Sample ACF of the GNP data. Lag is in terms of years.*

differencing is needed. Because the polynomial $\phi(z)(1-z)^d$ has a unit root, the sample ACF, $\hat{\rho}(h)$, will not decay to zero fast as h increases. Thus, a slow decay in $\hat{\rho}(h)$ 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. Note that it cannot be the case that both the ACF and PACF cut off. Because we are dealing with estimates, it will not always be clear whether the sample ACF or PACF is tailing off or cutting off. Also, two models that are seemingly different can actually be very similar. With this in mind, we should not worry about being so precise at this stage of the model fitting. At this point, a few preliminary values of p, d, and q should be at hand, and we can start estimating the parameters.

Example 3.27 Analysis of GNP Data

In this example, we consider the analysis of quarterly U.S. GNP from 1947(1) to 2002(3), $n = 223$ observations. The data are real U.S. gross national product in billions of chained 1996 dollars and have been seasonally adjusted. The data were obtained from the Federal Reserve Bank of St. Louis (http://research.stlouisfed.org/). Figure 3.11 shows a plot of the data, say, y_t. Because strong trend tends to obscure other effects, it is difficult to see any other variability in data except for periodic large dips in the economy. When reports of GNP and similar economic indicators are given, it is often in growth rate (percent change) rather than in actual (or adjusted) values that is of interest. The growth rate, say, $x_t = \nabla \log(y_t)$, is plotted in Figure 3.12, and it appears to be a stable process.

The sample ACF and PACF of the quarterly growth rate are plotted in Figure 3.13. 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 lag 1. This suggests an

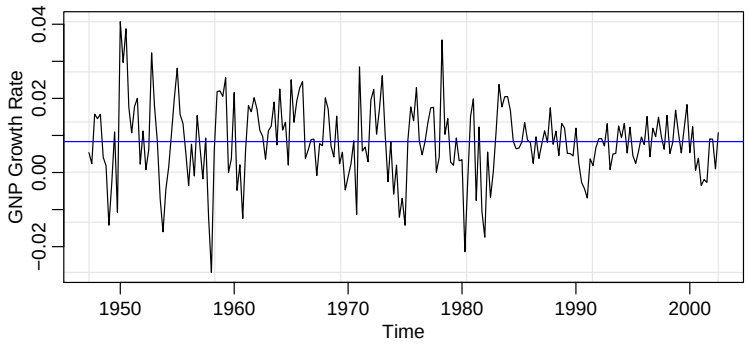


Fig. 3.12. *U.S. GNP quarterly growth rate. The horizontal line displays the average growth of the process, which is close to 1%.*

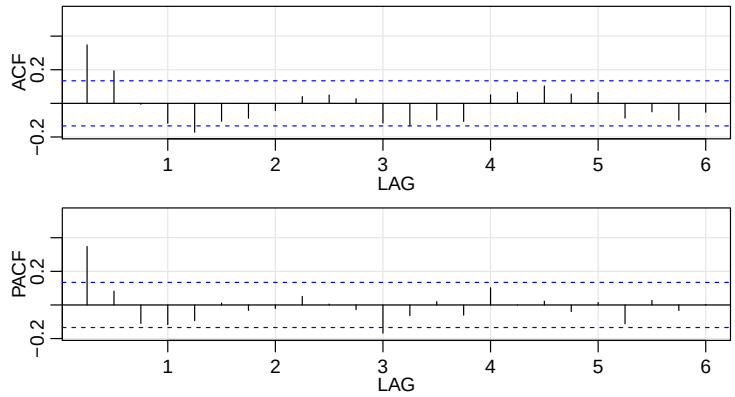


Fig. 3.13. *Sample ACF and PACF of the GNP quarterly growth rate. Lag is in years.*

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

$$\hat{x}_t = .008_{(.001)} + .303_{(.065)}\hat{w}_{t-1} + .204_{(.064)}\hat{w}_{t-2} + \hat{w}_t, \tag{3.58}$$

where $\hat{\sigma}_w = .0094$ 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 3.12). We leave it to the reader to investigate what happens when the constant is not included.

The estimated AR(1) model is

$$\hat{x}_t = .008_{(.001)}\,(1 - .347) + .347_{(.063)}x_{t-1} + \hat{w}_t, \tag{3.59}$$

where $\hat{\sigma}_w = .0095$ on 220 degrees of freedom; note that the constant in (3.59) 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 (3.58) and (3.59)? In fact, the fitted models are nearly the same. To show this, consider an AR(1) model of the form in (3.59) 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 (3.58).

The analysis can be performed in R as follows; partial output is shown.

```
tsplot(gnp);  acf2(gnp, 50)
gnpgr = diff(log(gnp))  # growth rate
tsplot(gnpgr);  acf2(gnpgr, 24)
sarima(gnpgr, 1, 0, 0)  # AR(1)
 $ttable
         Estimate     SE t.value p.value
  ar1     0.3467 0.0627  5.5255       0
  xmean   0.0083 0.0010  8.5398       0
sarima(gnpgr, 0, 0, 2)      # MA(2)
 $ttable
       Estimate     SE t.value p.value
  ma1     0.3028 0.0654  4.6272  0.0000
  ma2     0.2035 0.0644  3.1594  0.0018
  xmean   0.0083 0.0010  8.7178  0.0000
round( ARMAtoMA(ar=.35, ma=0, 10), 3)  # prints psi-weights
 [1] 0.350 0.122 0.043 0.015 0.005 0.002 0.001 0.000 0.000 0.000
```

The next step in model fitting is 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), $x_t - \hat{x}_t^{t-1}$, or of the standardized innovations

$$e_t = \left(x_t - \hat{x}_t^{t-1}\right) \,/\, \sqrt{\hat{P}_t^{t-1}}, \tag{3.60}$$

where $\hat{x}_t^{t-1}$ is the one-step-ahead prediction of x_t based on the fitted model and $\hat{P}_t^{t-1}$ is the estimated one-step-ahead error variance. If the model fits well, the standardized residuals should behave as an iid sequence with mean zero and variance one. The time plot should be inspected for any obvious departures from this assumption.

Investigation of marginal normality can be accomplished visually by looking at a histogram of the residuals. In addition to this, a normal Q-Q plot can help in identifying departures from normality.

We could also inspect the sample autocorrelations of the residuals, say, $\hat{\rho}_e(h)$, for any patterns or large values. In addition to plotting $\hat{\rho}_e(h)$, we can perform a general test of whiteness that takes into consideration the magnitudes of $\hat{\rho}_e(h)$ as a group. The Ljung–Box–Pierce Q-statistic given by

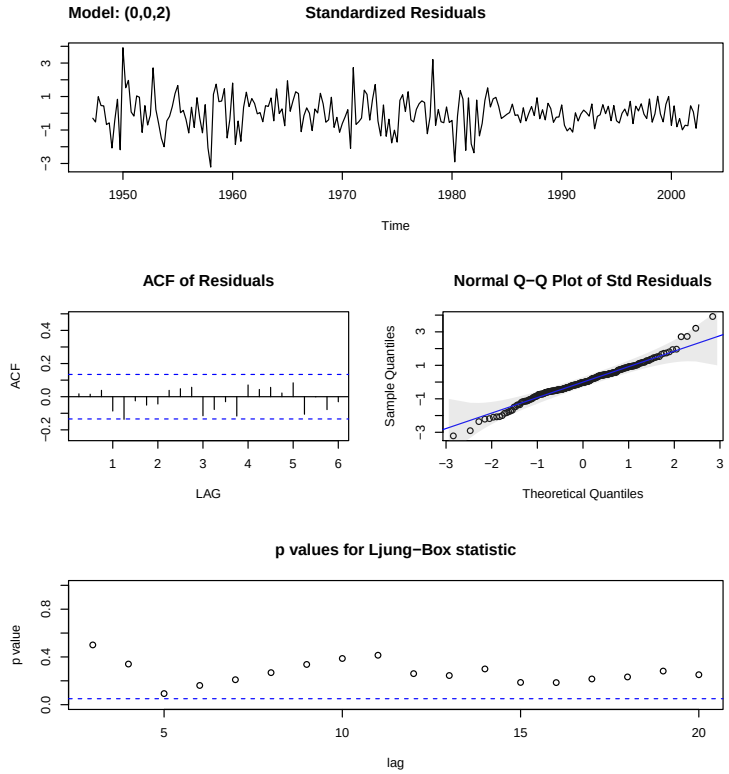


Fig. 3.14. *Diagnostics of the residuals from MA(2) fit on GNP growth rate.*

$$Q = n(n+2)\sum_{h=1}^{H} \frac{\hat{\rho}_e^2(h)}{n-h} \tag{3.61}$$

can be used to perform such a test. The value H in (3.61) 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 1.3, $n\hat{\rho}_w^2(h)$, for $h = 1, \ldots, H$, are asymptotically independent χ^2_1 random variables. This means that $n\sum_{h=1}^{H}\hat{\rho}_w^2(h)$ 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 (3.61) are used to adjust the statistic to better match the asymptotic chi-squared distribution.

Example 3.28 Diagnostics for GNP Growth Rate Example

We will focus on the MA(2) fit from Example 3.27; the analysis of the AR(1) residuals is similar. Figure 3.14 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, (3.61), at lags $H = 3$ through $H = 20$ (with corresponding degrees of freedom $H - 2$).

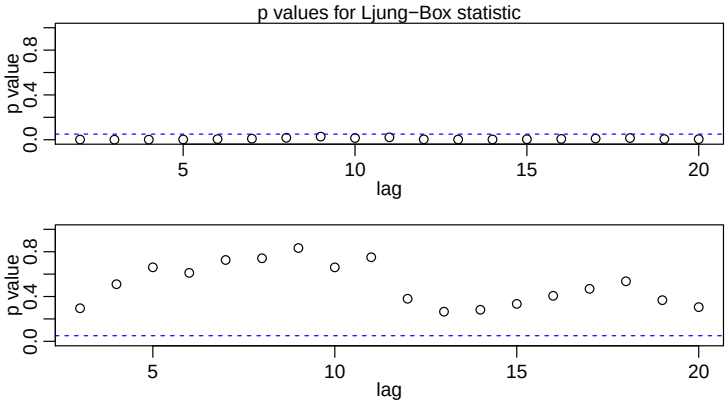


Fig. 3.15. *Q-statistic p-values for the ARIMA*$(0,1,1)$ *fit (top) and the ARIMA*$(1,1,1)$ *fit (bottom) to the logged varve data.*

Inspection of the time plot of the standardized residuals in Figure 3.14 shows no obvious patterns. Notice that there may be 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 suggests that the assumption of normality is appropriate. The diagnostics shown in Figure 3.14 are a by-product of the `sarima` command from the previous example.

Example 3.29 Diagnostics for the Glacial Varve Series

In Example 3.21, 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 3.15.

To adjust for this problem, we fit an ARIMA$(1,1,1)$ to the logged varve data and obtained the estimates

$$\hat{\phi} = .23_{(.05)},\ \hat{\theta} = -.89_{(.03)},\ \hat{\sigma}_w^2 = .23.$$

Hence the AR term is significant. The Q-statistic p-values for this model are also displayed in Figure 3.15, and it appears this model fits the data well.

As previously stated, the diagnostics are byproducts of the individual `sarima` runs. We note that we did not fit a constant in either model because there is no apparent drift in the differenced, logged varve series. This fact can be verified by noting the constant is not significant when the command `no.constant=TRUE` is removed in the code:

```
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 3.27, 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

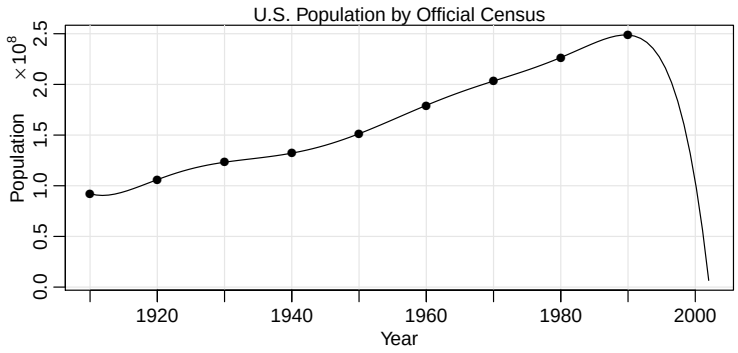


Fig. 3.16. *A perfect fit and a terrible forecast.*

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 3.30 A Problem with Overfitting

Figure 3.16 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 predicted 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! I see dead people.

The final step of model fitting is model choice or model selection. That is, we must decide which model we will retain for forecasting. The most popular techniques, AIC, AICc, and BIC, were described in Section 2.1 in the context of regression models.

Example 3.31 Model Choice for the U.S. GNP Series

To follow up on Example 3.28, 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 3.27, but for convenience, we display them again here (recall the growth rate data are in `gnpgr`):

```
sarima(gnpgr, 1, 0, 0)  # AR(1)
  $AIC: -8.294403   $AICc: -8.284898   $BIC: -9.263748
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. In this case, it is reasonable to retain the AR(1) because pure autoregressive models are easier to work.

3.8 Regression with Autocorrelated Errors

In Section 2.1, we covered the classical regression model with uncorrelated errors w_t. In this section, we discuss the modifications that might be considered when the errors are correlated. That is, consider the regression model

$$y_t = \beta_1 z_{t1} + \cdots + \beta_r z_{tr} + x_t = \sum_{j=1}^{r} \beta_j z_{tj} + x_t \tag{3.62}$$

where x_t is a process with some covariance function $\gamma_x(s,t)$. In ordinary least squares, the assumption is that x_t is white Gaussian noise, in which case $\gamma_x(s,t) = 0$ for $s \neq t$ and $\gamma_x(t,t) = \sigma^2$, independent of t. If this is not the case, then weighted least squares should be used.

In the time series case, it is often possible to assume a stationary covariance structure for the error process x_t that corresponds to a linear process and try to find an ARMA representation for x_t. For example, if we have a pure AR(p) error, then

$$\phi(B)x_t = w_t,$$

and $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ is the linear transformation that, when applied to the error process, produces the white noise w_t. Multiplying the regression equation through by the transformation $\phi(B)$ yields,

$$\underbrace{\phi(B)y_t}_{y_t^*} = \sum_{j=1}^{r} \beta_j \underbrace{\phi(B)z_{tj}}_{z_{tj}^*} + \underbrace{\phi(B)x_t}_{w_t},$$

and we are back to the linear regression model where the observations have been transformed so that $y_t^* = \phi(B)y_t$ is the dependent variable, $z_{tj}^* = \phi(B)z_{tj}$ for $j = 1, \ldots, r$, are the independent variables, but the βs are the same as in the original model.

For example, suppose we have the regression model

$$y_t = \alpha + \beta z_t + x_t$$

where $x_t = \phi x_{t-1} + w_t$ is AR(1). Then, transform the data as $y_t^* = y_t - \phi y_{t-1}$ and $z_t^* = z_t - \phi z_{t-1}$ so that the new model is

$$\underbrace{y_t - \phi y_{t-1}}_{y_t^*} = \underbrace{(1-\phi)\alpha}_{\alpha^*} + \underbrace{\beta(z_t - \phi z_{t-1})}_{\beta z_t^*} + \underbrace{(x_t - \phi x_{t-1})}_{w_t}$$

In the AR case, we may set up the least squares problem as minimizing the error sum of squares

$$S(\phi, \beta) = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} \Big[\phi(B)y_t - \sum_{j=1}^{r} \beta_j \phi(B) z_{tj}\Big]^2$$

with respect to all the parameters, $\phi = \{\phi_1, \ldots, \phi_p\}$ and $\beta = \{\beta_1, \ldots, \beta_r\}$. Of course, this is done using numerical methods.

If the error process is ARMA(p,q), i.e., $\phi(B)x_t = \theta(B)w_t$, then in the above discussion, we transform by $\pi(B)x_t = w_t$, where, recalling (3.16),

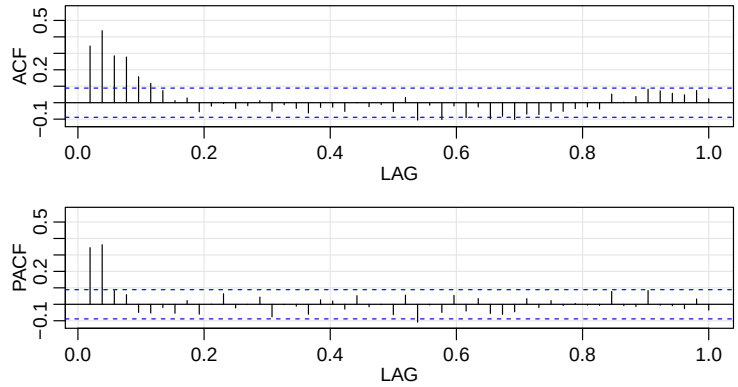


Fig. 3.17. *Sample ACF and PACF of the mortality residuals indicating an AR(2) process.*

$\pi(B) = \theta(B)^{-1}\phi(B)$. In this case the error sum of squares also depends on $\theta = \{\theta_1, \ldots, \theta_q\}$:

$$S(\phi, \theta, \beta) = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} \Big[\pi(B)y_t - \sum_{j=1}^{r} \beta_j \pi(B) z_{tj}\Big]^2$$

At this point, the main problem is that we do not typically know the behavior of the noise x_t prior to the analysis. An easy way to tackle this problem was first presented in Cochrane and Orcutt (1949), and with the advent of cheap computing is modernized below:

(i) First, run an ordinary regression of y_t on $z_{t1}, \ldots, z_{tr}$ (acting as if the errors are uncorrelated). Retain the residuals, $\hat{x}_t = y_t - \sum_{j=1}^{r} \hat{\beta}_j z_{tj}$.
(ii) Identify ARMA model(s) for the residuals $\hat{x}_t$.
(iii) Run weighted least squares (or MLE) on the regression model with autocorrelated errors using the model specified in step (ii).
(iv) Inspect the residuals $\hat{w}_t$ for whiteness, and adjust the model if necessary.

Example 3.32 Mortality, Temperature and Pollution

We consider the analyses presented in Example 2.2, relating mean adjusted temperature T_t, and particulate levels P_t to cardiovascular mortality M_t. We consider the regression model

$$M_t = \beta_0 + \beta_1 t + \beta_2 T_t + \beta_3 T_t^2 + \beta_4 P_t + x_t, \tag{3.63}$$

where, for now, we assume that x_t is white noise. The sample ACF and PACF of the residuals from the ordinary least squares fit of (3.63) are shown in Figure 3.17, and the results suggest an AR(2) model for the residuals. The next step is to fit the model (3.63) where x_t is AR(2),

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

and w_t is white noise. The model can be fit using `sarima` as follows (partial output shown).

```
trend = time(cmort); temp = tempr - mean(tempr); temp2 = temp^2
fit = lm(cmort~trend + temp + temp2 + part, na.action=NULL)
acf2(resid(fit), 52)   # implies AR2
sarima(cmort, 2,0,0, xreg=cbind(trend, temp, temp2, part) )
 $ttable
             Estimate       SE t.value p.value
   ar1          0.3848   0.0436  8.8329  0.0000
   ar2          0.4326   0.0400 10.8062  0.0000
   intercept 3075.1482 834.7157  3.6841  0.0003
   trend       -1.5165   0.4226 -3.5882  0.0004
   temp        -0.0190   0.0495 -0.3837  0.7014
   temp2        0.0154   0.0020  7.6117  0.0000
   part         0.1545   0.0272  5.6803  0.0000
```

The residual analysis output from `sarima` (not shown) shows no obvious departure of the residuals from whiteness. Also, note that `temp`, T_t, is not significant because it has been centered, that is $T_t = {}^\circ\mathrm{F_t} - {}^\circ\overline{\mathrm{F}}$ where ${}^\circ\mathrm{F_t}$ is the actual temperature measured in degrees Fahrenheit 451. Thus `temp2` is $T_t^2 = ({}^\circ\mathrm{F_t} - {}^\circ\overline{\mathrm{F}})^2$, so a linear term for temperature is in the model twice.

Example 3.33 Regression with Lagged Variables (cont)

In Example 2.9 we fit the model

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

where R_t is Recruitment, S_t is SOI, and D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. However, residual analysis indicates that the residuals are not white noise. The sample (P)ACF of the residuals indicates that an AR(2) model might be appropriate, which is similar to the results of Example 3.32. We display partial results of the final model below.

```
dummy = ifelse(soi<0, 0, 1)
fish  = ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6),
           dframe=TRUE)
summary(fit <- lm(rec ~soiL6*dL6, data=fish, na.action=NULL))
attach(fish)
tsplot(resid(fit))
acf2(resid(fit))      # indicates AR(2)
intract = soiL6*dL6   # interaction term
sarima(rec,2,0,0, xreg = cbind(soiL6, dL6, intract))
 $ttable
             Estimate     SE  t.value p.value
   ar1         1.3624 0.0440  30.9303  0.0000
   ar2        -0.4703 0.0444 -10.5902  0.0000
   intercept  64.8028 4.1121  15.7590  0.0000
   soiL6       8.6671 2.2205   3.9033  0.0001
   dL6        -2.5945 0.9535  -2.7209  0.0068
   intract   -10.3092 2.8311  -3.6415  0.0003
detach(fish)
```

There appears to be some correlation left at the seasonal lags. The next section discusses how to handle seasonal autocorrelation.

3.9 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

$$\Phi_P(B^s)x_t = \Theta_Q(B^s)w_t, \tag{3.64}$$

where the operators

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps} \tag{3.65}$$

and

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs} \tag{3.66}$$

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 3.34 A Seasonal AR Series

A first-order seasonal autoregressive series that might run over months could be written as

$$(1 - \Phi B^{12})x_t = w_t$$

or

$$x_t = \Phi x_{t-12} + w_t.$$

This model exhibits the series x_t in terms of past lags at the multiple of the yearly seasonal period $s = 12$ months. It is clear 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 $|\Phi| < 1$.

We simulated 3 years of data from the model with $\Phi = .9$, and exhibit the *theoretical* ACF and PACF of the model; see Figure 3.18.

```
set.seed(666)
phi = c(rep(0,11),.9)
sAR = arima.sim(list(order=c(12,0,0), ar=phi), n=37)
sAR = ts(sAR, freq=12)
layout(matrix(c(1,2, 1,3), nc=2))
par(mar=c(3,3,2,1), mgp=c(1.6,.6,0))
plot(sAR, axes=FALSE, main='seasonal AR(1)', xlab="year", type='c')
```

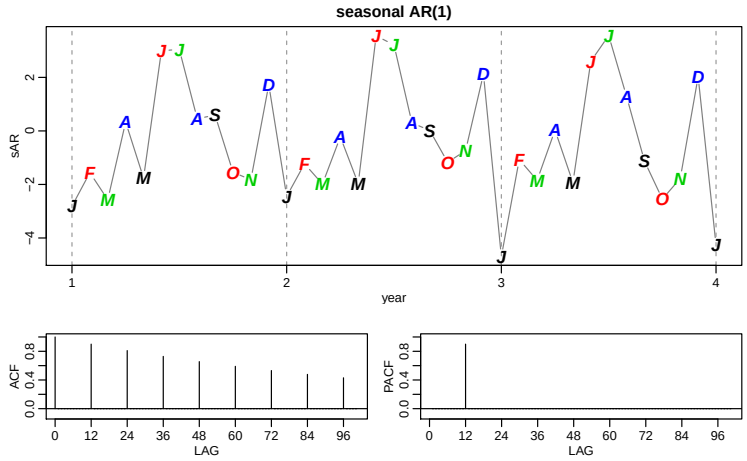


Fig. 3.18. *Data generated from a seasonal ($s = 12$) AR(1), and the true ACF and PACF of the model* $x_t = .9x_{t-12} + w_t$.

```
Months = c("J","F","M","A","M","J","J","A","S","O","N","D")
points(sAR, pch=Months, cex=1.25, font=4, col=1:4)
axis(1, 1:4)
abline(v=1:4, lty=2, col=gray(.6))
axis(2)
box()
ACF  = ARMAacf(ar=phi, ma=0, 100)
PACF = ARMAacf(ar=phi, ma=0, 100, pacf=TRUE)
plot(ACF,  type="h", xlab="lag", ylim=c(-.1,1))
abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.1,1))
abline(h=0)
```

For the first-order seasonal ($s = 12$) MA model, $x_t = w_t + \Theta w_{t-12}$, it is easy to verify that

$$\begin{aligned} \gamma(0) &= (1+\Theta^2)\sigma^2 \\ \gamma(\pm 12) &= \Theta\sigma^2 \\ \gamma(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

Thus, the only nonzero correlation, aside from lag zero, is

$$\rho(\pm 12) = \Theta/(1+\Theta^2).$$

For the first-order seasonal ($s = 12$) AR model, using the techniques of the nonseasonal AR(1), we have

$$\begin{aligned} \gamma(0) &= \sigma^2/(1-\Phi^2) \\ \gamma(\pm 12k) &= \sigma^2\Phi^k/(1-\Phi^2) \quad k = 1, 2, \ldots \\ \gamma(h) &= 0, \quad \text{otherwise.} \end{aligned}$$

In this case, the only non-zero correlations are

$$\boxed{\rho(\pm 12k) = \Phi^k, \quad k = 0, 1, 2, \ldots .}$$

These results can be verified using the general result that

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.

$$\gamma(h) = \Phi\gamma(h-12) \quad \text{for } h \geq 1\,.$$

For example, when $h = 1$, $\gamma(1) = \Phi\gamma(11)$, but when $h = 11$, we have $\gamma(11) = \Phi\gamma(1)$, which implies that $\gamma(1) = \gamma(11) = 0$. In addition to these results, the PACF have the analogous extensions from nonseasonal to seasonal models. These results are demonstrated in Figure 3.18.

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

$$\Phi_P(B^s)\phi(B)x_t = \Theta_Q(B^s)\theta(B)w_t \tag{3.67}$$

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 Table 3.1 and Table 3.3. In fitting such models, focusing on the seasonal autoregressive and moving average components first generally leads to more satisfactory results.

Example 3.35 A Mixed Seasonal Model

Consider an $\text{ARMA}(0,1) \times (1,0)_{12}$ 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

$$\begin{aligned} \rho(12h) &= \Phi^h \quad h = 1, 2, \ldots \\ \rho(12h-1) = \rho(12h+1) &= \frac{\theta}{1+\theta^2}\Phi^h \quad h = 0, 1, 2, \ldots, \end{aligned}$$

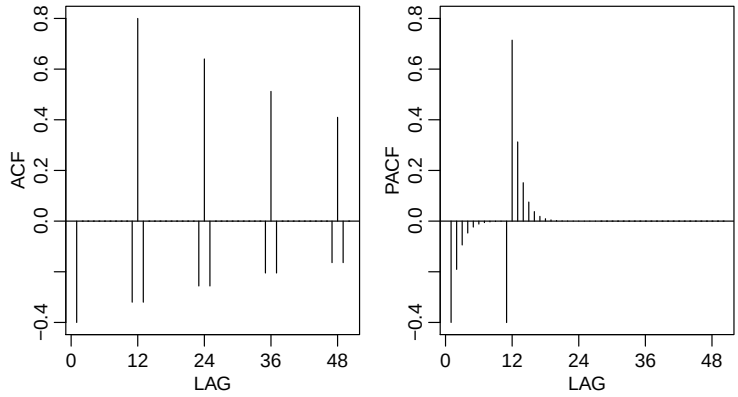


Fig. 3.19. *ACF and PACF of the mixed seasonal ARMA model* $x_t = .8x_{t-12} + w_t - .5w_{t-1}$.

$$\rho(h) = 0, \quad \text{otherwise.}$$

The ACF and PACF for this model, with $\Phi = .8$ and $\theta = -.5$, are shown in Figure 3.19. These type of correlation relationships, although idealized here, are typically seen with seasonal data.

To reproduce Figure 3.19 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)
```

The pattern in the ACF is typical of seasonal time series. Try this on your own and compare it to Figure 3.19.

```
par(mfrow=c(3,1),mar=c(2,2,0,0)+1, mgp=c(1.6,.6,0))
tsplot(birth)                 # monthly number of births in US
tsplot( diff(log(birth)) )    # the growth rate
acf1( diff(log(birth)), 61)   # the sample ACF
```

Seasonal persistence occurs 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 a little from one year to the next, according to a random walk,

$$S_t = S_{t-12} + v_t.$$

In this model, w_t and v_t are uncorrelated white noise processes.

For another example, consider the quarterly occupancy rate of Hawaiian hotels shown in Figure 3.20 (recall Example 2.15). The seasonal component, shown below the data, is extracted by removing the trend component from the data. Note that the occupancy rate for the first and third quarters is always up 2%

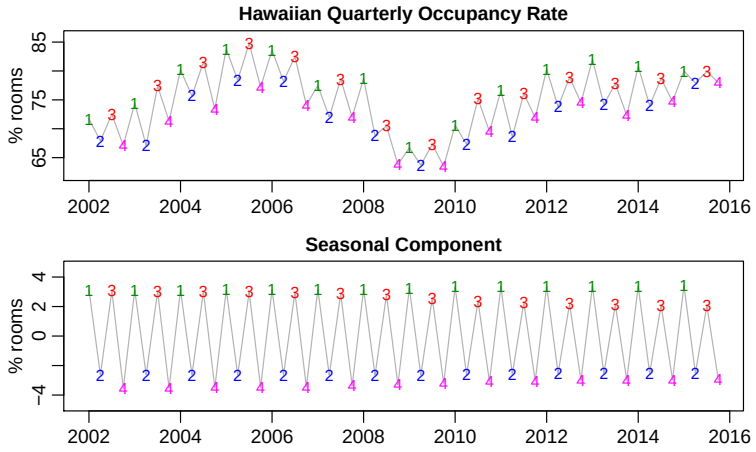


Fig. 3.20. *Seasonal persistence: The quarterly occupancy rate of Hawaiian hotels and the extracted seasonal component.*

to 4%, while the occupancy rate for the second and fourth quarters is always down 2% to 4%.

The tendency of data to follow this type of behavior will be exhibited in a sample ACF that is large and decays very slowly at lags $h = 12k$, 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 $\mathrm{MA}(1)_{12}$, 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

$$\nabla_s^D x_t = (1 - B^s)^D x_t, \tag{3.68}$$

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 3.8 *The multiplicative* **seasonal autoregressive integrated moving average** *model, or* **SARIMA** *model is given by*

$$\Phi_P(B^s)\phi(B)\nabla_s^D\nabla^d x_t = \delta + \Theta_Q(B^s)\theta(B)w_t, \tag{3.69}$$

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, and the seasonal autoregressive and moving average components by $\Phi_P(B^s)$ and $\Theta_Q(B^s)$ of orders P and Q and ordinary and seasonal difference components by $\nabla^d = (1 - B)^d$ and $\nabla_s^D = (1 - B^s)^D$.*

Example 3.36 An SARIMA Model

Consider the following model, which often provides a reasonable representation for seasonal, nonstationary, economic time series. We exhibit the equations for the model, denoted by $\text{ARIMA}(0,1,1)\times(0,1,1)_{12}$ in the notation given above, where the seasonal fluctuations occur every 12 months. Then, with $\delta = 0$, the model (3.69) becomes

$$\nabla_{12}\nabla x_t = \Theta(B^{12})\theta(B)w_t$$

or

$$(1-B^{12})(1-B)x_t = (1+\Theta B^{12})(1+\theta B)w_t. \tag{3.70}$$

Expanding both sides of (3.70) 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 (3.69) is a daunting task, and we usually 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 Table 3.1 and Table 3.3. In considering whether the model is satisfactory, the diagnostic techniques discussed in Section 3.7 still apply.

Example 3.37 Air Passengers

We consider the R data set `AirPassengers`, which are the monthly totals of international airline passengers, 1949 to 1960, taken from Box & Jenkins (1970). Various plots of the data and transformed data are shown in Figure 3.21 and were obtained as follows:

```
x = AirPassengers
lx = log(x); dlx = diff(lx); ddlx = diff(dlx, 12)
plot.ts(cbind(x,lx,dlx,ddlx), yax.flip=TRUE, main="")
# below of interest for showing seasonal RW (not shown here):
par(mfrow=c(2,1))
monthplot(dlx); monthplot(ddlx)
```

Note that `x` is the original series, which shows trend plus increasing variance. The logged data are in `lx`, and the transformation stabilizes the variance. The logged data are then differenced to remove trend, and are stored in `dlx`. It is

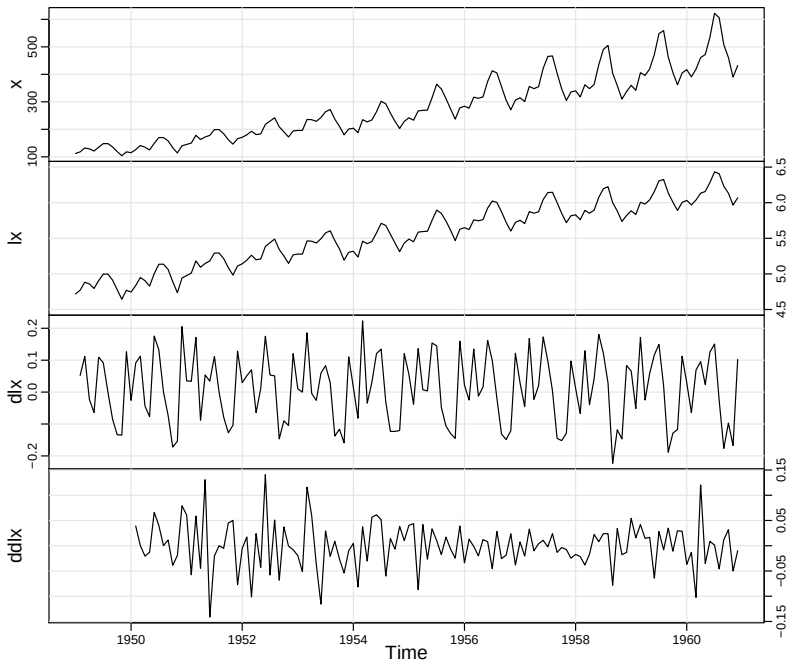


Fig. 3.21. *R data set* `AirPassengers`*, which are the monthly totals of international airline passengers* `x`*, and the transformed data:* `lx` $= \log x_t$*,* `dlx` $= \nabla \log x_t$*, and* `ddlx` $= \nabla_{12}\nabla \log x_t$*.*

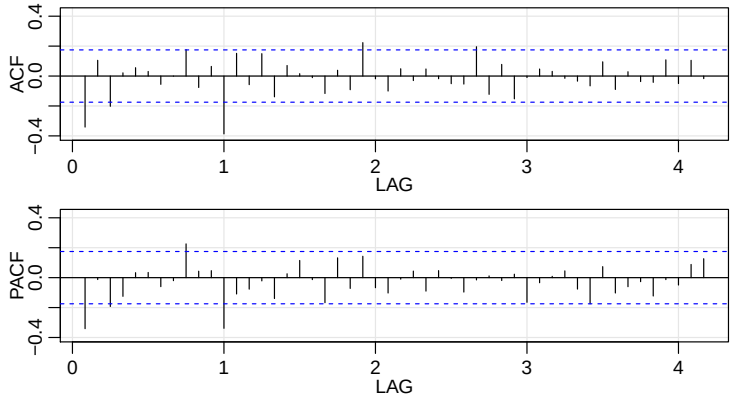


Fig. 3.22. *Sample ACF and PACF of* `ddlx` *(*$\nabla_{12}\nabla \log x_t$*).*

clear the there is still persistence in the seasons (i.e., $\texttt{dlx}_t \approx \texttt{dlx}_{t-12}$), so that a twelfth-order difference is applied and stored in `ddlx`. The transformed data appears to be stationary and we are now ready to fit a model.

The sample ACF and PACF of `ddlx` ($\nabla_{12}\nabla \log x_t$) are shown in Figure 3.22. The R code is:

```
acf2(ddlx, 50)
```

Seasonal: It appears that at the seasons, the ACF is cutting off a lag $1s$ ($s = 12$), whereas the PACF is tailing off at lags $1s, 2s, 3s, 4s, \ldots$. These results implies an SMA(1), $P = 0$, $Q = 1$, in the season ($s = 12$).

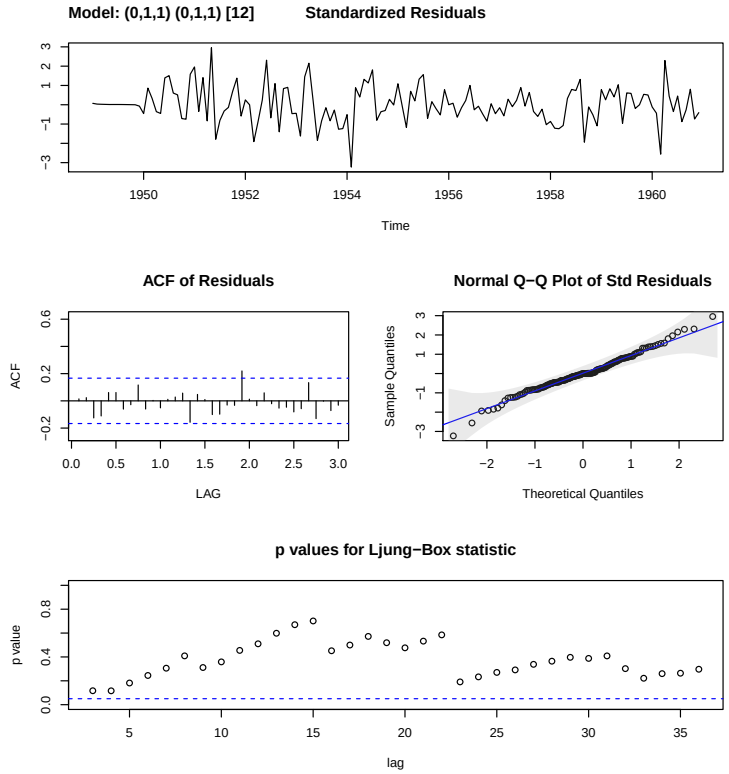


Fig. 3.23. *Residual analysis for the ARIMA*$(0,1,1) \times (0,1,1)_{12}$ *fit to the logged air passengers data set.*

Non-Seasonal: Inspecting the sample ACF and PACF at the lower lags, it appears as though both are tailing off. This suggests an ARMA$(1,1)$ within the seasons, $p = q = 1$.

Thus, we first try an ARIMA$(1,1,1) \times (0,1,1)_{12}$ on the logged data:

```
sarima(lx, 1,1,1, 0,1,1,12)
           ar1      ma1     sma1
        0.1960  -0.5784  -0.5643
s.e.  0.2475   0.2132   0.0747
sigma^2 estimated as 0.001341
$AIC -5.5726   $AICc -5.55671  $BIC -6.510729
```

However, the AR parameter is not significant, so we should try dropping one parameter from the within seasons part. In this case, we try both an ARIMA$(0,1,1) \times (0,1,1)_{12}$ and an ARIMA$(1,1,0) \times (0,1,1)_{12}$ model:

```
sarima(lx, 0,1,1, 0,1,1,12)
            ma1     sma1
         -0.4018  -0.5569
s.e.   0.0896   0.0731
sigma^2 estimated as 0.001348
$AIC -5.5813  $AICc -5.5663  $BIC -6.5401

sarima(lx, 1,1,0, 0,1,1,12)
            ar1     sma1
         -0.3395  -0.5619
s.e.   0.0822   0.0748
```

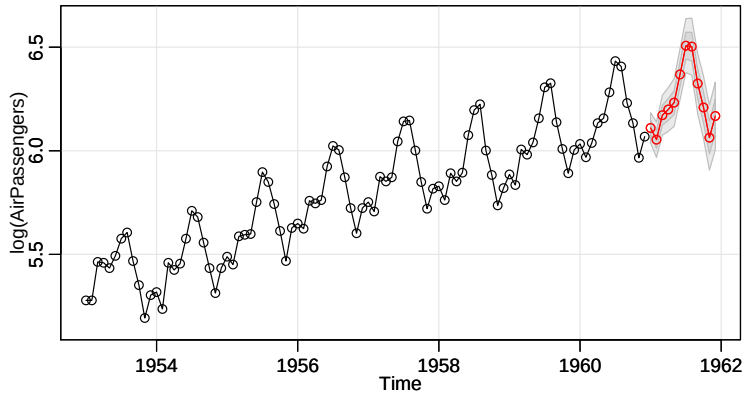


Fig. 3.24. *Twelve month forecast using the ARIMA*$(0,1,1)\times(0,1,1)_{12}$ *model on the logged air passenger data set.*

```
sigma^2 estimated as 0.001367
$AIC -5.5671  $AICc -5.5520  $BIC -6.5258
```

All information criteria prefer the ARIMA$(0,1,1)\times(0,1,1)_{12}$ model, which is the model displayed in (3.70). The residual diagnostics are shown in Figure 3.23, and the model seems to fit well.

Finally, we forecast the logged data out twelve months, and the results are shown in Figure 3.24.

```
sarima.for(lx, 12, 0,1,1, 0,1,1,12)
```

Problems

3.1 For an MA(1), $x_t = w_t + \theta w_{t-1}$, show that $|\rho_x(1)| \le 1/2$ for any number θ. For which values of θ does $\rho_x(1)$ attain its maximum and minimum?

3.2 Let $\{w_t; t = 0,1,\dots\}$ be a white noise process with variance σ_w^2 and let $|\phi| < 1$ be a constant. Consider the process $x_0 = w_0$, and

$$x_t = \phi x_{t-1} + w_t, \quad t = 1,2,\dots .$$

We might use this method to simulate an AR(1) process from simulated white noise.

(a) Show that $x_t = \sum_{j=0}^{t} \phi^j w_{t-j}$ for any $t = 0,1,\dots$.
(b) Find the $E(x_t)$.
(c) Show that, for $t = 0,1,\dots$,[6]

$$\text{var}(x_t) = \frac{\sigma_w^2}{1-\phi^2}(1-\phi^{2(t+1)})$$

(d) Show that, for $h \ge 0$,

[6] $\sum_{j=0}^{k} a^j = (1-a^{k+1})/(1-a)$ for $|a| \ne 1$

$$\text{cov}(x_{t+h}, x_t) = \phi^h \text{var}(x_t)$$

Hint: Use the idea in footnote 1 where, by iterating backward, we showed $x_t = \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j w_{t-j}$.

(e) Is x_t stationary?

(f) Argue that, as $t \to \infty$, the process becomes stationary, so in a sense, x_t is "asymptotically stationary."

(g) Comment on how you could use these results to simulate n observations of a stationary Gaussian AR(1) model from simulated iid N(0,1) values.

(h) Now suppose $x_0 = w_0/\sqrt{1-\phi^2}$. Is this process stationary? *Hint*: Show $\text{var}(x_t)$ is constant.

3.3 Using Example 3.7 as a guide, identify the following models as ARMA(p, q) models (watch out for parameter redundancy), and determine whether they are causal and/or invertible. If the model is causal, use R to find the first 10 ψ-weights, and if the model is invertible, use R to find the first 10 π-weights.

(a) $x_t = .80x_{t-1} - .15x_{t-2} + w_t - .30w_{t-1}$.

(b) $x_t = x_{t-1} - .50x_{t-2} + w_t - w_{t-1}$.

3.4 For the AR(2) model given by $x_t = -.9x_{t-2} + w_t$, follow the R code in Example 3.12 to find the roots of the autoregressive polynomial, find the pseudo period of the process, and then plot the theoretical ACF, $\rho(h)$.

3.5 (a) Compare the *theoretical* ACF and PACF of an ARMA$(1,1)$, an ARMA$(1,0)$, and an ARMA$(0,1)$ series by plotting the ACFs and PACFs of the three series for $\phi = .6$, $\theta = .9$. Comment on the capability of the ACF and PACF to determine the order of the models. *Hint:* See the code for Example 3.14.

(b) Use `arima.sim` to generate $n = 100$ observations from each of the three models discussed in (a). Compute the sample ACFs and PACFs for each model and compare it to the theoretical values. How do the results compare with the general results given in Table 3.1?

(c) Repeat (b) but with $n = 500$. Comment.

3.6 Let c_t be the cardiovascular mortality series (`cmort`) discussed in Chapter 2, Example 2.2 and let $x_t = \nabla c_t$ be the differenced data.

(a) Plot x_t and compare it to the actual data plotted in Figure 2.2. Why does differencing seem reasonable in this case?

(b) Calculate and plot the sample ACF and PACF of x_t and using Table 3.1, argue that an AR(1) is appropriate for x_t.

(c) Fit an AR(1) to x_t using maximum likelihood (basically unconditional least squares) as in **??**. The easiest way to do this is to use `sarima` from `astsa`. Comment on the significance of the regression parameter estimates of the model. What is the estimate of the white noise variance?

(d) Examine the residuals and comment on whether or not you think the residuals are white.

(e) Assuming the fitted model is the true model, find the forecasts over a four-week horizon, x_{n+m}^n, for $m = 1, 2, 3, 4$, and the corresponding 95%

prediction intervals; $n = 508$ here. The easiest way to do this is to use `sarima.for` from `astsa`.

(f) Show how the values obtained in part (e) were calculated.

(g) What is the one-step-ahead forecast of the actual value of cardiovascular mortality; i.e., what is c_{n+1}^n?

3.7 For an AR(1) model, determine the general form of the m-step-ahead forecast x_{n+m}^n and show

$$E[(x_{n+m} - x_{n+m}^n)^2] = \sigma_w^2 \frac{1 - \phi^{2m}}{1 - \phi^2}.$$

3.8 Repeat the following numerical exercise five times. Generate $n = 100$ iid N(0, 1) observations. Fit an ARMA(1, 1) model to the data. Compare the parameter estimates in each case and explain the results.

3.9 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.

3.10 Using Example 3.20 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.

3.11 Verify that the IMA(1,1) model given in (3.55) can be inverted and written as (3.56).

3.12 For the logarithm of the glacial varve data, say, x_t, presented in Example 3.21, use the first 100 observations and calculate the EWMA, x_{n+1}^n, discussed in Example 3.26, for $n = 1, \ldots, 100$, using $\lambda = .25, .50$, and $.75$, and plot the EWMAs and the data superimposed on each other. Comment on the results.

3.13 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.

3.14 Fit an ARIMA(p, d, q) model to the land-based global temperature data `globtempl` in `astsa` performing all of the necessary diagnostics; include a model choice analysis. After deciding on an appropriate model, forecast (with limits) the next 10 years. Comment.

3.15 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.

3.16 Let S_t represent the monthly sales data in `sales` ($n = 150$), and let L_t be the leading indicator in `lead`.

(a) Fit an ARIMA model to S_t, the monthly sales data. Discuss your model fitting in a step-by-step fashion, presenting your (A) initial examination of the data, (B) transformations and differencing orders, if necessary, (C) initial identification of the dependence orders, (D) parameter estimation, (E) residual diagnostics and model choice.

(b) Use the CCF and lag plots between ∇S_t and ∇L_t to argue that a regression of ∇S_t on ∇L_{t-3} is reasonable. [*Note:* In `lag2.plot()`, the first named series is the one that gets lagged.]

(c) Fit the regression model $\nabla S_t = \beta_0 + \beta_1 \nabla L_{t-3} + x_t$, where x_t is an ARMA process (explain how you decided on your model for x_t). Discuss your results. *R help: If you have to work with various transformations of series in* `x` *and* `y`*, first align the data:*

```
dog = ts.intersect( lag(x,-11),  diff(y,97) )
xnew = dog[,1]  # dog has 2 columns, the first is lag(x,-11) ...
ynew = dog[,2]  # ... and the second column is diff(y,97)
plot(dog)    # now you can manipulate xnew and ynew simultaneously
lag2.plot(xnew, ynew, 5)
```

3.17 One of the remarkable technological developments in the computer industry has been the ability to store information densely on a hard drive. In addition, the cost of storage has steadily declined causing problems of *too much data* as opposed to *big data*. The data set for this assignment is `cpg`, which consists of the median annual retail price per GB of hard drives, say c_t, taken from a sample of manufacturers from 1980 to 2008.

(a) Plot c_t and describe what you see.

(b) Argue that the curve c_t versus t behaves like $c_t \approx \alpha e^{\beta t}$ by fitting a linear regression of $\log c_t$ on t and then plotting the fitted line to compare it to the logged data. Comment.

(c) Inspect the residuals of the linear regression fit and comment.

(d) Fit the regression again, but now using the fact that the errors are autocorrelated. Comment.

3.18 Redo Problem 2.2 without assuming the error term is white noise.

3.19 Plot the theoretical ACF of the seasonal ARIMA$(0,1) \times (1,0)_{12}$ model with $\Phi = .8$ and $\theta = .5$ out to lag 50.

3.20 Fit a seasonal ARIMA model of your choice to the chicken price data in `chicken`. Use the estimated model to forecast the next 12 months.

3.21 Fit a seasonal ARIMA model of your choice to the unemployment data, `UnempRate`. Use the estimated model to forecast the next 12 months.

3.22 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.

3.23 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.

3.24 In Example 3.33, it was noted that a seasonal model might do better. Rerun the model in that example but fit an ARIMA$(2,0,0) \times (0,1,1)_{12}$ to the residuals. Does this improve the fit?

Chapter 4

Spectral Analysis and Filtering

4.1 Introduction

The cyclic behavior of data is the focus of this chapter. For example, in the Johnson & Johnson data set in Figure 1.1, the predominant frequency of oscillation is one cycle per year (4 quarters), or $1/4$ cycles per observation. The predominant frequency in the SOI and fish populations series in Figure 1.5 is also one cycle per year, but this corresponds to 1 cycle every 12 months, or $1/12$ cycles per observation. The *period* of a time series is defined as the number of points in a cycle, i.e., $1/\omega$. Hence, the predominant period of the Johnson & Johnson series is 4 quarters per cycle, whereas the predominant period of the SOI series is 12 months per cycle. As stated in the Preface, `complex numbers` (a pdf) may be helpful for this chapter.

4.2 Periodicity and Cyclical Behavior

The general notion of periodicity can be made more precise by introducing some terminology. In order to define the rate at which a series oscillates, we first define a cycle as one complete period of a sine or cosine function defined over a unit time interval. As in (1.5), we consider the periodic process

$$x_t = A\cos(2\pi\omega t + \phi) \tag{4.1}$$

for $t = 0, \pm 1, \pm 2, \ldots$, where ω is a *frequency* index, defined in cycles per unit time with A determining the height or *amplitude* of the function and ϕ, called the *phase*, determining the start point of the cosine function. We can introduce random variation in this time series by allowing the amplitude and phase to vary randomly.

As discussed in Example 2.10, for purposes of data analysis, it is easier to use a trigonometric identity[1] and write (4.1) as

$$x_t = U_1\cos(2\pi\omega t) + U_2\sin(2\pi\omega t), \tag{4.2}$$

Both depend on amplitude and phase. If we consider them random, U_1, U_2 are random

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

where U_1 and U_2 are often taken to be independent normal random variables.

If we assume that U_1 and U_2 are uncorrelated random variables with mean 0 and variance σ^2, then x_t in (4.2) is stationary because $E(x_t) = 0$ and writing $c_t = \cos(2\pi\omega t)$ $s_t = \sin(2\pi\omega t)$ so that $x_t = U_1 c_t + U_2 s_t$, the autocovariance function is

$$\begin{aligned}\gamma(h) &= \operatorname{cov}(x_{t+h}, x_t) = \operatorname{cov}(U_1 c_{t+h} + U_2 s_{t+h}, U_1 c_t + U_2 s_t) \\ &= \operatorname{cov}(U_1 c_{t+h}, U_1 c_t) + \operatorname{cov}(U_1 c_{t+h}, U_2 s_t) \\ &\quad + \operatorname{cov}(U_2 s_{t+h}, U_1 c_t) + \operatorname{cov}(U_2 s_{t+h}, U_2 s_t) \\ &= \sigma^2 c_{t+h} c_t + 0 + 0 + \sigma^2 s_{t+h} s_t = \sigma^2 \cos(2\pi\omega h),\end{aligned} \tag{4.3}$$

using footnote 1 and noting that $\operatorname{cov}(U_1, U_2) = 0$.

The random process in (4.2) is function of its frequency, ω. For example, if $\omega = 1/4$, the series makes a cycle every four time units, and so on. Generally we consider data that occur at discrete time points, so we will need at least two points to determine a cycle. This means the highest frequency of interest is $1/2$ cycles per point. This frequency is called the *folding frequency* and defines the highest frequency that can be seen in discrete sampling. Higher frequencies sampled this way will appear at lower frequencies, called *aliases*. An example is the way a camera samples a rotating wheel on a moving automobile in a movie, in which the wheel appears to be rotating at a different rate. For example, movies are recorded at 24 frames per second. If the camera is filming a wheel that is rotating at the rate of 24 cycles per second (or 24 Hertz), the wheel will appear to stand still; see https://www.youtube.com/watch?v=6XwgbHjRo30.

Consider a generalization of (4.2) that allows mixtures of periodic series with multiple frequencies and amplitudes,

$$x_t = \sum_{k=1}^{q} \left[U_{k1} \cos(2\pi\omega_k t) + U_{k2} \sin(2\pi\omega_k t) \right], \tag{4.4}$$

where U_{k1}, U_{k2}, for $k = 1, 2, \ldots, q$, are independent zero-mean random variables with variances σ_k^2, and the ω_k are distinct frequencies. Notice that (4.4) exhibits the process as a sum of independent components, with variance σ_k^2 for frequency ω_k. As in (4.3), it is easy to show (Problem 4.2) that the autocovariance function of the process is

$$\gamma(h) = \sum_{k=1}^{q} \sigma_k^2 \cos(2\pi\omega_k h), \tag{4.5}$$

and we note the autocovariance function is the sum of periodic components with weights proportional to the variances σ_k^2. Hence, x_t is a mean-zero stationary processes with variance

$$\gamma(0) = \operatorname{var}(x_t) = \sum_{k=1}^{q} \sigma_k^2, \tag{4.6}$$

which exhibits the overall variance as a sum of variances of each of the component parts.

If we could observe $U_{k1} = a_k$ and $U_{k2} = b_k$ for $k = 1, \ldots, q$, then an estimate of the kth variance component, σ_k^2, of $\operatorname{var}(x_t)$, would be the sample

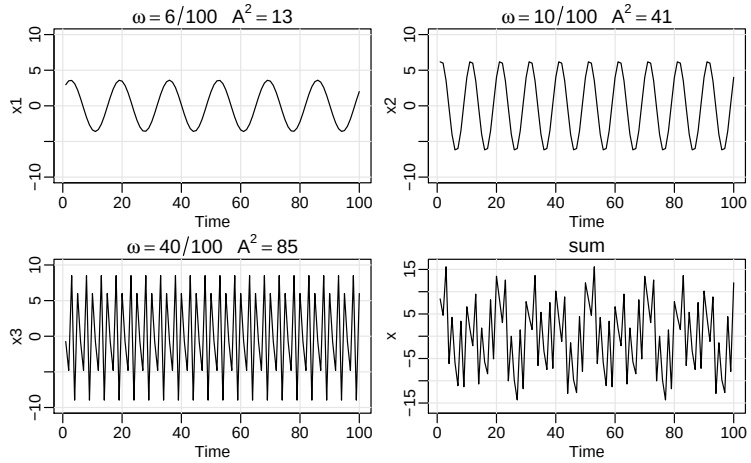


Fig. 4.1. *Periodic components and their sum as described in Example 4.1.*

variance $S_k^2 = a_k^2 + b_k^2$. In addition, an estimate of the total variance of x_t, namely, $\gamma_x(0)$ would be the sum of the sample variances,

$$\hat{\gamma}_x(0) = \widehat{\text{var}}(x_t) = \sum_{k=1}^{q} (a_k^2 + b_k^2)\,. \tag{4.7}$$

Hold on to this idea because we will use it in Example 4.2.

Example 4.1 A Periodic Series

Figure 4.1 shows an example of the mixture (4.4) with $q = 3$ constructed in the following way. First, for $t = 1, \ldots, 100$, we generated three series

$$x_{t1} = 2\cos(2\pi t\, 6/100) + 3\sin(2\pi t\, 6/100)$$
$$x_{t2} = 4\cos(2\pi t\, 10/100) + 5\sin(2\pi t\, 10/100)$$
$$x_{t3} = 6\cos(2\pi t\, 40/100) + 7\sin(2\pi t\, 40/100)$$

These three series are displayed in Figure 4.1 along with the corresponding frequencies and squared amplitudes. For example, the squared amplitude of x_{t1} is $A^2 = 2^2 + 3^2 = 13$. Hence, the maximum and minimum values that x_{t1} will attain are $\pm\sqrt{13} = \pm 3.61$.

Finally, we constructed

$$x_t = x_{t1} + x_{t2} + x_{t3}$$

and this series is also displayed in Figure 4.1. We note that x_t appears to behave as some of the periodic series we have already seen. The systematic sorting out of the essential frequency components in a time series, including their relative contributions, constitutes one of the main objectives of spectral analysis.

The R code to reproduce Figure 4.1 is

```
x1 = 2*cos(2*pi*1:100*6/100)  + 3*sin(2*pi*1:100*6/100)
x2 = 4*cos(2*pi*1:100*10/100) + 5*sin(2*pi*1:100*10/100)
x3 = 6*cos(2*pi*1:100*40/100) + 7*sin(2*pi*1:100*40/100)
x  = x1 + x2 + x3
par(mfrow=c(2,2))
tsplot(x1, ylim=c(-10,10), main=expression(omega==6/100~~~A^2==13))
tsplot(x2, ylim=c(-10,10), main=expression(omega==10/100~~~A^2==41))
```

```
tsplot(x3, ylim=c(-10,10), main=expression(omega==40/100~~~A^2==85))
tsplot(x,  ylim=c(-16,16), main="sum")
```

The model given in (4.4), along with its autocovariance given (4.5), is a population construct. If the model is correct, our next step would be to estimate the variances σ_k^2 and frequencies ω_k that form the model (4.4). In the next example, we consider the problem of estimation of these quantities.

Example 4.2 Estimation and the Periodogram

For any time series sample $x_1, \ldots, x_n$, where n is odd, we may write, *exactly*

$$x_t = a_0 + \sum_{j=1}^{(n-1)/2} \left[a_j \cos(2\pi t\, j/n) + b_j \sin(2\pi t\, j/n)\right], \tag{4.8}$$

This is exact for any sample, x_t, t=1,2,...,n

for $t = 1, \ldots, n$ and suitably chosen coefficients. If n is even, the representation (4.8) can be modified by summing to $(n/2 - 1)$ and adding an additional component given by $a_{n/2}\cos(2\pi t\, \frac{1}{2}) = a_{n/2}(-1)^t$. The crucial point here is that (4.8) is exact for any sample. Hence (4.4) may be thought of as an approximation to (4.8), the idea being that many of the coefficients in (4.8) may be close to zero.

Using the regression results from Chapter 2, the coefficients are a_j and b_j are of the form $\sum_{t=1}^n x_t z_{tj} / \sum_{t=1}^n z_{tj}^2$, where z_{tj} is either $\cos(2\pi t\, j/n)$ or $\sin(2\pi t\, j/n)$. Using Problem 4.22, $\sum_{t=1}^n z_{tj}^2 = n/2$ when $j/n \neq 0, 1/2$, so the regression coefficients in (4.8) can be written as $a_0 = \bar{x}$, and

$$a_j = \frac{2}{n}\sum_{t=1}^n x_t \cos(2\pi t j/n) \quad \text{and} \quad b_j = \frac{2}{n}\sum_{t=1}^n x_t \sin(2\pi t j/n)\,,$$

Note that these can be computed via the Fast Fourier Transform. See my notes.

for $j = 1, \ldots, n$.

Definition 4.1 *We define the* **scaled periodogram** *to be*

$$P(j/n) = a_j^2 + b_j^2, \tag{4.9}$$

because it indicates which frequency components in (4.8) *are large in magnitude and which components are small. The frequencies* $\omega_j = j/n$ *(or* j *cycles in* n *time points) are called the* **Fourier** *or* **fundamental frequencies**.

Large values of $P(j/n)$ indicate which frequencies $\omega_j = j/n$ are predominant in the series, whereas small values of $P(j/n)$ may be associated with noise.

It is not necessary to run a large (saturated) regression to obtain the values of a_j and b_j because they can be computed quickly if n is a highly composite integer. Although we will discuss it in more detail in Section 4.4, the discrete Fourier transform (DFT) is a complex-valued weighted average of the data given by[2]

[2] Useful information from high school math... Euler's formula: $e^{i\alpha} = \cos(\alpha) + i\sin(\alpha)$. Thus, $\cos(\alpha) = \frac{e^{i\alpha}+e^{-i\alpha}}{2}$, and $\sin(\alpha) = \frac{e^{i\alpha}-e^{-i\alpha}}{2i}$. Also, $1/i = -i$ because $-i \times i = 1$. If $z = a + ib$ is complex, then $|z|^2 = z\bar{z} = (a+ib)(a-ib) = a^2 + b^2$.

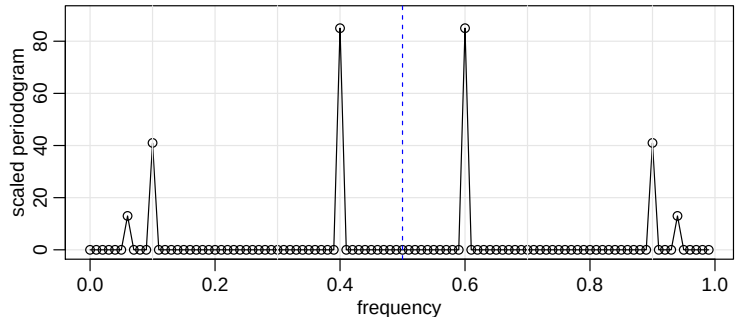


Fig. 4.2. *The scaled periodogram (4.12) of the data generated in Example 4.1.*

$$\begin{aligned} d(j/n) &= n^{-1/2} \sum_{t=1}^{n} x_t \, \mathrm{e}^{-2\pi i t j/n} \\ &= n^{-1/2} \left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n) - i \sum_{t=1}^{n} x_t \sin(2\pi t j/n) \right), \end{aligned} \tag{4.10}$$

for $j = 0, 1, \ldots, n-1$. Because of a large number of redundancies in the calculation, (4.10) may be computed quickly using the fast Fourier transform (FFT). Note that

$$|d(j/n)|^2 = \frac{1}{n} \left(\sum_{t=1}^{n} x_t \cos(2\pi t j/n) \right)^2 + \frac{1}{n} \left(\sum_{t=1}^{n} x_t \sin(2\pi t j/n) \right)^2 \tag{4.11}$$

Periodogram

and it is this quantity that is called the periodogram. We may calculate the scaled periodogram, (4.9), using the periodogram as

$$P(j/n) = \tfrac{4}{n} |d(j/n)|^2 . \tag{4.12}$$

The scaled periodogram of the data, x_t, simulated in Example 4.1 is shown in Figure 4.2, and it clearly identifies the three components x_{t1}, x_{t2}, and x_{t3} of x_t. Note that

$$P(j/n) = P(1 - j/n), \quad j = 0, 1, \ldots, n-1,$$

Only the (n-1)/2 need to be plotted. The rest are equal (in magitude).

so there is a mirroring effect at the folding frequency of ½; consequently, the periodogram is typically not plotted for frequencies higher than the folding frequency. In addition, note that the heights of the scaled periodogram shown in the figure are

$$P(\tfrac{6}{100}) = P(\tfrac{94}{100}) = 13, \quad P(\tfrac{10}{100}) = P(\tfrac{90}{100}) = 41, \quad P(\tfrac{40}{100}) = P(\tfrac{60}{100}) = 85,$$

and $P(j/n) = 0$ otherwise. These are exactly the values of the squared amplitudes of the components generated in Example 4.1.

Assuming the simulated data, `x`, were retained from the previous example, the R code to reproduce Figure 4.2 is

```
n  = length(x)
P  = (4/n) * Mod( fft(x)/sqrt(n) )^2
Fr = 0:99/100
tsplot(Fr, P, type="o", xlab="frequency", ylab="scaled periodogram")
```

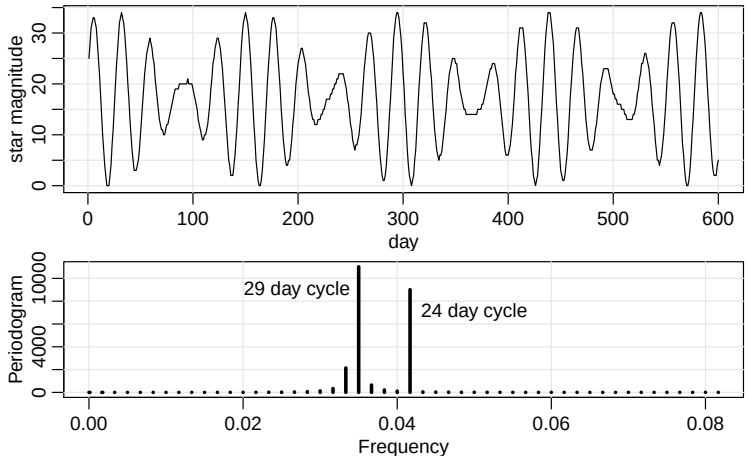
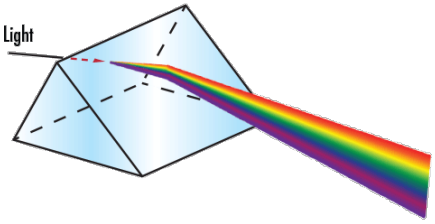


Fig. 4.3. *Star magnitudes and part of the corresponding periodogram.*

Different packages scale the FFT differently, so it is a good idea to consult the documentation. R computes it without the factor $n^{-1/2}$ and with an additional factor of $e^{2\pi i\omega_j}$ that can be ignored because we will be interested in the squared modulus.



If we consider the data x_t in Example 4.1 as a color (waveform) made up of primary colors x_{t1}, x_{t2}, x_{t3} at various strengths (amplitudes), then we might consider the periodogram as a prism that decomposes the color x_t into its primary colors (spectrum). Hence the term *spectral analysis*.

The following is an example using actual data.

Example 4.3 Star Magnitude

The data in Figure 4.3 are the magnitude of a star taken at midnight for 600 consecutive days. The data are taken from the classic text, *The Calculus of Observations, a Treatise on Numerical Mathematics*, by E.T. Whittaker and G. Robinson, (1923, Blackie & Son, Ltd.).

The periodogram for frequencies less than .08 is also displayed in the figure; the periodogram for frequencies higher than .08 are essentially zero. Note that the 29 day cycle and the 25 day cycle are the most prominent periodic components of the data. The R code to reproduce Figure 4.3 is

```
n = length(star)
par(mfrow=c(2,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
tsplot(star, ylab="star magnitude", xlab="day")
Per  = Mod(fft(star-mean(star)))^2/n
Freq = (1:n -1)/n
plot(Freq[1:50], Per[1:50], type='h', lwd=3, ylab="Periodogram",
           xlab="Frequency")
u = which.max(Per[1:50])          # 22  freq=21/600=.035 cycles/day
uu = which.max(Per[1:50][-u])     # 25  freq=25/600=.041 cycles/day
1/Freq[22]; 1/Freq[26]            # period = days/cycle
text(.05, 7000, "24 day cycle"); text(.027, 9000, "29 day cycle")
### another way to find the two peaks is to order on Per
y = cbind(1:50, Freq[1:50], Per[1:50]);  y[order(y[,3]),]
```

The periodogram, which was introduced in Schuster (1898) and used in Schuster (1906) for studying the periodicities in the sunspot series (shown in

Figure 4.21 in the Problems section) is a sample based statistic. In Example 4.2 and Example 4.3, we discussed the fact that the periodogram may be giving us an idea of the variance components associated with each frequency, as presented in (4.6), of a time series. These variance components, however, are population parameters. The concepts of population parameters and sample statistics, as they relate to spectral analysis of time series can be generalized to cover stationary time series and that is the topic of the next section.

4.3 The Spectral Density

Recall Eq. 4.7

The idea that a time series is composed of periodic components appearing in proportion to their underlying variances is fundamental to spectral analysis.

A result called the ***Spectral Representation Theorem***, which is quite technical, states that ***decomposition*** (4.4) ***is approximately true for any stationary time series***.

The examples in the previous section, however, are not generally realistic because time series are rarely exactly of that form (but only approximately of that form). In this section, we deal with a more realistic situation.

Property 4.1 The Spectral Density

If the autocovariance function, $\gamma(h)$, of a stationary process satisfies

This is important for the existence. For example, Eq. 4.5 does not satisfy it.

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty, \tag{4.13}$$

then it has the representation

$$\gamma(h) = \int_{-1/2}^{1/2} e^{2\pi i\omega h} f(\omega)\, d\omega \quad h = 0, \pm 1, \pm 2, \ldots, \tag{4.14}$$

The Fourier transform of the autocorrelation sequence is the Power Spectral Density.

as the inverse transform of the spectral density, which has the representation

$$f(\omega) = \sum_{h=-\infty}^{\infty} \gamma(h) e^{-2\pi i\omega h} \quad -1/2 \le \omega \le 1/2. \tag{4.15}$$

Note that ω here is a CONTINUOUS variable. In the discrete Fourier transform only discrete frequanecies enter, i.e., ω_j=j/n

The examples of the previous section were analogous to probability mass functions, or discrete distributions. In other words, all the weight (support) is on a discrete set of values. The pictures of the periodgram in Figure 4.2 and Figure 4.3 are akin to histograms. The spectral density is the analogue of a probability density function, or of continuous distributions. We note that the absolute summability condition, (4.13), is not satisfied by (4.5), the example that we have used to introduce the idea of a spectral representation. The condition, however, is satisfied for ARMA models.

The autocovariance function is symmetric, $\gamma(h) = \gamma(-h)$, so the spectral density is real-valued. The fact that $\gamma(h)$ is non-negative definite ensures $f(\omega) \ge 0$ for all ω. It follows immediately from (4.15) that

$$f(\omega) = f(-\omega)$$

verifying the spectral density is an even function. Because of the evenness, we will typically only plot $f(\omega)$ for $\omega \geq 0$. In addition, putting $h = 0$ in (4.14) yields

$$\gamma(0) = \mathrm{var}(x_t) = \int_{-1/2}^{1/2} f(\omega)\, d\omega,$$

The integral, i.e., area under f(ω), is equal to the variance of the stationary process.

which expresses the total variance as the integrated spectral density over all of the frequencies. This result shows that the spectral density is a density, not a probability density, but a variance density. We will explore this idea further as we proceed.

It is illuminating to examine the spectral density for the series that we have looked at in earlier discussions.

Example 4.4 White Noise Series

As a simple example, consider the theoretical power spectrum of a sequence of uncorrelated random variables, w_t, with variance σ_w^2. A simulated set of data is displayed in the top of Figure 1.7. Because the autocovariance function was computed in Example 1.14 as $\gamma_w(h) = \sigma_w^2$ for $h = 0$, and zero, otherwise, it follows from (4.15), that

$$f_w(\omega) = \sigma_w^2$$

All frequencies, low as well as high contribute equally to the power spectrum

for $-1/2 \leq \omega \leq 1/2$. Hence the process contains equal power at all frequencies. This property is seen in the realization, which seems to contain all different frequencies in a roughly equal mix. In fact, the name white noise comes from the analogy to white light, which contains all frequencies in the color spectrum at the same level of intensity. Figure 4.4 shows a plot of the white noise spectrum for $\sigma_w^2 = 1$.

If x_t is ARMA, its spectral density can be obtained explicitly using the fact that it is a linear process, i.e., $x_t = \sum_{j=0}^{\infty} \psi_j w_{t-j}$, where $\sum_{j=0}^{\infty} |\psi_j| < \infty$. In the following property, we exhibit the form of the spectral density of an ARMA model. The proof of the property follows directly from the proof of a more general result, Property 4.4, by using the additional fact that $\psi(z) = \theta(z)/\phi(z)$. The result is analogous to the fact that if $X = aY$, then $\mathrm{var}(X) = a^2\mathrm{var}(Y)$.

Property 4.2 The Spectral Density of ARMA

If x_t is ARMA(p, q), $\phi(B)x_t = \theta(B)w_t$, its spectral density is given by

$$f_x(\omega) = \sigma_w^2 |\psi(\mathrm{e}^{-2\pi i\omega})|^2 = \sigma_w^2 \frac{|\theta(\mathrm{e}^{-2\pi i\omega})|^2}{|\phi(\mathrm{e}^{-2\pi i\omega})|^2} \qquad (4.16)$$

The proof of it later on.

where $\phi(z) = 1 - \sum_{k=1}^{p} \phi_k z^k$, $\theta(z) = 1 + \sum_{k=1}^{q} \theta_k z^k$, and $\psi(z) = \sum_{k=0}^{\infty} \psi_k z^k$.

Example 4.5 Moving Average

As an example of a series that does not have an equal mix of frequencies, we consider a moving average model. Specifically, consider the MA(1) model given by

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

A sample realization with $\theta = \pm.9$ was shown in Figure 3.2. Note the realization with positive θ has less of the higher or faster frequencies. The spectral density will verify this observation.

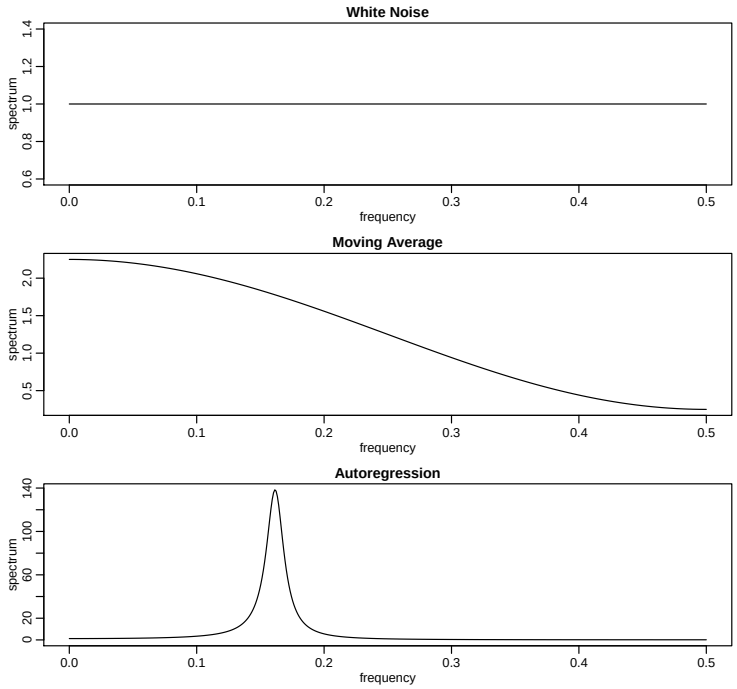


Fig. 4.4. *Theoretical spectra of white noise (top), a first-order moving average (middle), and a second-order autoregressive process (bottom).*

The autocovariance function is displayed in Example 3.3, and for this particular example, we have

$$\gamma(0) = (1 + .5^2)\sigma_w^2 = 1.25\sigma_w^2; \quad \gamma(\pm 1) = .5\sigma_w^2; \quad \gamma(\pm h) = 0 \text{ for } h > 1.$$

Substituting this directly into the definition given in (4.15), we have

$$\begin{aligned} f(\omega) &= \sum_{h=-\infty}^{\infty} \gamma(h)\, \mathrm{e}^{-2\pi i\omega h} = \sigma_w^2 \left[1.25 + .5\left(\mathrm{e}^{-2\pi i\omega} + \mathrm{e}^{2\pi i\omega}\right)\right] \\ &= \sigma_w^2 \left[1.25 + \cos(2\pi\omega)\right]. \end{aligned} \tag{4.17}$$

We can also compute the spectral density using Property 4.2, which states that for an MA, $f(\omega) = \sigma_w^2 |\theta(\mathrm{e}^{-2\pi i\omega})|^2$. Because $\theta(z) = 1 + .5z$, we have

$$\begin{aligned} |\theta(\mathrm{e}^{-2\pi i\omega})|^2 &= |1 + .5\mathrm{e}^{-2\pi i\omega}|^2 = (1 + .5\mathrm{e}^{-2\pi i\omega})(1 + .5\mathrm{e}^{2\pi i\omega}) \\ &= 1 + .5\mathrm{e}^{-2\pi i\omega} + .5\mathrm{e}^{2\pi i\omega} + .25\,\mathrm{e}^{-2\pi i\omega} \cdot \mathrm{e}^{2\pi i\omega} \\ &= 1.25 + .5\left(\mathrm{e}^{-2\pi i\omega} + \mathrm{e}^{2\pi i\omega}\right) \end{aligned}$$

which leads to agreement with (4.17).

Plotting the spectrum for $\sigma_w^2 = 1$, as in the middle of Figure 4.4, shows the lower or slower frequencies have greater power than the higher or faster frequencies.

Example 4.6 A Second-Order Autoregressive Series

We now consider the spectrum of an AR(2) series of the form

$$x_t - \phi_1 x_{t-1} - \phi_2 x_{t-2} = w_t,$$

for the special case $\phi_1 = 1$ and $\phi_2 = -.9$. Figure 1.8 shows a sample realization of such a process for $\sigma_w = 1$. We note the data exhibit a strong periodic component that makes a cycle about every six points.

To use Property 4.2, note that $\theta(z) = 1$, $\phi(z) = 1 - z + .9z^2$ and

$$\begin{aligned} |\phi(e^{-2\pi i\omega})|^2 &= (1 - \mathrm{e}^{-2\pi i\omega} + .9\mathrm{e}^{-4\pi i\omega})(1 - \mathrm{e}^{2\pi i\omega} + .9\mathrm{e}^{4\pi i\omega}) \\ &= 2.81 - 1.9(\mathrm{e}^{2\pi i\omega} + \mathrm{e}^{-2\pi i\omega}) + .9(\mathrm{e}^{4\pi i\omega} + \mathrm{e}^{-4\pi i\omega}) \\ &= 2.81 - 3.8\cos(2\pi\omega) + 1.8\cos(4\pi\omega). \end{aligned}$$

Using this result in (4.16), we have that the spectral density of x_t is

$$f_x(\omega) = \frac{\sigma_w^2}{2.81 - 3.8\cos(2\pi\omega) + 1.8\cos(4\pi\omega)}.$$

Setting $\sigma_w = 1$, the bottom of Figure 4.4 displays $f_x(\omega)$ and shows a strong power component at about $\omega = .16$ cycles per point or a period between six and seven cycles per point and very little power at other frequencies. In this case, the series is nearly sinusoidal, but not exact, which seems more realistic for actual data.

To reproduce Figure 4.4, use the `arma.spec` script from `astsa`:

```
par(mfrow=c(3,1))
arma.spec(log="no", main="White Noise")
arma.spec(ma=.5, log="no", main="Moving Average")
arma.spec(ar=c(1,-.9), log="no", main="Autoregression")
```

Play with the values of the parameters and plot the corresponding PSDs.

The above examples motivate the use of the power spectrum for describing the theoretical variance fluctuations of a stationary time series. The plot of the function $f(\omega)$ over the frequency argument ω can even be thought of as an analysis of variance, in which the effects are the frequencies, indexed by ω.

4.4 Periodogram and Discrete Fourier Transform

We are now ready to tie together the periodogram, which is the sample-based concept presented in Section 4.2, with the spectral density, which is the population-based concept of Section 4.3.

Definition 4.2 *Given data $x_1, \ldots, x_n$, we define the* **discrete Fourier transform (DFT)** *to be*

$$d(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \, \mathrm{e}^{-2\pi i \omega_j t} \tag{4.18}$$

for $j = 0, 1, \ldots, n-1$, where the frequencies $\omega_j = j/n$ are the Fourier or fundamental frequencies.

If n is a highly composite integer (i.e., it has many factors), the DFT can be computed by the fast Fourier transform (FFT) introduced in Cooley and Tukey (1965). Sometimes it is helpful to exploit the inversion result for DFTs which shows the linear transformation is one-to-one. For the inverse DFT we have,

$$x_t = n^{-1/2} \sum_{j=0}^{n-1} d(\omega_j)\, \mathrm{e}^{2\pi i \omega_j t} \tag{4.19}$$

for $t = 1, \ldots, n$. The following example shows how to calculate the DFT and its inverse in R for the data set $\{1, 2, 3, 4\}$; note that R writes a complex number $z = a + ib$ as `a+bi`.

```
(dft = fft(1:4)/sqrt(4))
  [1]  5+0i  -1+1i  -1+0i  -1-1i
(idft = fft(dft, inverse=TRUE)/sqrt(4))
  [1] 1+0i  2+0i  3+0i  4+0i
(Re(idft))  # keep it real
  [1] 1  2  3  4
```

We now define the periodogram as the squared modulus[3] of the DFT.

Definition 4.3 *Given data $x_1, \ldots, x_n$, we define the* **periodogram** *to be*

$$I(\omega_j) = \left|d(\omega_j)\right|^2 \tag{4.20}$$

for $j = 0, 1, 2, \ldots, n-1$.

Note that $I(0) = n\bar{x}^2$, where $\bar{x}$ is the sample mean. Also, for $j \neq 0$,[4]

$$\begin{aligned} I(\omega_j) = \left|d(\omega_j)\right|^2 &= n^{-1} \sum_{t=1}^{n} \sum_{s=1}^{n} (x_t - \bar{x})(x_s - \bar{x}) e^{-2\pi i \omega_j (t-s)} \\ &\overset{h=t-s}{=} n^{-1} \sum_{h=-(n-1)}^{n-1} \sum_{t=1}^{n-|h|} (x_{t+|h|} - \bar{x})(x_t - \bar{x}) e^{-2\pi i \omega_j h} \\ &= \sum_{h=-(n-1)}^{n-1} \widehat{\gamma}(h) e^{-2\pi i \omega_j h}. \end{aligned} \tag{4.21}$$

The periodogram is the sampled version of the PSD

In view of (4.21), the periodogram, $I(\omega_j)$, is the sample version of $f(\omega_j)$ given in (4.15). That is, we may think of the periodogram as the "sample spectral density" of x_t. Although (4.21) seems like a reasonable estimate of $f(\omega)$, recall from Example 4.2 that $I(\omega_j)$, for any j, is based on only 2 pieces of information (degrees of freedom).

It is sometimes useful to work with the real and imaginary parts of the DFT individually. To this end, we define the following transforms.

Definition 4.4 *Given data $x_1, \ldots, x_n$, we define the* **cosine transform**

$$d_c(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \cos(2\pi\omega_j t) \tag{4.22}$$

and the **sine transform**

[3] Recall that if $z = a + ib$, then $\bar{z} = a - ib$, and $|z|^2 = z\bar{z} = a^2 + b^2$.

[4] In this case, the DFTs of x_t and of $(x_t - \bar{x})$ are the same because $\sum_{t=1}^{n} \mathrm{e}^{-2\pi i t j/n} = 0$.

$$d_s(\omega_j) = n^{-1/2} \sum_{t=1}^{n} x_t \sin(2\pi\omega_j t) \tag{4.23}$$

where $\omega_j = j/n$ *for* $j = 0, 1, \ldots, n-1$.

Note that $d_c(\omega_j)$ and $d_s(\omega_j)$ are averages like the sample mean, but with difference weights (the sample mean has weights $1/n$ for each observation). Under appropriate conditions, there is central limit theorem for these quantities. In non-technical terms, the result is similar to the central limit theorem for sample means, that is,

$$d_c(\omega_j) \dot{\sim} \mathrm{N}(0, \tfrac{1}{2} f(\omega_j)) \quad \text{and} \quad d_s(\omega_j) \dot{\sim} \mathrm{N}(0, \tfrac{1}{2} f(\omega_j)) \tag{4.24}$$

where $\dot{\sim}$ means *approximately distributed as* for n large. Moreover, it can be shown that for large n, $d_c(\omega_j) \perp d_s(\omega_j) \perp d_c(\omega_k) \perp d_s(\omega_k)$, as long as $\omega_j \neq \omega_k$, where $\perp$ is read *is independent of*.

We note that $d(\omega_j) = d_c(\omega_j) - i\, d_s(\omega_j)$ and hence the periodogram is

$$I(\omega_j) = d_c^2(\omega_j) + d_s^2(\omega_j), \tag{4.25}$$

which for large n is the sum of the squares of two independent normal random variables, which we know has a chi-squared (χ^2) distribution. Thus, for large samples, $I(\omega_j) \dot{\sim} \frac{1}{2} f(\omega_j) \chi^2_2$, or equivalently,

$$\frac{2\, I(\omega_j)}{f(\omega_j)} \dot{\sim} \chi^2_2, \tag{4.26}$$

where χ^2_2 is the chi-squared distribution with 2 degrees of freedom. Since the mean and variance of a χ^2_ν are ν and 2ν, respectively, it follows from (4.26) that

$$E[I(\omega_j)] \approx f(\omega_j) \quad \text{and} \quad \mathrm{var}[I(\omega_j)] \approx f^2(\omega_j). \tag{4.27}$$

The periodogram is unbiased but not consistent etimator of the PSD

This is bad news because, while the periodgram is approximately unbiased, its variance does not go to zero, and hence it is not consistent. In fact, no matter how large n, the variance of the periodogram does not change. Contrast this with the mean $\bar{x}$ of a random sample of size n for which $E[\bar{x}] = \mu$ and $\mathrm{var}[\bar{x}] = \sigma^2/n \to 0$ as $n \to \infty$.

The distributional result (4.26) can be used to derive an approximate confidence interval for the spectrum in the usual way. Let $\chi^2_\nu(\alpha)$ denote the lower α probability tail for the chi-squared distribution with ν degrees of freedom. Then, an approximate $100(1-\alpha)\%$ confidence interval for the spectral density function would be of the form

$$\frac{2\, I(\omega_j)}{\chi^2_2(1-\alpha/2)} \leq f(\omega) \leq \frac{2\, I(\omega_j)}{\chi^2_2(\alpha/2)}. \tag{4.28}$$

The log transform is the variance stabilizing transformation. In this case, the confidence intervals are of the form

$$\begin{aligned} &\big[\log I(\omega_j) + \log 2 - \log \chi^2_2(1-\alpha/2), \\ &\qquad \log I(\omega_j) + \log 2 - \log \chi^2_2(\alpha/2)\big]. \end{aligned} \tag{4.29}$$

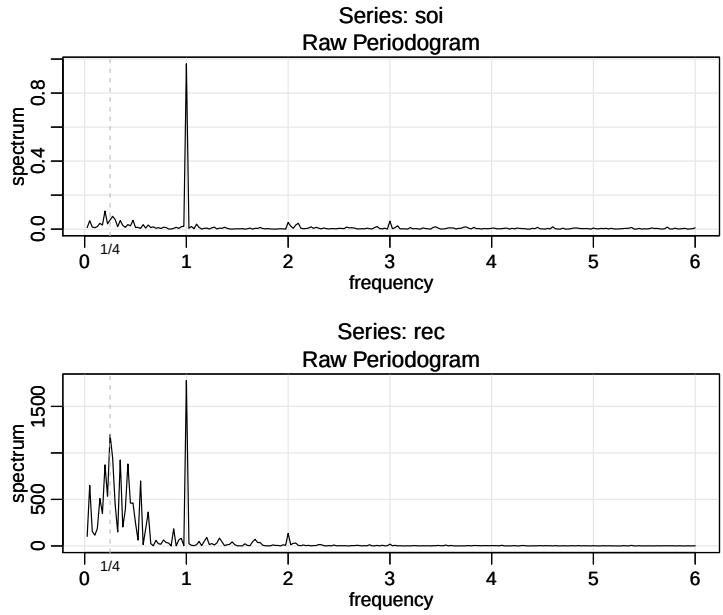


Fig. 4.5. *Periodogram of SOI and Recruitment, $n = 453$ ($n' = 480$), where the frequency axis is labeled in multiples of years. Note the common peaks at $\omega = 1$, or one cycle per year, and some larger values near $\omega = 1/4$, or one cycle every four years.*

Often, nonstationary trends are present that should be eliminated before computing the periodogram. Trends introduce extremely low frequency components in the periodogram that tend to obscure the appearance at higher frequencies. For this reason, it is usually conventional to center the data prior to a spectral analysis using either mean-adjusted data of the form $x_t - \bar{x}$ to eliminate the zero or d-c component or to use detrended data of the form $x_t - \widehat{\beta}_1 - \widehat{\beta}_2 t$. Note that higher order polynomial regressions in t or nonparametric smoothing (linear filtering) could be used in cases where the trend is nonlinear.

It is good practice to center the data prior to their use.

As previously indicated, it is often convenient to calculate the DFTs, and hence the periodogram, using the fast Fourier transform algorithm. The FFT utilizes a number of redundancies in the calculation of the DFT when n is highly composite; that is, an integer with many factors of 2, 3, or 5, the best case being when $n = 2^p$ is a factor of 2. Details may be found in Cooley and Tukey (1965). To accommodate this property, we can pad the centered (or detrended) data of length n to the next highly composite integer n' by adding zeros, i.e., setting $x^c_{n+1} = x^c_{n+2} = \cdots = x^c_{n'} = 0$, where x^c_t denotes the centered data. This means that the fundamental frequency ordinates will be $\omega_j = j/n'$ instead of j/n. We illustrate by considering the periodogram of the SOI and Recruitment series, as has been given in Figure 1.5

FFT works with data whose number is power of two

. Recall that they are monthly series and $n = 453$ months. To find n' in R, use the command `nextn(453)` to see that $n' = 480$ will be used in the spectral analyses by default.

Example 4.7 Periodogram of SOI and Recruitment Series

Figure 4.5 shows the periodograms of each series, where the frequency axis is labeled in multiples of years. As previously indicated, the centered data have

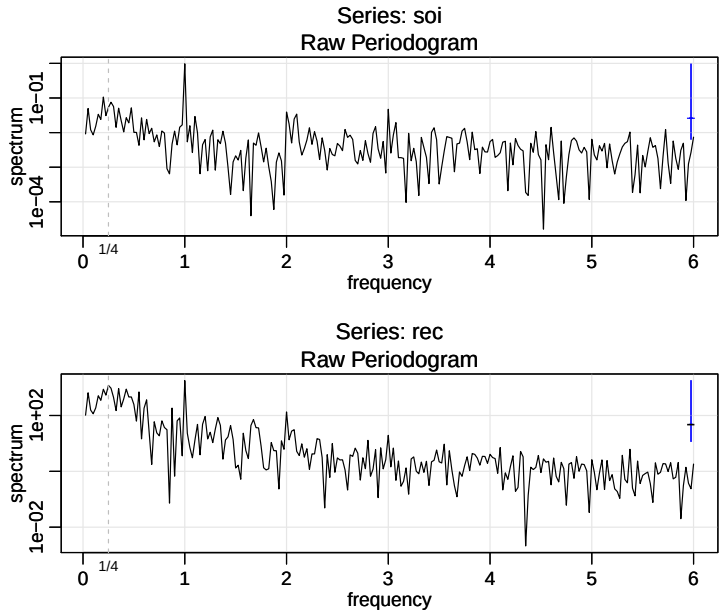


Fig. 4.6. *Log-periodogram of SOI and Recruitment. 95% confidence intervals are indicated by the blue line in the upper right corner. Imagine placing the horizontal tick mark on the log-periodogram ordinate at a desired frequency; the vertical line then gives the interval.*

been padded to a series of length 480. We notice a narrow-band peak at the obvious yearly cycle, $\omega = 1$. In addition, there is considerable power in a wide band at the lower frequencies that is centered around the four-year cycle $\omega = 1/4$ representing a possible El Niño effect. This wide band activity suggests that the possible El Niño cycle is irregular, but tends to be around four years on average. We will continue to address this problem as we move to more sophisticated analyses.

We can pull confidence intervals from the R object, but plotting the spectra on a log scale will also produce a generic interval as seen in Figure 4.6. Notice that, because there are only 2 degrees of freedom at each frequency, the generic confidence interval is too wide to be of much use. We will address this problem next.

The R code is as follows (remove `log="no"` to plot on a log scale):

```
par(mfrow=c(2,1))     # raw periodogram
mvspec(soi, log="no");  abline(v=1/4, lty="dotted")
mvspec(rec, log="no");  abline(v=1/4, lty="dotted")
```

The periodogram is susceptible to LARGE UNCERTAINTIES

It's clear that the periodogram as an estimator is susceptible to large uncertainties. This happens because the periodogram has only two df for any sample The solution to this dilemma is smoothing. As an analogy to using the periodogram to estimate the spectral density, consider the problem of taking a random sample and then trying to estimate a probability density based on a histogram with many cells. This approach is demonstrated in Figure 4.7.

4.5 Nonparametric Spectral Estimation

To continue the discussion that ended the previous section, we introduce a frequency band, $\mathcal{B}$, of $L \ll n$ contiguous fundamental frequencies, centered around frequency $\omega_j = j/n$, which is chosen close to a frequency of interest, ω. Let

$$\mathcal{B} = \{\omega_j + k/n : \ k = 0, \pm 1, \ldots, \pm m\}, \tag{4.30}$$

Frequency Band of L samples around each frequancy bin, ω_j

where

$$L = 2m + 1 \tag{4.31}$$

is an odd number, chosen such that the spectral values in the interval $\mathcal{B}$,

$$f(\omega_j + k/n), \quad k = -m, \ldots, 0, \ldots, m$$

are approximately equal to $f(\omega)$. For example, to see a small section of the AR(2) spectrum—near the peak—shown in Figure 4.4, use

```
arma.spec(ar=c(1,-.9), xlim=c(.15,.151), n.freq=100000)
```

which is displayed in Figure 4.8.

We now define an averaged (or smoothed) periodogram as the average of the periodogram values, say,

$$\bar{f}(\omega) = \frac{1}{L} \sum_{k=-m}^{m} I(\omega_j + k/n), \tag{4.32}$$

Smoothed Periodogram

over the band $\mathcal{B}$. Under the assumption that the spectral density is fairly constant in the band $\mathcal{B}$, and in view of the discussion around (4.24), we can show that the

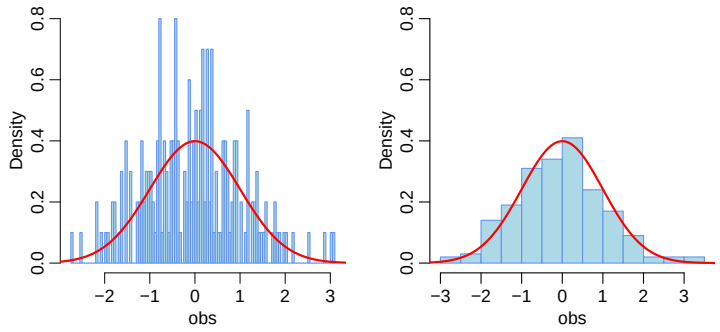


Fig. 4.7. Left: *Histogram of* 200 *standard normals with 100 cells and with the standard normal density superimposed. The periodogram is to the spectral density as the histogram is to the normal density.* Right: *Histogram of the same data with much wider cells.*

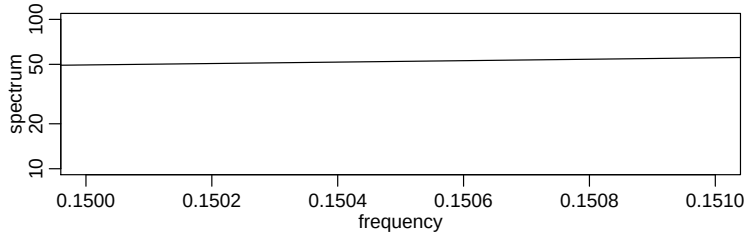


Fig. 4.8. *A small section (near the peak) of the AR(2) spectrum shown in Figure 4.4 .*

periodograms in (4.32) are approximately distributed as independent $f(\omega)\chi^2_2/2$ random variables, as long as we keep L fairly small relative to n. Thus, under appropriate conditions, $L\bar{f}(\omega)$ is the sum of L approximately independent $f(\omega)\chi^2_2/2$ random variables. It follows that, for large n,

$$\frac{2L\bar{f}(\omega)}{f(\omega)} \sim \chi^2_{2L}. \tag{4.33}$$

Now we have

$$E[\bar{f}(\omega)] \approx f(\omega) \quad \text{and} \quad \text{var}[\bar{f}(\omega)] \approx f^2(\omega)/L, \tag{4.34}$$

which can be compared to (4.27). In this case, we have consistency if we let $L \to \infty$ as $n \to \infty$, but L must grow much slower than n, of course.

In this scenario, where we smooth the periodogram by simple averaging, the width of the frequency interval defined by (4.30),

$$\mathrm{B} = \frac{L}{n} \tag{4.35}$$

Bandwidth

is called the *bandwidth*. The result (4.33) can be rearranged to obtain an approximate $100(1-\alpha)\%$ confidence interval of the form

$$\frac{2L\bar{f}(\omega)}{\chi^2_{2L}(1-\alpha/2)} \le f(\omega) \le \frac{2L\bar{f}(\omega)}{\chi^2_{2L}(\alpha/2)} \tag{4.36}$$

for the true spectrum, $f(\omega)$.

As previously discussed, the visual impact of a spectral density plot will be improved by plotting the logarithm of the spectrum, which is the variance stabilizing transformation in this situation. This phenomenon can occur when regions of the spectrum exist with peaks of interest much smaller than some of the main power components. For the log spectrum, we obtain an interval of the form

$$\big[\log\bar{f}(\omega) + \log 2L - \log\chi^2_{2L}(1-\alpha/2), \\ \log\bar{f}(\omega) + \log 2L - \log\chi^2_{2L}(\alpha/2)\big]. \tag{4.37}$$

If zeros are appended before computing the spectral estimators, we need to adjust the degrees of freedom because you can't get something for nothing (unless your dad is rich). An approximation that works well is to replace $2L$ by $2Ln/n'$. Hence, we define the adjusted degrees of freedom as

$$df = \frac{2Ln}{n'} \tag{4.38}$$

and use it instead of $2L$ in the confidence intervals (4.36) and (4.37). For example, (4.36) becomes

$$\frac{df\,\bar{f}(\omega)}{\chi^2_{df}(1-\alpha/2)} \le f(\omega) \le \frac{df\,\bar{f}(\omega)}{\chi^2_{df}(\alpha/2)}. \tag{4.39}$$

Before proceeding further, we pause to consider computing the average periodograms for the SOI and Recruitment series, as shown in Figure 4.9.

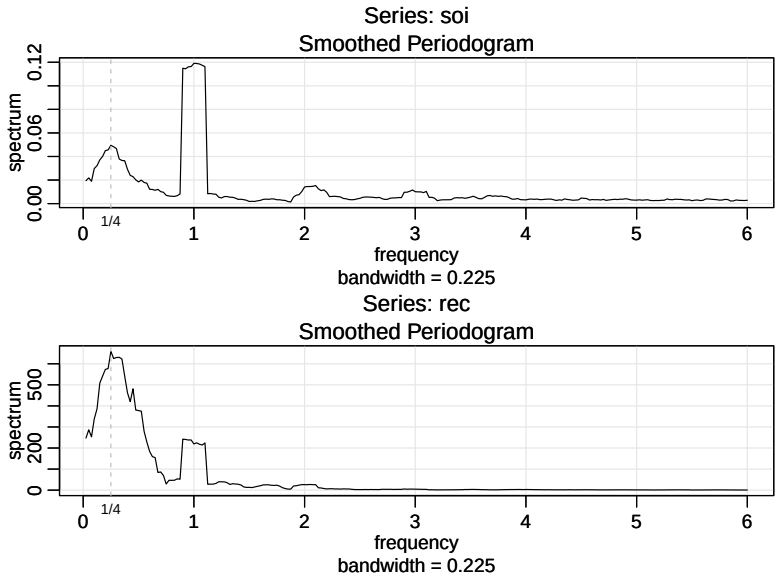


Fig. 4.9. *The averaged periodogram of the SOI and Recruitment series* $n = 453$, $n' = 480$, $L = 9$, $df = 17$, *showing common peaks at the four year period,* $\omega = {}^{1\,\text{cycle}}/_{4\,\text{years}}$, *the yearly period,* $\omega = {}^{1\,\text{cycle}}/_{\text{year}}$ *and some of its harmonics* $\omega = k$ *for* $k = 2, 3$.

Example 4.8 Averaged Periodogram for SOI and Recruitment

Generally, it is a good idea to try several bandwidths that seem to be compatible with the general overall shape of the spectrum, as suggested by the periodogram. The SOI and Recruitment series periodograms, previously computed in Figure 4.5, suggest the power in the lower El Niño frequency needs smoothing to identify the predominant overall period. Trying values of L leads to the choice $L = 9$ as a reasonable value, and the result is displayed in Figure 4.9.

The smoothed spectra shown in Figure 4.9 provide a sensible compromise between the noisy version, shown in Figure 4.5, and a more heavily smoothed spectrum, which might lose some of the peaks. An undesirable effect of averaging can be noticed at the yearly cycle, $\omega = 1$, where the narrow band peaks that appeared in the periodograms in Figure 4.5 have been flattened and spread out to nearby frequencies. We also notice, and have marked, the appearance of harmonics of the yearly cycle, that is, frequencies of the form $\omega = k$ for $k = 1, 2, \ldots$. Harmonics typically occur when a periodic component is present, but not in a sinusoidal fashion; see Example 4.9.

Figure 4.9 can be reproduced in R using the following commands. To compute averaged periodograms, use the Daniell kernel, and specify m, where $L = 2m + 1$ ($L = 9$ and $m = 4$ in this example). We will explain the kernel concept later in this section, specifically just prior to Example 4.10.

```
par(mfrow=c(2,1))
(k = kernel("daniell", 4))
soi.ave = mvspec(soi, k, log="no"); abline(v=c(.25,1,2,3), lty=2)
rec.ave = mvspec(rec, k, log="no"); abline(v=c(.25,1,2,3), lty=2)
soi.ave$bandwidth       # = 0.225
soi.ave$df              # = 16.9875
```

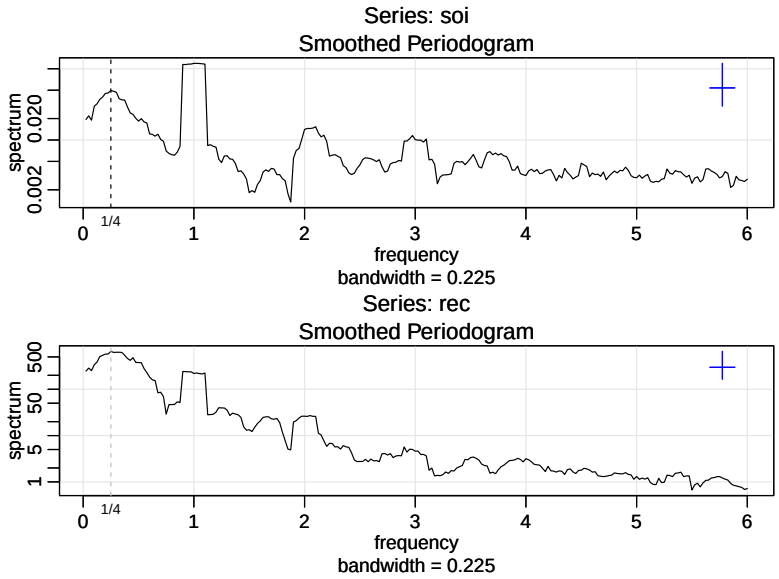


Fig. 4.10. *Figure 4.9 with the average periodogram ordinates plotted on a* $\log_{10}$ *scale. The display in the upper right-hand corner represents a generic 95% confidence interval.*

The displayed bandwidth (.225) is adjusted for the fact that the frequency scale of the plot is in terms of cycles per year instead of cycles per month (the original unit of the data). Using (4.35), the bandwidth in terms of months is $9/480 = .01875$; the displayed value is simply converted to years, $.01875 \frac{cycles}{month} \times 12 \frac{months}{year} = .225 \frac{cycles}{year}$.

The adjusted degrees of freedom are $df = 2(9)(453)/480 \approx 17$. We can use this value for the 95% confidence intervals shown in Figure 4.10, which are the spectral estimates on a log scale. For the two frequency bands identified as having the maximum power, we may look at the 95% confidence intervals and see whether the lower limits are substantially larger than adjacent baseline spectral levels. For example, peak at the El Niño period of 4 years has lower limits that exceed the values the spectrum would have if there were simply a smooth underlying spectral function without the peaks.

Example 4.9 Harmonics

In the previous example, we saw that the spectra of the annual signals displayed minor peaks at the harmonics. That is, there was a a large peak at $\omega = 1$ cycles/year and minor peaks at its harmonics $\omega = k$ for $k = 2, 3, \ldots$ (two-, three-, and so on, cycles per year). This will often be the case because most signals are not perfect sinusoids (or perfectly cyclic). In this case, the harmonics are needed to capture the non-sinusoidal behavior of the signal. As an example, consider the signal formed in Figure 4.11 from a (fundamental) sinusoid oscillating at two cycles per unit time along with the second through sixth harmonics at decreasing amplitudes. In particular, the signal was formed as

$$x_t = \sin(2\pi 2t) + .5\sin(2\pi 4t) + .4\sin(2\pi 6t) + .3\sin(2\pi 8t) + .2\sin(2\pi 10t) + .1\sin(2\pi 12t) \qquad (4.40)$$

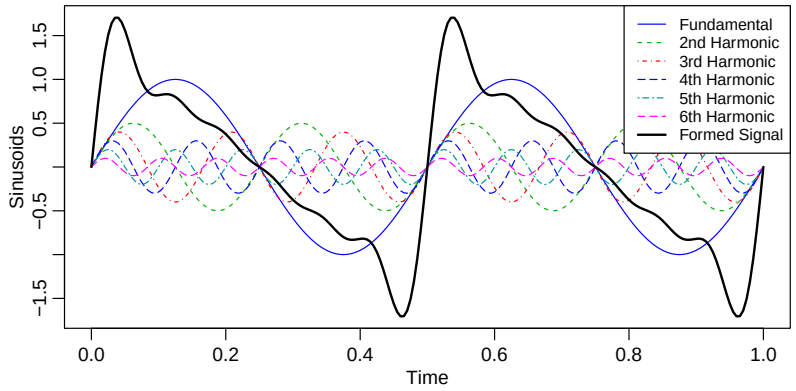


Fig. 4.11. *A signal (thick solid line) formed by a fundamental sinusoid (thin solid line) oscillating at two cycles per unit time and its harmonics as specified in* (4.40)*.*

for $0 \leq t \leq 1$. Notice that the signal is non-sinusoidal in appearance and rises quickly then falls slowly. The code for Figure 4.11 is:

```
t = seq(0, 1, by=1/200)
amps = c(1, .5, .4, .3, .2, .1)
x = matrix(0, 201, 6)
for (j in 1:6) x[,j] = amps[j]*sin(2*pi*t*2*j)
x = ts(cbind(x, rowSums(x)), start=0, deltat=1/200)
ts.plot(x, lty=c(1:6, 1), lwd=c(rep(1,6), 2), ylab="Sinusoids")
names = c("Fundamental","2nd Harmonic","3rd Harmonic","4th Harmonic", "5th
           Harmonic", "6th Harmonic", "Formed Signal")
legend("topright", names, lty=c(1:6, 1), lwd=c(rep(1,6), 2) )
```

Example 4.8 points out the necessity for having some relatively systematic procedure for deciding whether peaks are significant. The question of deciding whether a single peak is significant usually rests on establishing what we might think of as a baseline level for the spectrum, defined rather loosely as the shape that one would expect to see if no spectral peaks were present. This profile can usually be guessed by looking at the overall shape of the spectrum that includes the peaks; usually, a kind of baseline level will be apparent, with the peaks seeming to emerge from this baseline level. If the lower confidence limit for the spectral value is still greater than the baseline level at some predetermined level of significance, we may claim that frequency value as a statistically significant peak. To be consistent with our stated indifference to the upper limits, we might use a one-sided confidence interval.

Care must be taken when we make a decision about the bandwidth B over which the spectrum will be essentially constant. Taking too broad a band will tend to smooth out valid peaks in the data when the constant variance assumption is not met over the band. Taking too narrow a band will lead to confidence intervals so wide that peaks are no longer statistically significant. Thus, we note that there is a conflict here between variance properties or bandwidth stability, which can be improved by increasing B and resolution, which can be improved by decreasing B. A common approach is to try a number of different bandwidths and to look qualitatively at the spectral estimators for each case.

To address the problem of resolution, it should be evident that the flattening of the peaks in Figure 4.9 and Figure 4.10 was due to the fact that simple

averaging was used in computing $\bar{f}(\omega)$ defined in (4.32). There is no particular reason to use simple averaging, and we might improve the estimator by employing a weighted average, say

$$\widehat{f}(\omega) = \sum_{k=-m}^{m} h_k I(\omega_j + k/n), \tag{4.41}$$

using the same definitions as in (4.32) but where the weights $h_k > 0$ satisfy

$$\sum_{k=-m}^{m} h_k = 1.$$

In particular, it seems reasonable that the resolution of the estimator will improve if we use weights that decrease as distance from the center weight h_0 increases; we will return to this idea shortly. To obtain the averaged periodogram, $\bar{f}(\omega)$, in (4.41), set $h_k = L^{-1}$, for all k, where $L = 2m + 1$. The large sample theory established for $\bar{f}(\omega)$ still holds for $\widehat{f}(\omega)$ provided that the weights satisfy the additional condition that if $m \to \infty$ as $n \to \infty$ but $m/n \to 0$, then

$$\sum_{k=-m}^{m} h_k^2 \to 0.$$

Under these conditions, for n large, we have

$$E[\widehat{f}(\omega)] \approx f(\omega) \quad \text{and} \quad \text{var}[\widehat{f}(\omega)] \approx f^2(\omega) \sum_{k=-m}^{m} h_k^2 \tag{4.42}$$

which can be compared to (4.34); as before, we have that $\widehat{f}(\omega)$ is consistent. We have already seen this result in the case of $\bar{f}(\omega)$, where the weights are constant, $h_k = L^{-1}$, in which case $\sum_{k=-m}^{m} h_k^2 = L^{-1}$. The distributional properties of (4.41) are more difficult now because $\widehat{f}(\omega)$ is a weighted linear combination of asymptotically independent χ^2 random variables. An approximation that seems to work well is to replace L by $\left(\sum_{k=-m}^{m} h_k^2\right)^{-1}$. That is, define

$$L_h = \left(\sum_{k=-m}^{m} h_k^2 \right)^{-1} \tag{4.43}$$

and use the approximation

$$\frac{2L_h \widehat{f}(\omega)}{f(\omega)} \overset{\cdot}{\sim} \chi^2_{2L_h}. \tag{4.44}$$

In analogy to (4.35), we will define the bandwidth in this case to be

$$\mathrm{B} = \frac{L_h}{n}. \tag{4.45}$$

Using the approximation (4.44) we obtain an approximate $100(1-\alpha)\%$ confidence interval of the form

$$\frac{2L_h \widehat{f}(\omega)}{\chi^2_{2L_h}(1-\alpha/2)} \le f(\omega) \le \frac{2L_h \widehat{f}(\omega)}{\chi^2_{2L_h}(\alpha/2)} \tag{4.46}$$

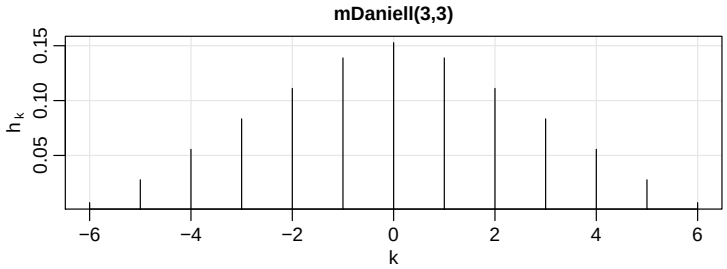


Fig. 4.12. *Modified Daniell kernel weights used in Example 4.10*

for the true spectrum, $f(\omega)$. If the data are padded to n', then replace $2L_h$ in (4.46) with $df = 2L_h n/n'$ as in (4.38).

An easy way to generate the weights in R is by repeated use of the Daniell kernel. For example, with $m = 1$ and $L = 2m + 1 = 3$, the Daniell kernel has weights $\{h_k\} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$; applying this kernel to a sequence of numbers, $\{u_t\}$, produces

$$\widehat{u}_t = \tfrac{1}{3}u_{t-1} + \tfrac{1}{3}u_t + \tfrac{1}{3}u_{t+1}.$$

We can apply the same kernel again to the $\widehat{u}_t$,

$$\widehat{\widehat{u}}_t = \tfrac{1}{3}\widehat{u}_{t-1} + \tfrac{1}{3}\widehat{u}_t + \tfrac{1}{3}\widehat{u}_{t+1},$$

which simplifies to

$$\widehat{\widehat{u}}_t = \tfrac{1}{9}u_{t-2} + \tfrac{2}{9}u_{t-1} + \tfrac{3}{9}u_t + \tfrac{2}{9}u_{t+1} + \tfrac{1}{9}u_{t+2}.$$

The modified Daniell kernel puts half weights at the end points, so with $m = 1$ the weights are $\{h_k\} = \{\frac{1}{4}, \frac{2}{4}, \frac{1}{4}\}$ and

$$\widehat{u}_t = \tfrac{1}{4}u_{t-1} + \tfrac{1}{2}u_t + \tfrac{1}{4}u_{t+1}.$$

Applying the same kernel again to $\widehat{u}_t$ yields

$$\widehat{\widehat{u}}_t = \tfrac{1}{16}u_{t-2} + \tfrac{4}{16}u_{t-1} + \tfrac{6}{16}u_t + \tfrac{4}{16}u_{t+1} + \tfrac{1}{16}u_{t+2}.$$

These coefficients can be obtained in R by issuing the `kernel` command. For example, `kernel("modified.daniell", c(1,1))` would produce the coefficients of the last example.

Example 4.10 Smoothed Periodogram for SOI and Recruitment

In this example, we estimate the spectra of the SOI and Recruitment series using the smoothed periodogram estimate in (4.41). We used a modified Daniell kernel twice, with $m = 3$ both times. This yields $L_h = 1/\sum h_k^2 = 9.232$, which is close to the value of $L = 9$ used in Example 4.8. In this case, the bandwidth is $B = 9.232/480 = .019$ and the modified degrees of freedom is $df = 2L_h 453/480 = 17.43$. The weights, h_k, can be obtained and graphed in R as follows; see Figure 4.12.

```
kernel("modified.daniell", c(3,3))        # for a list
plot(kernel("modified.daniell", c(3,3)))  # for a plot
```

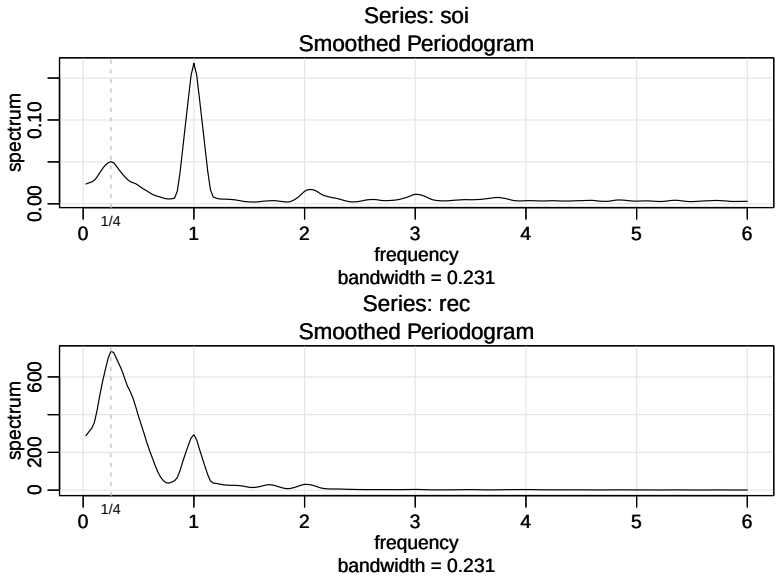


Fig. 4.13. *Smoothed (tapered) spectral estimates of the SOI and Recruitment series; see Example 4.10 for details.*

The resulting spectral estimates can be viewed in Figure 4.13 and we notice that the estimates more appealing than those in Figure 4.9. Figure 4.13 was generated in R as follows:

```
k = kernel("modified.daniell", c(3,3))
soi.smo = mvspec(soi, k,  taper=.1, log="no") # a taper is used
abline(v=c(1/4,1), lty="dotted")
## Repeat above lines with rec replacing soi in line 3
df = soi.smo$df           # df = 17.42618
soi.smo$bandwidth         # Bw = 0.2308103 = 12*9.232/480
```

Reissuing the `mvspec` commands with `log="no"` removed will result in a figure similar to Figure 4.10; see Figure 4.13. Finally, we mention that the modified Daniell kernel is used by default. For example, an easier way to obtain `soi.smo` is to issue the command:

```
soi.smo = mvspec(soi, taper=.1, spans=c(7,7))
```

Notice that `spans` is a vector of odd integers, given in terms of $L = 2m + 1$ instead of m. These values give the widths of the modified Daniell smoother to be used to smooth the periodogram.

Tapering

We are now ready to briefly introduce the concept of *tapering*; a more detailed discussion may be found in Bloomfield (2000, §9.5). Suppose x_t is a mean-zero, stationary process with spectral density $f_x(\omega)$. If we replace the original series by the tapered series

$$y_t = h_t x_t, \tag{4.47}$$

for $t = 1, 2, \ldots, n$, use the modified DFT

$$d_y(\omega_j) = n^{-1/2} \sum_{t=1}^{n} h_t x_t e^{-2\pi i \omega_j t}, \tag{4.48}$$

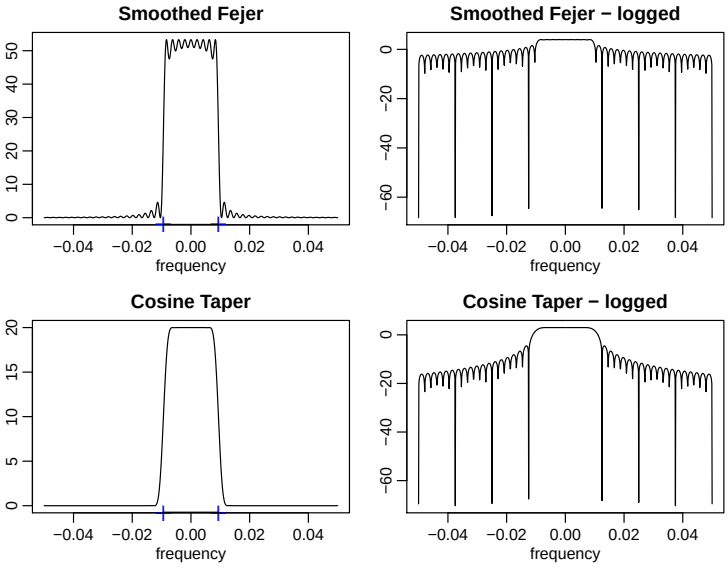


Fig. 4.14. *Averaged Fejér window (top row) and the corresponding cosine taper window (bottom row) for* $L = 9$, $n = 480$. *The extra tic marks on the horizontal axis of the left-hand plots exhibit the predicted bandwidth,* $B_w = 9/480 = .01875$.

and let $I_y(\omega_j) = |d_y(\omega_j)|^2$, we will obtain

$$E[I_y(\omega_j)] = \int_{-1/2}^{1/2} W_n(\omega_j - \omega)\, f_x(\omega)\, d\omega. \tag{4.49}$$

The value $W_n(\omega)$ is called a spectral window because, in view of (4.49), it is determining which part of the spectral density $f_x(\omega)$ is being "seen" by the estimator $I_y(\omega_j)$ on average. In the case that $h_t = 1$ for all t, $I_y(\omega_j) = I_x(\omega_j)$ is simply the periodogram of the data and the window is

$$W_n(\omega) = \frac{\sin^2(n\pi\omega)}{n\sin^2(\pi\omega)} \tag{4.50}$$

with $W_n(0) = n$, which is known as the Fejér or modified Bartlett kernel. If we consider the averaged periodogram in (4.32), namely

$$\bar{f}_x(\omega) = \frac{1}{L}\sum_{k=-m}^{m} I_x(\omega_j + k/n),$$

the window, $W_n(\omega)$, in (4.49) will take the form

$$W_n(\omega) = \frac{1}{nL}\sum_{k=-m}^{m} \frac{\sin^2[n\pi(\omega + k/n)]}{\sin^2[\pi(\omega + k/n)]}\,. \tag{4.51}$$

Tapers generally have a shape that enhances the center of the data relative to the extremities, such as a cosine bell of the form

$$h_t = .5\left[1 + \cos\left(\frac{2\pi(t - \bar{t})}{n}\right)\right], \tag{4.52}$$

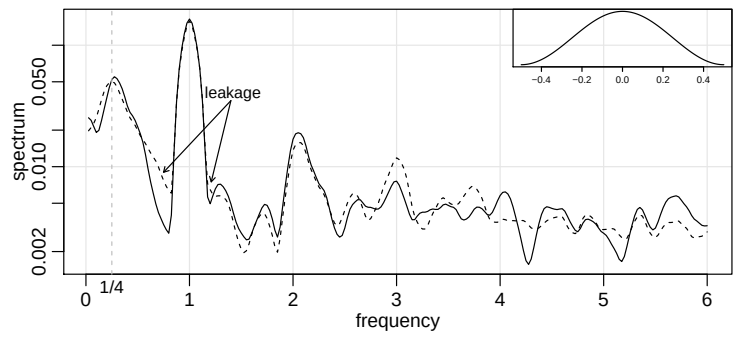


Fig. 4.15. *Smoothed spectral estimates of the SOI without tapering (dashed line) and with full tapering (solid line); see Example 4.11. The insert shows a full cosine bell taper, (4.52), with horizontal axis* $(t-\bar{t})/n$*, for* $t = 1, \ldots, n$*.*

where $\bar{t} = (n+1)/2$, favored by Blackman and Tukey (1959). In Figure 4.14, we have plotted the shapes of two windows, $W_n(\omega)$, for $n = 480$ and $L = 9$, when (i) $h_t \equiv 1$, in which case, (4.51) applies, and (ii) h_t is the cosine taper in (4.52). In both cases the predicted bandwidth should be $B_w = 9/480 = .01875$ cycles per point, which corresponds to the "width" of the windows shown in Figure 4.14. Both windows produce an integrated average spectrum over this band but the untapered window in the top panels shows considerable ripples over the band and outside the band. The ripples outside the band are called sidelobes and tend to introduce frequencies from outside the interval that may contaminate the desired spectral estimate within the band. For example, a large dynamic range for the values in the spectrum introduces spectra in contiguous frequency intervals several orders of magnitude greater than the value in the interval of interest. This effect is sometimes called leakage. Figure 4.14 emphasizes the suppression of the sidelobes in the Fejér kernel when a cosine taper is used.

Example 4.11 The Effect of Tapering the SOI Series

In this example, we examine the effect of tapering on the estimate of the spectrum of the SOI series. The results for the Recruitment series are similar. Figure 4.15 shows two spectral estimates plotted on a log scale. The degree of smoothing here is the same as in Example 4.10. The dashed line in Figure 4.15 shows the estimate without any tapering and hence it is the same as the estimated spectrum displayed in the top of Figure 4.13. The solid line shows the result with full tapering. Notice that the tapered spectrum does a better job in separating the yearly cycle ($\omega = 1$) and the El Niño cycle ($\omega = 1/4$).

The following R session was used to generate Figure 4.15. We note that, by default, `mvspec` does not taper. For full tapering, we use the argument `taper=.5` to instruct `mvspec` to taper 50% of each end of the data; any value between 0 and .5 is acceptable.

```
s0 = mvspec(soi, spans=c(7,7), plot=FALSE)              # no taper
s50 = mvspec(soi, spans=c(7,7), taper=.5, plot=FALSE)  # full taper
plot(s0$freq, s0$spec, log="y", type="l", lty=2, ylab="spectrum",
           xlab="frequency")      # dashed line
lines(s50$freq, s50$spec)         # solid line
```

4.6 Parametric Spectral Estimation

The methods of Section 4.5 lead to estimators generally referred to as nonparametric spectra because no assumption is made about the parametric form of the spectral density. In Property 4.2, we exhibited the spectrum of an ARMA process and we might consider basing a spectral estimator on this function, substituting the parameter estimates from an ARMA(p,q) fit on the data into the formula for the spectral density $f_x(\omega)$ given in (4.16). Such an estimator is called a parametric spectral estimator. For convenience, a parametric spectral estimator is obtained by fitting an AR(p) to the data, where the order p is determined by one of the model selection criteria, such as AIC, AICc, and BIC, defined in (2.15)-(2.17). Parametric autoregressive spectral estimators will often have superior resolution in problems when several closely spaced narrow spectral peaks are present and are preferred by engineers for a broad variety of problems (see Kay, 1988). The development of autoregressive spectral estimators has been summarized by Parzen (1983).

If $\widehat{\phi}_1, \widehat{\phi}_2, \ldots, \widehat{\phi}_p$ and $\widehat{\sigma}_w^2$ are the estimates from an AR(p) fit to x_t, then based on Property 4.2, a parametric spectral estimate of $f_x(\omega)$ is attained by substituting these estimates into (4.16), that is,

$$\widehat{f}_x(\omega) = \frac{\widehat{\sigma}_w^2}{|\widehat{\phi}(\mathrm{e}^{-2\pi i\omega})|^2}, \tag{4.53}$$

where

$$\widehat{\phi}(z) = 1 - \widehat{\phi}_1 z - \widehat{\phi}_2 z^2 - \cdots - \widehat{\phi}_p z^p. \tag{4.54}$$

An interesting fact about rational spectra of the form (4.16) is that any spectral density can be approximated, arbitrarily close, by the spectrum of an AR process.

Property 4.3 AR Spectral Approximation

Let $g(\omega)$ be the spectral density of a stationary process, x_t. Then, given $\epsilon > 0$, there is an AR(p) representation

$$x_t = \sum_{k=1}^{p} \phi_k x_{t-k} + w_t$$

where w_t is white noise with variance σ_w^2, such that

$$|f_x(\omega) - g(\omega)| < \epsilon \quad \text{for all } \omega \in [-1/2, 1/2].$$

Moreover, p is finite and the roots of $\phi(z) = 1 - \sum_{k=1}^{p} \phi_k z^k$ are outside the unit circle.

One drawback, however, is that the property does not tell us how large p must be before the approximation is reasonable; in some situations p may be extremely large. Property 4.3 also holds for MA and for ARMA processes in general, and a proof of the result may be found in Fuller (1996, Ch 4). We demonstrate the technique in the following example.

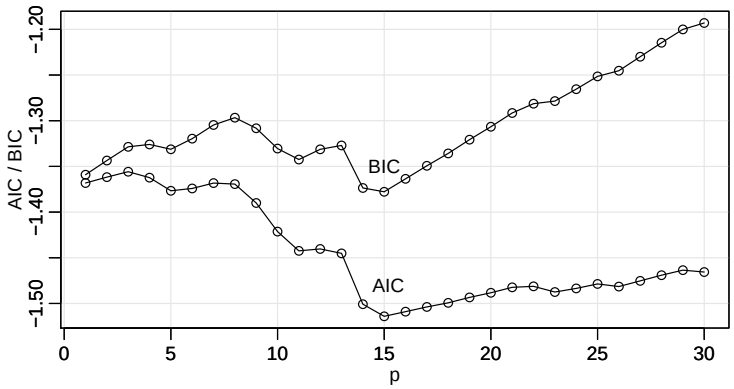


Fig. 4.16. *Model selection criteria AIC and BIC as a function of order p for autoregressive models fitted to the SOI series.*

Example 4.12 Autoregressive Spectral Estimator for SOI

Consider obtaining results comparable to the nonparametric estimators shown in Figure 4.9 for the SOI series. Fitting successively higher order AR(p) models for $p = 1, 2, \ldots, 30$ yields a minimum BIC and a minimum AIC at $p = 15$, as shown in Figure 4.16. We can see from Figure 4.16 that BIC is very definite about which model it chooses; that is, the minimum BIC is very distinct. On the other hand, it is not clear what is going to happen with AIC; that is, the minimum is not so clear, and there is some concern that AIC will start decreasing after $p = 30$. Minimum AICc selects the $p = 15$ model, but suffers from the same uncertainty as AIC. The spectrum is shown in Figure 4.17, and we note the strong peaks near the four year and one year cycles as in the nonparametric estimates obtained in Section 4.5. In addition, the harmonics of the yearly period are evident in the estimated spectrum.

To perform a similar analysis in R, the command `spec.ar` can be used to fit the best model via AIC and plot the resulting spectrum. A quick way to obtain the AIC values is to run the `ar` command as follows.

```
spaic = spec.ar(soi, log="no")    # min AIC spec
abline(v=frequency(soi)*1/52, lty="dotted")  # El Nino Cycle
(soi.ar = ar(soi, order.max=30))  # estimates and AICs
dev.new()
plot(1:30, soi.ar$aic[-1], type="o")   # plot AICs
```

R works only with the AIC in this case. To generate Figure 4.16 we used the following code to obtain AIC and BIC. We added 1 to the BIC to reduce white space in the graphic.

```
n = length(soi)
c() -> AIC -> AICc -> BIC
for (k in 1:30){
 sigma2  = ar(soi, order=k, aic=FALSE)$var.pred
 BIC[k]  = log(sigma2) + (k*log(n)/n)
 AIC[k]  = log(sigma2) + ((n+2*k)/n)
}
IC = cbind(AIC, BIC+1)
ts.plot(IC, type="o", xlab="p", ylab="AIC / BIC")
grid()
```

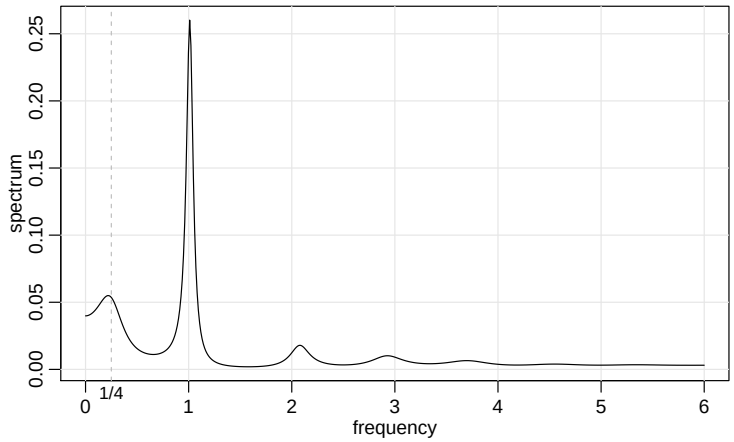


Fig. 4.17. *Autoregressive spectral estimator for the SOI series using the AR(15) model selected by AIC, AICc, and BIC.*

4.7 Linear Filters

Some of the examples of the previous sections have hinted at the possibility the distribution of power or variance in a time series can be modified by making a linear transformation. In this section, we explore that notion further by defining a linear filter and showing how it can be used to extract signals from a time series. The linear filter modifies the spectral characteristics of a time series in a predictable way, and the systematic development of methods for taking advantage of the special properties of linear filters is an important topic in time series analysis.

A linear filter uses a set of specified coefficients a_j, for $j = 0, \pm 1, \pm 2, \ldots$, to transform an input series, x_t, producing an output series, y_t, of the form

$$y_t = \sum_{j=-\infty}^{\infty} a_j x_{t-j}, \qquad \sum_{j=-\infty}^{\infty} |a_j| < \infty. \tag{4.55}$$

The form (4.55) is also called a convolution in some statistical contexts. The coefficients, collectively called the *impulse response function*, are required to satisfy absolute summability so y_t in (4.55) exists as a limit in mean square and the infinite Fourier transform

$$A_{yx}(\omega) = \sum_{j=-\infty}^{\infty} a_j \, \mathrm{e}^{-2\pi i \omega j}, \tag{4.56}$$

called the *frequency response function*, is well defined. We have already encountered several linear filters, for example, the simple three-point moving average in Example 1.7, which can be put into the form of (4.55) by letting $a_{-1} = a_0 = a_1 = 1/3$ and taking $a_j = 0$ for $|j| \geq 2$.

The importance of the linear filter stems from its ability to enhance certain parts of the spectrum of the input series. We now state the following result.

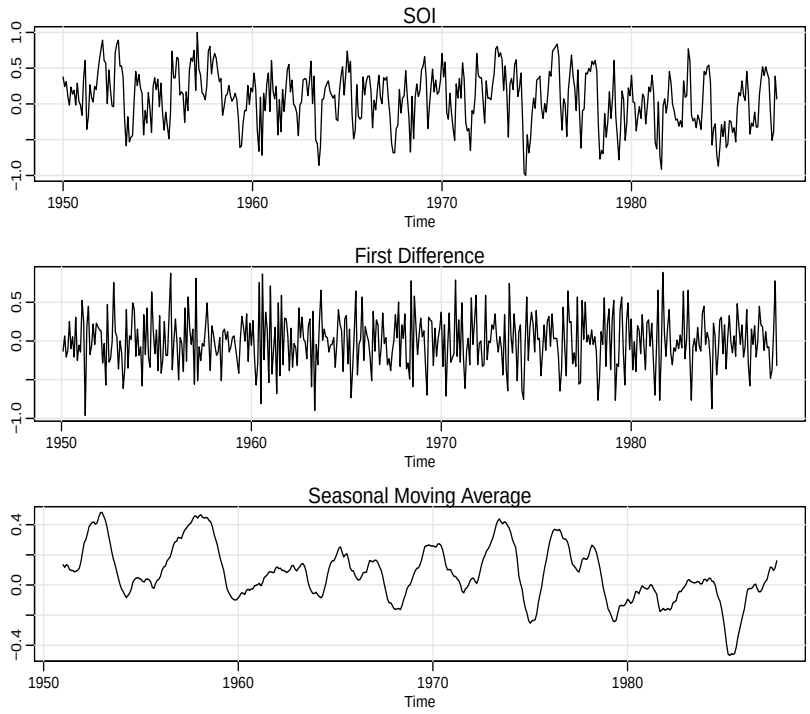


Fig. 4.18. *SOI series (top) compared with the differenced SOI (middle) and a centered 12-month moving average (bottom).*

Property 4.4 Output Spectrum of a Filtered Stationary Series

Assuming existence of spectra, the spectrum of the filtered output y_t in (4.55) is related to the spectrum of the input x_t by

$$f_{yy}(\omega) = |A_{yx}(\omega)|^2 \, f_{xx}(\omega), \tag{4.57}$$

where the frequency response function $A_{yx}(\omega)$ is defined in (4.56).

The result (4.57) enables us to calculate the exact effect on the spectrum of any given filtering operation. This important property shows the spectrum of the input series is changed by filtering and the effect of the change can be characterized as a frequency-by-frequency multiplication by the squared magnitude of the frequency response function. Again, an obvious analogy to a property of the variance in classical statistics holds, namely, if x is a random variable with variance σ_x^2, then $y = ax$ will have variance $\sigma_y^2 = a^2\sigma_x^2$, so the variance of the linearly transformed random variable is changed by multiplication by a^2 in much the same way as the linearly filtered spectrum is changed in (4.57).

Finally, we mention that Property 4.2, which was used to get the spectrum of an ARMA process, is just a special case of Property 4.4 where in (4.55), $x_t = w_t$ is white noise, in which case $f_{xx}(\omega) = \sigma_w^2$, and $a_j = \psi_j$, in which case

$$A_{yx}(\omega) = \psi(\mathrm{e}^{-2\pi i\omega}) = \theta(\mathrm{e}^{-2\pi i\omega}) \,/\, \phi(\mathrm{e}^{-2\pi i\omega}).$$

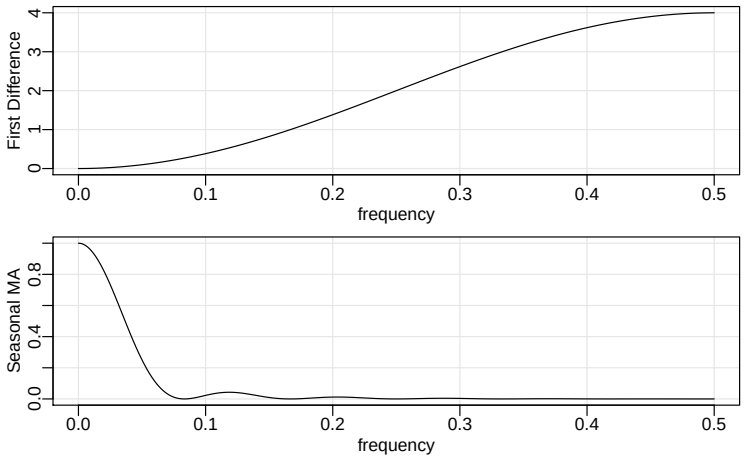


Fig. 4.19. *Squared frequency response functions of the first difference (top) and twelve-month moving average (bottom) filters.*

Example 4.13 First Difference and Moving Average Filters

We illustrate the effect of filtering with two common examples, the first difference filter

$$y_t = \nabla x_t = x_t - x_{t-1}$$

and the symmetric moving average filter

$$y_t = \tfrac{1}{24}\left(x_{t-6} + x_{t+6}\right) + \tfrac{1}{12}\sum_{r=-5}^{5} x_{t-r},$$

which is a modified Daniell kernel with $m = 6$. The results of filtering the SOI series using the two filters are shown in the middle and bottom panels of Figure 4.18. Notice that the effect of differencing is to roughen the series because it tends to retain the higher or faster frequencies. The centered moving average smoothes the series because it retains the lower frequencies and tends to attenuate the higher frequencies. In general, differencing is an example of a *high-pass filter* because it retains or passes the higher frequencies, whereas the moving average is a *low-pass filter* because it passes the lower or slower frequencies.

Notice that the slower periods are enhanced in the symmetric moving average and the seasonal or yearly frequencies are attenuated. The filtered series makes about 9 cycles in the length of the data (about one cycle every 52 months) and the moving average filter tends to enhance or extract the signal that is associated with El Niño. Moreover, by the low-pass filtering of the data, we get a better sense of the El Niño effect and its irregularity.

Now, having done the filtering, it is essential to determine the exact way in which the filters change the input spectrum. We shall use (4.56) and (4.57) for this purpose. The first difference filter can be written in the form (4.55) by letting $a_0 = 1, a_1 = -1$, and $a_r = 0$ otherwise. This implies that

$$A_{yx}(\omega) = 1 - \mathrm{e}^{-2\pi i\omega},$$

and the squared frequency response becomes

$$|A_{yx}(\omega)|^2 = (1 - e^{-2\pi i\omega})(1 - e^{2\pi i\omega}) = 2[1 - \cos(2\pi\omega)]. \quad (4.58)$$

The top panel of Figure 4.19 shows that the first difference filter will attenuate the lower frequencies and enhance the higher frequencies because the multiplier of the spectrum, $|A_{yx}(\omega)|^2$, is large for the higher frequencies and small for the lower frequencies. Generally, the slow rise of this kind of filter does not particularly recommend it as a procedure for retaining only the high frequencies.

For the centered 12-month moving average, we can take $a_{-6} = a_6 = 1/24$, $a_k = 1/12$ for $-5 \le k \le 5$ and $a_k = 0$ elsewhere. Substituting and recognizing the cosine terms gives

$$A_{yx}(\omega) = \tfrac{1}{12}\Big[1 + \cos(12\pi\omega) + 2\sum_{k=1}^{5}\cos(2\pi\omega k)\Big]. \quad (4.59)$$

Plotting the squared frequency response of this function as in Figure 4.19 shows that we can expect this filter to cut most of the frequency content above .05 cycles per point. This corresponds to eliminating periods shorter than $1/.05 = 20$ time points. In particular, this drives down the yearly components with periods of 12 months and enhances the El Niño frequency, which is somewhat lower. The filter is not completely efficient at attenuating high frequencies; some power contributions are left at higher frequencies, as shown in the function $|A_{yx}(\omega)|^2$ and in the spectrum of the moving average shown in Figure 4.4.

The following R session shows how to filter the data, perform the spectral analysis of a filtered series, and plot the squared frequency response curves of the difference and moving average filters.

```
par(mfrow=c(3,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
tsplot(soi)                          # plot data
tsplot(diff(soi))                    # plot first difference
k = kernel("modified.daniell", 6)    # filter weights
tsplot(soif <- kernapply(soi, k))    # plot 12 month filter
dev.new()
spectrum(soif, spans=9, log="no") # spectral analysis (not shown)
abline(v=12/52, lty="dashed")
dev.new()
##-- frequency responses --##
par(mfrow=c(2,1), mar=c(3,3,1,1), mgp=c(1.6,.6,0))
w = seq(0, .5, by=.01)
FRdiff = abs(1-exp(2i*pi*w))^2
plot(w, FRdiff, type='l', xlab='frequency', panel.first=grid())
u = cos(2*pi*w)+cos(4*pi*w)+cos(6*pi*w)+cos(8*pi*w)+cos(10*pi*w)
FRma = ((1 + cos(12*pi*w) + 2*u)/12)^2
plot(w, FRma, type='l', xlab='frequency', panel.first=grid())
```

4.8 Multiple Series and Cross-Spectra

The notion of analyzing frequency fluctuations using classical statistical ideas extends to the case in which there are several jointly stationary series, for example, x_t and y_t. In this case, we can introduce the idea of a correlation indexed by frequency, called the coherence. The autocovariance function

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

has a spectral representation given by

$$\gamma_{xy}(h) = \int_{-1/2}^{1/2} f_{xy}(\omega) e^{2\pi i \omega h}\, d\omega \quad h = 0, \pm 1, \pm 2, ..., \tag{4.60}$$

where the cross-spectrum is defined as the Fourier transform

$$f_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h)\, e^{-2\pi i \omega h} \quad -1/2 \le \omega \le 1/2, \tag{4.61}$$

assuming that the cross-covariance function is absolutely summable, as was the case for the autocovariance. The cross-spectrum is generally a complex-valued function, and it is often written as[5]

$$f_{xy}(\omega) = c_{xy}(\omega) - i q_{xy}(\omega), \tag{4.62}$$

where

$$c_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h)\, \cos(2\pi\omega h) \tag{4.63}$$

and

$$q_{xy}(\omega) = \sum_{h=-\infty}^{\infty} \gamma_{xy}(h)\, \sin(2\pi\omega h) \tag{4.64}$$

are defined as the cospectrum and quadspectrum, respectively. Because of the relationship $\gamma_{yx}(h) = \gamma_{xy}(-h)$, it follows, by substituting into (4.61) and rearranging, that

$$f_{yx}(\omega) = \overline{f_{xy}(\omega)}. \tag{4.65}$$

This result, in turn, implies that the cospectrum and quadspectrum satisfy

$$c_{yx}(\omega) = c_{xy}(\omega) \tag{4.66}$$

and

$$q_{yx}(\omega) = -q_{xy}(\omega). \tag{4.67}$$

An important example of the application of the cross-spectrum is to the problem of predicting an output series y_t from some input series x_t through a linear filter relation such as the three-point moving average considered below. A measure of the strength of such a relation is the squared coherence function, defined as

Prediction of one time series forme another

$$\rho^2_{y\cdot x}(\omega) = \frac{|f_{yx}(\omega)|^2}{f_{xx}(\omega) f_{yy}(\omega)}, \tag{4.68}$$

Square Coherence

where $f_{xx}(\omega)$ and $f_{yy}(\omega)$ are the individual spectra of the x_t and y_t series, respectively. Although we consider a more general form of this that applies to multiple inputs later, it is instructive to display the single input case as (4.68) to emphasize the analogy with conventional squared correlation, which takes the form

[5] For this section, it will be useful to recall the facts $e^{-i\alpha} = \cos(\alpha) - i\sin(\alpha)$ and if $z = a + ib$, then $\bar{z} = a - ib$.

$$\rho_{yx}^2 = \frac{\sigma_{yx}^2}{\sigma_x^2 \sigma_y^2},$$

for random variables with variances σ_x^2 and σ_y^2 and covariance $\sigma_{yx} = \sigma_{xy}$. This motivates the interpretation of squared coherence and the squared correlation between two time series at frequency ω.

Example 4.14 Three-Point Moving Average

As a simple example, we compute the cross-spectrum between x_t and the three-point moving average $y_t = (x_{t-1} + x_t + x_{t+1})/3$, where x_t is a stationary input process with spectral density $f_{xx}(\omega)$. First,

$$\begin{aligned}
\gamma_{xy}(h) &= \operatorname{cov}(x_{t+h}, y_t) = \tfrac{1}{3}\operatorname{cov}(x_{t+h}, x_{t-1} + x_t + x_{t+1}) \\
&= \frac{1}{3}\big(\gamma_{xx}(h+1) + \gamma_{xx}(h) + \gamma_{xx}(h-1)\big) \\
&= \frac{1}{3}\int_{-1/2}^{1/2} \left(e^{2\pi i\omega} + 1 + e^{-2\pi i\omega}\right) e^{2\pi i\omega h} f_{xx}(\omega)\, d\omega \\
&= \frac{1}{3}\int_{-1/2}^{1/2} [1 + 2\cos(2\pi\omega)] f_{xx}(\omega) e^{2\pi i\omega h}\, d\omega,
\end{aligned}$$

where we have use (4.14). Using the uniqueness of the Fourier transform, we argue from the spectral representation (4.60) that

$$f_{xy}(\omega) = \tfrac{1}{3}\left[1 + 2\cos(2\pi\omega)\right] f_{xx}(\omega)$$

so that the cross-spectrum is real in this case. From Example 4.5, the spectral density of y_t is

$$\begin{aligned}
f_{yy}(\omega) &= \tfrac{1}{9}[3 + 4\cos(2\pi\omega) + 2\cos(4\pi\omega)] f_{xx}(\omega) \\
&= \tfrac{1}{9}\left[1 + 2\cos(2\pi\omega)\right]^2 f_{xx}(\omega),
\end{aligned}$$

using the identity $\cos(2\alpha) = 2\cos^2(\alpha) - 1$ in the last step. Substituting into (4.68) yields the squared coherence between x_t and y_t as unity over all frequencies. This is a characteristic inherited by more general linear filters. However, if some noise is added to the three-point moving average, the coherence is not unity; these kinds of models will be considered in detail later.

Property 4.5 Spectral Representation of a Vector Process

If the elements of the $p \times p$ autocovariance function matrix

$$\Gamma(h) = E[(x_{t+h} - \mu)(x_t - \mu)']$$

of a p-dimensional stationary time series, $x_t = (x_{t1}, x_{t2}, \ldots, x_{tp})'$, has elements satisfying

$$\sum_{h=-\infty}^{\infty} |\gamma_{jk}(h)| < \infty \tag{4.69}$$

for all $j, k = 1, \ldots, p$, then $\Gamma(h)$ has the representation

$$\Gamma(h) = \int_{-1/2}^{1/2} e^{2\pi i\omega h} f(\omega)\, d\omega \quad h = 0, \pm 1, \pm 2, \ldots, \tag{4.70}$$

as the inverse transform of the spectral density matrix, $f(\omega) = \{f_{jk}(\omega)\}$, for $j, k = 1, \ldots, p$, with elements equal to the cross-spectral components. The matrix $f(\omega)$ has the representation

$$f(\omega) = \sum_{h=-\infty}^{\infty} \Gamma(h) \mathrm{e}^{-2\pi i \omega h} \quad -1/2 \le \omega \le 1/2. \tag{4.71}$$

Example 4.15 Spectral Matrix of a Bivariate Process

Consider a jointly stationary bivariate process (x_t, y_t). We arrange the autocovariances in the matrix

$$\Gamma(h) = \begin{pmatrix} \gamma_{xx}(h) & \gamma_{xy}(h) \\ \gamma_{yx}(h) & \gamma_{yy}(h) \end{pmatrix}.$$

The spectral matrix would be given by

$$f(\omega) = \begin{pmatrix} f_{xx}(\omega) & f_{xy}(\omega) \\ f_{yx}(\omega) & f_{yy}(\omega) \end{pmatrix},$$

where the Fourier transform (4.70) and (4.71) relate the autocovariance and spectral matrices.

The extension of spectral estimation to vector series is fairly obvious. For the vector series $x_t = (x_{t1}, x_{t2}, \ldots, x_{tp})'$, we may use the vector of DFTs, say $d(\omega_j) = (d_1(\omega_j), d_2(\omega_j), \ldots, d_p(\omega_j))'$, and estimate the spectral matrix by

$$\bar{f}(\omega) = L^{-1} \sum_{k=-m}^{m} I(\omega_j + k/n) \tag{4.72}$$

where now

$$I(\omega_j) = d(\omega_j)\, d^*(\omega_j) \tag{4.73}$$

is a $p \times p$ complex matrix.[6]

Again, the series may be tapered before the DFT is taken in (4.72) and we can use weighted estimation,

$$\widehat{f}(\omega) = \sum_{k=-m}^{m} h_k\, I(\omega_j + k/n) \tag{4.74}$$

where $\{h_k\}$ are weights as defined in (4.41). The estimate of squared coherence between two series, y_t and x_t is

$$\widehat{\rho}^2_{y\cdot x}(\omega) = \frac{|\widehat{f}_{yx}(\omega)|^2}{\widehat{f}_{xx}(\omega)\widehat{f}_{yy}(\omega)}. \tag{4.75}$$

If the spectral estimates in (4.75) are obtained using equal weights, we will write $\bar{\rho}^2_{y\cdot x}(\omega)$ for the estimate.

Under general conditions, if $\rho^2_{y\cdot x}(\omega) > 0$ then

[6] If Z is a complex matrix, then $Z^* = \overline{Z}'$ denotes the conjugate transpose operation. That is, Z^* is the result of replacing each element of Z by its complex conjugate and transposing the resulting matrix.

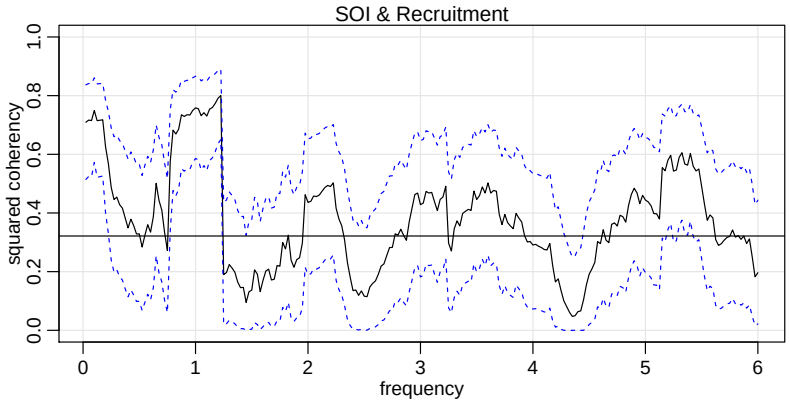


Fig. 4.20. *Squared coherency between the SOI and Recruitment series;* $L = 19$, $n = 453$, $n' = 480$, *and* $\alpha = .001$. *The horizontal line is* $C_{.001}$.

$$|\widehat{\rho}_{y\cdot x}(\omega)| \sim AN\left(|\rho_{y\cdot x}(\omega)|, \left(1-\rho^2_{y\cdot x}(\omega)\right)^2/2L_h\right) \tag{4.76}$$

where L_h is defined in (4.43); the details of this result may be found in Brockwell and Davis (1991, Ch 11). We may use (4.76) to obtain approximate confidence intervals for the squared coherency $\rho^2_{y\cdot x}(\omega)$.

We can test the hypothesis that $\rho^2_{y\cdot x}(\omega) = 0$ if we use $\bar{\rho}^2_{y\cdot x}(\omega)$ for the estimate with $L > 1$,[7] that is,

$$\bar{\rho}^2_{y\cdot x}(\omega) = \frac{|\bar{f}_{yx}(\omega)|^2}{\bar{f}_{xx}(\omega)\bar{f}_{yy}(\omega)}. \tag{4.77}$$

In this case, under the null hypothesis, the statistic

$$F = \frac{\bar{\rho}^2_{y\cdot x}(\omega)}{(1-\bar{\rho}^2_{y\cdot x}(\omega))}(L-1) \tag{4.78}$$

has an approximate F-distribution with 2 and $2L-2$ degrees of freedom. When the series have been extended to length n', we replace $2L-2$ by $df-2$, where df is defined in (4.38). Solving (4.78) for a particular significance level α leads to

$$C_\alpha = \frac{F_{2,2L-2}(\alpha)}{L-1+F_{2,2L-2}(\alpha)} \tag{4.79}$$

as the approximate value that must be exceeded for the original squared coherence to be able to reject $\rho^2_{y\cdot x}(\omega) = 0$ at an a priori specified frequency.

Example 4.16 Coherence Between SOI and Recruitment

Figure 4.20 shows the squared coherence between the SOI and Recruitment series over a wider band than was used for the spectrum. In this case, we used $L = 19$, $df = 2(19)(453/480) \approx 36$ and $F_{2,df-2}(.001) \approx 8.53$ at the significance level $\alpha = .001$. Hence, we may reject the hypothesis of no coherence for values of $\bar{\rho}^2_{y\cdot x}(\omega)$ that exceed $C_{.001} = .32$. We emphasize that this method is crude because, in addition to the fact that the F-statistic is

[7] If $L = 1$ then $\bar{\rho}^2_{y\cdot x}(\omega) \equiv 1$.

approximate, we are examining the squared coherence across all frequencies with the Bonferroni inequality in mind. Figure 4.20 also exhibits confidence bands as part of the R plotting routine. We emphasize that these bands are only valid for ω where $\rho^2_{y\cdot x}(\omega) > 0$.

In this case, the seasonal frequency and the El Niño frequencies ranging between about 3 and 7 year periods are strongly coherent. Other frequencies are also strongly coherent, although the strong coherence is less impressive because the underlying power spectrum at these higher frequencies is fairly small. Finally, we note that the coherence is persistent at the seasonal harmonic frequencies.

This example may be reproduced using the following R commands.

```
sr = mvspec(cbind(soi,rec), kernel('daniell',9), plot=FALSE)
(sr$df)
 [1] 35.8625
(f = qf(.999, 2, sr$df-2) )
 [1] 8.529792
(C = f/(18+f) )
 [1] 0.3215175
plot(sr, plot.type = "coh", ci.lty = 2)
abline(h = C)
```

Problems

4.1 Repeat the simulations and analyses in Example 4.1 and Example 4.2 with the following changes:

(a) Change the sample size to $n = 128$ and generate and plot the same series as in Example 4.1:

$$\begin{aligned} x_{t1} &= 2\cos(2\pi\,.06\,t) + 3\sin(2\pi\,.06\,t), \\ x_{t2} &= 4\cos(2\pi\,.10\,t) + 5\sin(2\pi\,.10\,t), \\ x_{t3} &= 6\cos(2\pi\,.40\,t) + 7\sin(2\pi\,.40\,t), \\ x_t &= x_{t1} + x_{t2} + x_{t3}. \end{aligned}$$

What is the major difference between these series and the series generated in Example 4.1? (Hint: The answer is *fundamental*. But if your answer is the series are longer, you may be punished severely.)

(b) As in Example 4.2, compute and plot the periodogram of the series, x_t, generated in (a) and comment.

(c) Repeat the analyses of (a) and (b) but with $n = 100$ (as in Example 4.1), and adding noise to x_t; that is

$$x_t = x_{t1} + x_{t2} + x_{t3} + w_t$$

where $w_t \sim$ iid $\mathrm{N}(0, \sigma_w = 5)$. That is, you should simulate and plot the data, and then plot the periodogram of x_t and comment.

4.2 Verify (4.5).

4.3 Consider an MA(1) process

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

where θ is a parameter.

(a) Derive a formula for the power spectrum of x_t, expressed in terms of θ and ω.
(b) Use `arma.spec()` to plot the spectral density of x_t for $\theta > 0$ and for $\theta < 0$ (just select arbitrary values).
(c) How should we interpret the spectra exhibited in part (b)?

4.4 Consider a first-order autoregressive model

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

where ϕ, for $|\phi| < 1$, is a parameter and the w_t are independent random variables with mean zero and variance σ_w^2.

(a) Show that the power spectrum of x_t is given by

$$f_x(\omega) = \frac{\sigma_w^2}{1 + \phi^2 - 2\phi\cos(2\pi\omega)}.$$

(b) Verify the autocovariance function of this process is

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

$h = 0, \pm 1, \pm 2, \ldots$, by showing that the inverse transform of $\gamma_x(h)$ is the spectrum derived in part (a).

4.5 In applications, we will often observe series containing a signal that has been delayed by some unknown time D, i.e.,

$$x_t = s_t + As_{t-D} + n_t,$$

where s_t and n_t are stationary and independent with zero means and spectral densities $f_s(\omega)$ and $f_n(\omega)$, respectively. The delayed signal is multiplied by some unknown constant A. Find the autocovariance function of x_t and use it to show

$$f_x(\omega) = [1 + A^2 + 2A\cos(2\pi\omega D)]f_s(\omega) + f_n(\omega).$$

4.6 Figure 4.21 shows the biyearly smoothed (12-month moving average) number of sunspots from June 1749 to December 1978 with $n = 459$ points that were taken twice per year; the data are contained in `sunspotz`. With Example 4.7 as a guide, perform a periodogram analysis identifying the predominant periods and obtaining confidence intervals for the identified periods. Interpret your findings.

4.7 The levels of salt concentration known to have occurred over rows, corresponding to the average temperature levels for the soil science are in `salt` and `saltemp`. Plot the series and then identify the dominant frequencies by performing separate spectral analyses on the two series. Include confidence intervals for the dominant frequencies and interpret your findings.

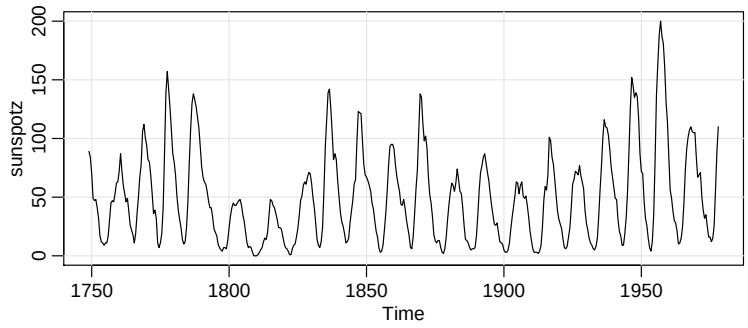


Fig. 4.21. *Smoothed 12-month sunspot numbers (`sunspotz`) sampled twice per year.*

4.8 Let the observed series x_t be composed of a periodic signal and noise so it can be written as

$$x_t = \beta_1 \cos(2\pi\omega_k t) + \beta_2 \sin(2\pi\omega_k t) + w_t,$$

where w_t is a white noise process with variance σ_w^2. The frequency $\omega_k \neq 0, \frac{1}{2}$ is assumed to be known and of the form k/n. Given data $x_1, \ldots, x_n$, suppose we consider estimating β_1, β_2 and σ_w^2 by least squares.

(a) Use simple regression formulas to show that for a fixed ω_k, the least squares regression coefficients are

$$\widehat{\beta}_1 = 2n^{-1/2} d_c(\omega_k) \quad \text{and} \quad \widehat{\beta}_2 = 2n^{-1/2} d_s(\omega_k),$$

where the cosine and sine transforms (4.22) and (4.23) appear on the right-hand side. *Hint:* See Problem 4.22.

(b) Prove that the error sum of squares can be written as

$$SSE = \sum_{t=1}^{n} x_t^2 - 2I_x(\omega_k)$$

so that the value of ω_k that minimizes squared error is the same as the value that maximizes the periodogram $I_x(\omega_k)$ estimator (4.20).

(c) Show that the sum of squares for the regression is given by

$$SSR = 2I_x(\omega_k).$$

(d) Under the Gaussian assumption and fixed ω_k, show that the F-test of no regression leads to an F-statistic that is a monotone function of $I_x(\omega_k)$.

4.9 Analyze the chicken price data (`chicken`) using a nonparametric spectral estimation procedure. Aside from the obvious annual cycle discovered in Example 2.5, what other interesting cycles are revealed?

4.10 Repeat Problem 4.6 using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

4.11 Repeat Problem 4.7 using a nonparametric spectral estimation procedure. In addition to discussing your findings in detail, comment on your choice of a spectral estimate with regard to smoothing and tapering.

4.12 Often, the periodicities in the sunspot series are investigated by fitting an autoregressive spectrum of sufficiently high order. The main periodicity is often stated to be in the neighborhood of 11 years. Fit an autoregressive spectral estimator to the sunspot data using a model selection method of your choice. Compare the result with a conventional nonparametric spectral estimator found in Problem 4.6.

4.13 Analyze the chicken price data (`chicken`) using a parametric spectral estimation procedure. Compare the results to Problem 4.9.

4.14 Fit an autoregressive spectral estimator to the Recruitment series and compare it to the results of Example 4.10.

4.15 The periodic behavior of a time series induced by echoes can also be observed in the spectrum of the series; this fact can be seen from the results stated in Problem 4.5(a). Using the notation of that problem, suppose we observe $x_t = s_t + As_{t-D} + n_t$, which implies the spectra satisfy $f_x(\omega) = [1 + A^2 + 2A\cos(2\pi\omega D)]f_s(\omega) + f_n(\omega)$. If the noise is negligible ($f_n(\omega) \approx 0$) then $\log f_x(\omega)$ is approximately the sum of a periodic component, $\log[1 + A^2 + 2A\cos(2\pi\omega D)]$, and $\log f_s(\omega)$. Bogart et al. (1962) proposed treating the detrended log spectrum as a pseudo time series and calculating its spectrum, or *cepstrum*, which should show a peak at a *quefrency* corresponding to $1/D$. The cepstrum can be plotted as a function of quefrency, from which the delaty D can be estimated.

For the speech series presented in `speech`, estimate the pitch period using cepstral analysis as follows.

(a) Calculate and display the log-periodogram of the data. Is the periodogram periodic, as predicted?

(b) Perform a cepstral (spectral) analysis on the detrended logged periodogram, and use the results to estimate the delay D.

4.16 Consider two time series

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

$$y_t = \frac{1}{2}(w_t + w_{t-1}),$$

formed from the white noise series w_t with variance $\sigma_w^2 = 1$.

(a) Are x_t and y_t jointly stationary? Recall the cross-covariance function must also be a function only of the lag h and cannot depend on time.

(b) Compute the spectra $f_y(\omega)$ and $f_x(\omega)$, and comment on the difference between the two results.

(c) Suppose sample spectral estimators $\bar{f}_y(.10)$ are computed for the series using $L = 3$. Find a and b such that

$$P\left\{a \le \bar{f}_y(.10) \le b\right\} = .90.$$

This expression gives two points that will contain 90% of the sample spectral values. Put 5% of the area in each tail.

4.17 Analyze the coherency between the temperature and salt data discussed in Problem 4.7. Discuss your findings.

4.18 Consider two processes

$$x_t = w_t \quad \text{and} \quad y_t = \phi x_{t-D} + v_t$$

where w_t and v_t are independent white noise processes with common variance σ^2, ϕ is a constant, and D is a fixed integer delay.

(a) Compute the coherency between x_t and y_t.
(b) Simulate $n = 1024$ normal observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 0$. Then estimate and plot the coherency between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

4.19 For the processes in Problem 4.18:

(a) Compute the phase between x_t and y_t.
(b) Simulate $n = 1024$ observations from x_t and y_t for $\phi = .9$, $\sigma^2 = 1$, and $D = 1$. Then estimate and plot the phase between the simulated series for the following values of L and comment:
(i) $L = 1$, (ii) $L = 3$, (iii) $L = 41$, and (iv) $L = 101$.

4.20 Consider the bivariate time series records containing monthly U.S. production as measured by the Federal Reserve Board Production Index (`prodn`) and monthly unemployment (`unemp`) that are included with `astsa`.

(a) Compute the spectrum and the log spectrum for each series, and identify statistically significant peaks. Explain what might be generating the peaks. Compute the coherence, and explain what is meant when a high coherence is observed at a particular frequency.
(b) What would be the effect of applying the filter

$$u_t = x_t - x_{t-1} \quad \text{followed by} \quad v_t = u_t - u_{t-12}$$

to the series given above? Plot the predicted frequency responses of the simple difference filter and of the seasonal difference of the first difference.
(c) Apply the filters successively to one of the two series and plot the output. Examine the output after taking a first difference and comment on whether stationarity is a reasonable assumption. Why or why not? Plot after taking the seasonal difference of the first difference. What can be noticed about the output that is consistent with what you have predicted from the frequency response? Verify by computing the spectrum of the output after filtering.

4.21 Let $x_t = \cos(2\pi\omega t)$, and consider the output $y_t = \sum_{k=-\infty}^{\infty} a_k x_{t-k}$, where $\sum_k |a_k| < \infty$. Show $y_t = |A(\omega)| \cos(2\pi\omega t + \phi(\omega))$, where $|A(\omega)|$ and $\phi(\omega)$ are the amplitude and phase of the filter, respectively. Interpret the result in terms of the relationship between the input series, x_t, and the output series, y_t.

4.22 * This is here for useful information. Verify that for any positive integer n and $j,k = 0,1,\ldots,[\![n/2]\!]$, where $[\![\cdot]\!]$ denotes the greatest integer function:

(a) Except for $j = 0$ or $j = n/2$,

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

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

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

(c) For $j \neq k$,

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

(d) Also, for any j and k,

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

* Note, $\sum_{t=1}^{n} z^t = z\frac{1-z^n}{1-z}$ for $z \neq 1$, and we'll do (a):

$$\begin{aligned}\textstyle\sum_{t=1}^{n} \cos^2(2\pi t\, j/n) &= \tfrac{1}{4}\textstyle\sum_{t=1}^{n} \left(e^{2\pi i t\, j/n} + e^{-2\pi i t\, j/n}\right)\left(e^{2\pi i t\, j/n} + e^{-2\pi i t\, j/n}\right) \\ &= \tfrac{1}{4}\textstyle\sum_{t=1}^{n} \left(e^{4\pi i t\, j/n} + 1 + 1 + e^{-4\pi i t\, j/n}\right) = \tfrac{n}{2}.\end{aligned}$$

Chapter 5

Some Additional Topics **

In this chapter, we present special or advanced topics in the time domain. This chapter consists of sections of independent topics that may be read in any order. The sections depend on a basic knowledge of ARMA models, forecasting and estimation, which is the material covered in Chapter 3.

5.1 GARCH Models

Various problems such as option pricing in finance have motivated the study of the volatility, or variability, of a time series. ARMA models were used to model the conditional mean of a process when the conditional variance was constant. For example, in the AR(1) model $x_t = \phi_0 + \phi_1 x_{t-1} + w_t$ we have

$$\mu_t = E(x_t \mid x_{t-1}, x_{t-2}, \dots) = \phi_0 + \phi_1 x_{t-1}$$
$$\sigma_t^2 = \text{var}(x_t \mid x_{t-1}, x_{t-2}, \dots) = \text{var}(w_t) = \sigma_w^2 .$$

In many problems, however, the assumption of a constant conditional variance will be violated. Models such as the autoregressive conditionally heteroscedastic or ARCH model, first introduced by Engle (1982), were developed to model changes in volatility. These models were later extended to generalized ARCH, or GARCH models by Bollerslev (1986).

In these problems, we are concerned with modeling the return or growth rate of a series. For example, if x_t is the value of an asset at time t, then the return or relative gain, r_t, of the asset at time t is

$$r_t = \frac{x_t - x_{t-1}}{x_{t-1}}. \tag{5.1}$$

Definition (5.1) implies that $x_t = (1 + r_t)x_{t-1}$. Thus, based on the discussion in Example 1.3 and Section 3.7, if the return represents a small (in magnitude) percentage change then

** This chapter requires skills beyond high school level mathematics and statistics.

$$\nabla \log(x_t) \approx r_t. \tag{5.2}$$

Either value, $\nabla \log(x_t)$ or $(x_t - x_{t-1})/x_{t-1}$, will be called the *return*,[1] and will be denoted by r_t. To see the difference in the growth of US GNP, you can use the following code (not displayed).

```
tsplot(diff(log(gnp)), type='o')          # using diff log
points(diff(gnp)/lag(gnp,-1), pch='+')    # actual quarterly growth
```

Typically, for financial series, the return r_t, does not have a constant conditional variance, and highly volatile periods tend to be clustered together. There appears to be a strong dependence of sudden bursts of variability in a return on the series own past. For example, Figure 1.4 shows the daily returns of the Dow Jones Industrial Average (DJIA) from April 20, 2006 to April 20, 2016. In this case, as is typical, the return r_t is fairly stable except for short-term bursts of high volatility.

The simplest ARCH model, the ARCH(1), models the return as

$$r_t = \sigma_t \epsilon_t \tag{5.3}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2, \tag{5.4}$$

where ϵ_t is standard Gaussian white noise, $\epsilon_t \sim$ iid $\mathrm{N}(0,1)$. The normal assumption may be relaxed; we will discuss this later. As with ARMA models, we must impose some constraints on the model parameters to obtain desirable properties. An obvious constraint is that $\alpha_0, \alpha_1 \geq 0$ because σ_t^2 is a variance.

It is possible to write the ARCH(1) model as a non-Gaussian AR(1) model in the square of the returns r_t^2. First, rewrite (5.3)–(5.4) as

$$\begin{aligned} r_t^2 &= \sigma_t^2 \epsilon_t^2 \\ \alpha_0 + \alpha_1 r_{t-1}^2 &= \sigma_t^2, \end{aligned}$$

and subtract the two equations to obtain

$$r_t^2 - (\alpha_0 + \alpha_1 r_{t-1}^2) = \sigma_t^2 \epsilon_t^2 - \sigma_t^2.$$

Now, write this equation as

$$r_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + v_t, \tag{5.5}$$

where $v_t = \sigma_t^2(\epsilon_t^2 - 1)$. Because ϵ_t^2 is the square of a $\mathrm{N}(0,1)$ random variable, $\epsilon_t^2 - 1$ is a shifted (to have mean-zero), χ_1^2 random variable.

Note that the basic model implies that the conditional distribution of r_t given r_{t-1} is Gaussian:

$$r_t \mid r_{t-1} \sim \mathrm{N}(0, \alpha_0 + \alpha_1 r_{t-1}^2). \tag{5.6}$$

In addition, unconditionally, r_t has a zero mean because[2]

$$E(r_t) = EE(r_t \mid r_{t-1}) = E0 = 0. \tag{5.7}$$

[1] Recall that if $r_t = (x_t - x_{t-1})/x_{t-1}$ is a small percentage, then $\log(1 + r_t) \approx r_t$. It is easier to program $\nabla \log x_t$, so this is often used instead of calculating r_t directly. Although it is a misnomer, $\nabla \log x_t$ is often called the *log-return*; but the returns are NOT being logged.

[2] We are using the Law of Iterated Expectations

We can also show that the returns are uncorrelated. For example,

$$\begin{aligned}\gamma_r(1) = \text{cov}(r_{t+1}, r_t) &= E(r_t r_{t+1}) = EE(r_t r_{t+1} \mid r_t) \\ &= E\{r_t E(r_{t+1} \mid r_t)\} = 0. \end{aligned} \tag{5.8}$$

Using similar arguments, we may show that $\gamma_r(h) = 0$ for all $h \neq 0$. A similar argument will establish the fact that the error process v_t in (5.5) is also an uncorrelated sequence. If $|\alpha_1| < 1$ and the variance of v_t is finite and constant with respect to time, then based on Property 3.1, (5.5) specifies a causal AR(1) process for r_t^2. Therefore, $E(r_t^2)$ and $\text{var}(r_t^2)$ must be constant with respect to time t. This, implies that

$$\gamma_r(0) = \text{var}(r_t) = E(r_t^2) = \frac{\alpha_0}{1-\alpha_1} \tag{5.9}$$

and, after some manipulations,

$$E(r_t^4) = \frac{3\alpha_0^2}{(1-\alpha_1)^2}\frac{1-\alpha_1^2}{1-3\alpha_1^2}, \tag{5.10}$$

provided $3\alpha_1^2 < 1$. Since

$$\text{var}(r_t^2) = E(r_t^4) - [E(r_t^2)]^2,$$

(5.10) implies that for the variance of the squared returns to be finite, we must have $3\alpha_1^2 < 1$.

Finally we have our restriction for α_1. In particular, if $0 \leq \alpha_1 < 1/\sqrt{3} \approx .577$, then r_t^2 follows a causal AR(1) model with ACF given by $\rho_{r^2}(h) = \alpha_1^h \geq 0$, for all $h > 0$. Since α_1 is restricted to a small interval, the autocorrelations must have fairly small values.

Also, these results imply that the kurtosis, κ, of r_t is

$$\kappa = \frac{E(r_t^4)}{[E(r_t^2)]^2} = 3\frac{1-\alpha_1^2}{1-3\alpha_1^2}, \tag{5.11}$$

which is never smaller than 3, the kurtosis of the normal distribution. Thus, the marginal distribution of the returns, r_t, is leptokurtic, or has "fat tails."

Estimation of the parameters α_0 and α_1 of the ARCH(1) model is typically accomplished by conditional MLE. The conditional likelihood of the data $r_2, \ldots, r_n$ given r_1, is given by

$$L(\alpha_0, \alpha_1 \mid r_1) = \prod_{t=2}^{n} f_{\alpha_0,\alpha_1}(r_t \mid r_{t-1}), \tag{5.12}$$

where the density $f_{\alpha_0,\alpha_1}(r_t \mid r_{t-1})$ is the normal density specified in (5.6). Hence, the conditional weighted sum of squares be minimized, $S(\alpha_0, \alpha_1) \propto -\ln L(\alpha_0, \alpha_1 \mid r_1)$, is given by

$$S(\alpha_0, \alpha_1) = \frac{1}{2}\sum_{t=2}^{n} \ln(\alpha_0 + \alpha_1 r_{t-1}^2) + \frac{1}{2}\sum_{t=2}^{n}\left(\frac{r_t^2}{\alpha_0 + \alpha_1 r_{t-1}^2}\right). \tag{5.13}$$

Estimation is accomplished by numerical methods, as described in Section 3.4.

The ARCH(1) model can be extended to the general ARCH(p) model in an obvious way. That is, (5.3), $r_t = \sigma_t \epsilon_t$, is retained, but (5.4) is extended to

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \cdots + \alpha_p r_{t-p}^2. \tag{5.14}$$

Estimation for ARCH(p) also follows in an obvious way from the discussion of estimation for ARCH(1) models.

It is also possible to combine a regression or an ARMA model for the conditional mean with an ARCH model for the errors. For example, a regression with ARCH(1) errors model would have the observations r_t as linear function of q regressors, $z_t = (z_{t1}, ..., z_{tq})'$, and ARCH(1) noise. For example, we might have

$$r_t = \mu_t + \sigma_t \epsilon_t \,, \tag{5.15}$$

where $\mu_t = \beta' z_t$ and σ_t^2 has ARCH behavior. Similarly, for example, a simple AR-ARCH model would have

$$\mu_t = \phi_0 + \phi_1 r_{t-1}.$$

Of course the model (5.15) can be generalized to have various types of behavior for μ_t.

To fit ARMA-ARCH models, simply follow these two steps:

1. First, look at the P/ACF of the *returns*, r_t, and identify an ARMA structure, if any. There is typically either no autocorrelation or very small autocorrelation and often a low order AR or MA will suffice if needed. Estimate μ_t in order to center the returns if necessary.
2. Look at the P/ACF of the *centered squared returns*, $(r_t - \hat{\mu}_t)^2$, and decide on an ARCH model. If the P/ACF indicate an AR structure (i.e., ACF tails off, PACF cuts off), then fit an ARCH. If the P/ACF indicate an ARMA structure (i.e., both tail off), use the approach discussed after the next example.

Example 5.1 Analysis of U.S. GNP

In Example 3.27, we fit an AR(1) model to the U.S. GNP series and we concluded that the residuals from both fits appeared to behave like a white noise process. Hence, we would propose that $\mu_t = \phi_0 + \phi_1 r_{t-1}$ where r_t is the quarterly growth rate in U.S. GNP.

It has been suggested that the U.S. GNP series has ARCH errors, and in this example, we will investigate this claim. If the GNP noise term is ARCH, the squares of the residuals from the fit should behave like a non-Gaussian AR(1) process, as pointed out in (5.5). Figure 5.1 shows the ACF and PACF of the squared residuals it appears that there may be some dependence, albeit small, left in the residuals. The figure was generated in R as follows.

```
res = resid( sarima(diff(log(gnp)), 1,0,0, details=FALSE)$fit )
acf2(res^2, 20)
```

We used the R package `fGarch` to fit an AR(1)-ARCH(1) model to the U.S. GNP returns with the following results. A partial output is shown; we note that `garch(1,0)` specifies an ARCH(1) in the code below (details later).

```
library(fGarch)
gnpr = diff(log(gnp))
```

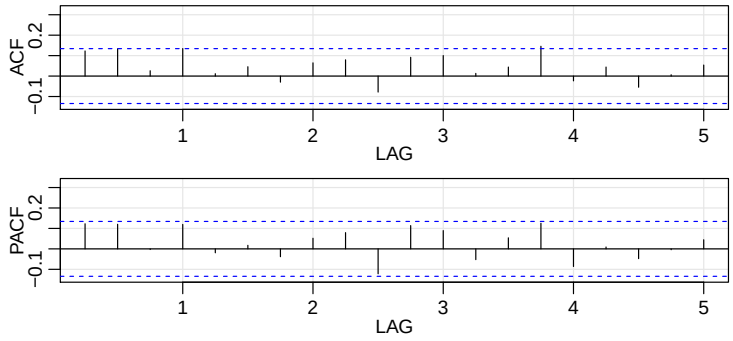


Fig. 5.1. *ACF and PACF of the squares of the residuals from the AR(1) fit on U.S. GNP.*

```
summary( garchFit(~arma(1,0) + garch(1,0), data = gnpr) )
          Estimate   Std.Error  t.value  p.value<- 2-sided
   mu         0.005       0.001    5.867    0.000
   ar1        0.367       0.075    4.878    0.000
   omega      0.000       0.000    8.135    0.000
   alpha1     0.194       0.096    2.035    0.042

   Standardised Residuals Tests:  Statistic p-Value
    Jarque-Bera Test   R    Chi^2     9.118   0.010
    Shapiro-Wilk Test  R    W         0.984   0.014
    Ljung-Box Test     R    Q(20)    23.414   0.269
    Ljung-Box Test     R^2  Q(20)    37.743   0.010
```

Note that the given p-values are two-sided, so they should be halved when considering the ARCH parameters. In this example, we obtain $\hat{\phi}_0 = .005$ (called `mu` in the output) and $\hat{\phi}_1 = .367$ (called `ar1`) for the AR(1) parameter estimates; in Example 3.27 the values were .005 and .347, respectively. The ARCH(1) parameter estimates are $\hat{\alpha}_0 = 0$ (called `omega`) for the constant and $\hat{\alpha}_1 = .194$, which is significant with a p-value of about .02. There are a number of tests that are performed on the residuals [`R`] or the squared residuals [`R^2`]. For example, the Jarque–Bera statistic tests the residuals of the fit for normality based on the observed skewness and kurtosis, and it appears that the residuals have some non-normal skewness and kurtosis. The Shapiro–Wilk statistic tests the residuals of the fit for normality based on the empirical order statistics. The other tests, primarily based on the Q-statistic, are used on the residuals and their squares.

Another extension of ARCH is the generalized ARCH or GARCH model developed by Bollerslev (1986). For example, a GARCH$(1,1)$ model retains (5.15), $r_t = \mu_t + \sigma_t \epsilon_t$, but extends (5.4) as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \tag{5.16}$$

Under the condition that $\alpha_1 + \beta_1 < 1$, using similar manipulations as in (5.5), the GARCH$(1,1)$ model, (5.3) and (5.16), admits a non-Gaussian ARMA$(1,1)$ model for the squared process

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1) r_{t-1}^2 + v_t - \beta_1 v_{t-1}, \tag{5.17}$$

where we have set $\mu_t = 0$ for ease, and where v_t is as defined in (5.5). Representation (5.17) follows by writing (5.3) as

$$r_t^2 - \sigma_t^2 = \sigma_t^2(\epsilon_t^2 - 1)$$
$$\beta_1(r_{t-1}^2 - \sigma_{t-1}^2) = \beta_1\sigma_{t-1}^2(\epsilon_{t-1}^2 - 1),$$

subtracting the second equation from the first, and using the fact that, from (5.16), $\sigma_t^2 - \beta_1\sigma_{t-1}^2 = \alpha_0 + \alpha_1 r_{t-1}^2$, on the left-hand side of the result. The GARCH(p, q) model retains (5.15) and extends (5.16) to

$$\sigma_t^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j r_{t-j}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2. \tag{5.18}$$

Estimation of the model parameters is similar to the estimation of ARCH parameters. We explore these concepts in the following example.

Example 5.2 GARCH Analysis of the DJIA Returns

As previously mentioned, the daily returns of the DJIA shown in Figure 1.4 exhibit classic GARCH features. In addition, there is some low level autocorrelation in the series itself, and to include this behavior, we used the R `fGarch` package to fit an AR(1)-GARCH$(1, 1)$ model to the series using t-errors:

```
library(xts)
djiar = diff(log(djia$Close))[-1]
acf2(djiar)     #  exhibits some autocorrelation  (not shown)
u = resid( sarima(djiar, 1,0,0, details=FALSE)$fit )
acf2(u^2)       #  oozes autocorrelation (not shown)
library(fGarch)
summary(djia.g <- garchFit(~arma(1,0)+garch(1,1), data=djiar,
           cond.dist='std'))
            Estimate   Std.Error  t.value    p.value
   mu      8.585e-04   1.470e-04    5.842   5.16e-09
   ar1    -5.531e-02   2.023e-02   -2.735   0.006239
   omega   1.610e-06   4.459e-07    3.611   0.000305
   alpha1  1.244e-01   1.660e-02    7.497   6.55e-14
   beta1   8.700e-01   1.526e-02   57.022   < 2e-16
   shape   5.979e+00   7.917e-01    7.552   4.31e-14
   ---
   Standardised Residuals Tests:
                              Statistic    p-Value
    Ljung-Box Test     R   Q(10)  16.81507  0.0785575
    Ljung-Box Test     R^2 Q(10)  15.39137  0.1184312
plot(djia.g, which=3) # similar to Figure 5.2
```

To explore the GARCH predictions of volatility, we calculated and plotted part of the data surrounding the financial crises of 2008 along with the one-step-ahead predictions of the corresponding volatility, σ_t^2 as a solid line in Figure 5.2.

Another model that we mention briefly is the *asymmetric power ARCH* model. The model retains (5.3), $r_t = \sigma_t\epsilon_t$, but the conditional variance is modeled as

$$\sigma_t^\delta = \alpha_0 + \sum_{j=1}^{p} \alpha_j(|r_{t-j}| - \gamma_j r_{t-j})^\delta + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^\delta\,. \tag{5.19}$$

Note that the model is GARCH when $\delta = 2$ and $\gamma_j = 0$, for $j \in \{1, \ldots, p\}$. The parameters γ_j ($|\gamma_j| \leq 1$) are the *leverage* parameters, which are a measure of asymmetry, and $\delta > 0$ is the parameter for the power term. A positive [negative]

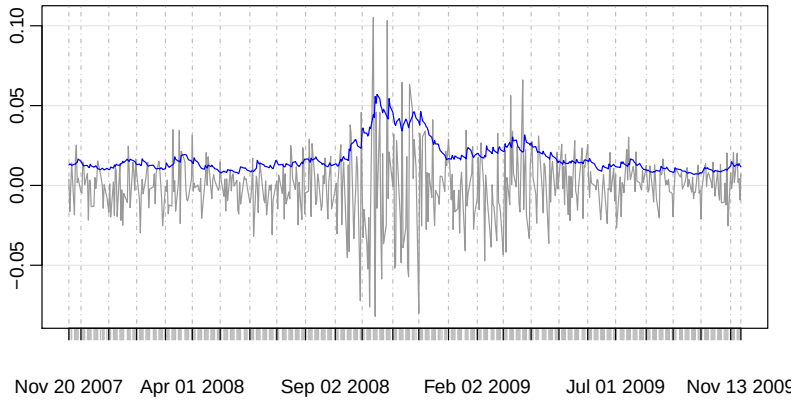


Fig. 5.2. *GARCH one-step-ahead predictions of the DJIA volatility, σ_t, superimposed on part of the data including the financial crisis of 2008.*

value of γ_j's means that past negative [positive] shocks have a deeper impact on current conditional volatility than past positive [negative] shocks. This model couples the flexibility of a varying exponent with the asymmetry coefficient to take the *leverage effect* into account. Further, to guarantee that $\sigma_t > 0$, we assume that $\alpha_0 > 0$, $\alpha_j \geq 0$ with at least one $\alpha_j > 0$, and $\beta_j \geq 0$.

We contiune the analysis of the DJIA returns in the following example.

Example 5.3 APARCH Analysis of the DJIA Returns

The R package `fGarch` was used to fit an AR-APARCH model to the DJIA returns discussed in Example 5.2. As in the previous example, we include an AR(1) in the model to account for the conditional mean. In this case, we may think of the model as $r_t = \mu_t + y_t$ where μ_t is an AR(1), and y_t is APARCH noise with conditional variance modeled as (5.19) with t-errors. A partial output of the analysis is given below. We do not include displays, but we show how to obtain them. The predicted volatility is, of course, different than the values shown in Figure 5.2, but appear similar when graphed.

```
lapply( c('xts', 'fGarch'), library, char=TRUE)  # load 2 packages
summary(fit <- garchFit(~arma(1,0)+aparch(1,1), data=djiar,
          cond.dist='std'))
plot(djia.ap)   # to see all plot options (none shown)
          Estimate  Std. Error  t value   p.value
  mu      5.234e-04   1.525e-04    3.432  0.000598
  ar1    -4.818e-02   1.934e-02   -2.491  0.012727
  omega   1.798e-04   3.443e-05    5.222  1.77e-07
  alpha1  9.809e-02   1.030e-02    9.525   < 2e-16
  gamma1  1.000e+00   1.045e-02   95.731   < 2e-16
  beta1   8.945e-01   1.049e-02   85.280   < 2e-16
  delta   1.070e+00   1.350e-01    7.928  2.22e-15
  shape   7.286e+00   1.123e+00    6.489  8.61e-11
  ---
  Standardised Residuals Tests:
                                  Statistic p-Value
   Ljung-Box Test     R    Q(10)  15.71403  0.108116
   Ljung-Box Test     R^2  Q(10)  16.87473  0.077182
```

In most applications, the distribution of the noise, ϵ_t in (5.3), is rarely normal. The R package `fGarch` allows for various distributions to be fit to the data; see the

help file for information. Some drawbacks of GARCH and related models are as follows. (i) The GARCH model assumes positive and negative returns have the same effect because volatility depends on squared returns; the asymmetric models help alleviate this problem. (ii) These models are often restrictive because of the tight constraints on the model parameters (e.g., for an ARCH(1), $0 \le \alpha_1 < 1/\sqrt{3}$). (iii) The likelihood is flat unless n is very large. (iv) The models tend to overpredict volatility because they respond slowly to large isolated returns.

Various extensions to the original model have been proposed to overcome some of the shortcomings we have just mentioned. For example, we have already discussed the fact that `fGarch` allows for asymmetric return dynamics. In the case of persistence in volatility, the integrated GARCH (IGARCH) model may be used. Recall (5.17) where we showed the GARCH$(1,1)$ model can be written as

$$r_t^2 = \alpha_0 + (\alpha_1 + \beta_1) r_{t-1}^2 + v_t - \beta_1 v_{t-1}$$

and r_t^2 is stationary if $\alpha_1 + \beta_1 < 1$. The IGARCH model sets $\alpha_1 + \beta_1 = 1$, in which case the IGARCH$(1,1)$ model is

$$r_t = \sigma_t \epsilon_t \quad \text{and} \quad \sigma_t^2 = \alpha_0 + (1 - \beta_1) r_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

There are many different extensions to the basic ARCH model that were developed to handle the various situations noticed in practice. Interested readers might find the general discussions in Engle et al. (1994) and Shephard (1996) worthwhile reading. Also, Gouriéroux (1997) gives a detailed presentation of ARCH and related models with financial applications and contains an extensive bibliography. Two excellent texts on financial time series analysis are Chan (2002) and Tsay (2002).

5.2 Unit Root Testing

The use of the first difference $\nabla x_t = (1 - B)x_t$ can be too severe a modification in the sense that the nonstationary model might represent an overdifferencing of the original process.

Consider a causal AR(1) process (we assume throughout this section that the noise is Gaussian),

$$x_t = \phi x_{t-1} + w_t. \tag{5.20}$$

A unit root test provides a way to test whether (5.20) is a random walk (the null case) as opposed to a causal process (the alternative). That is, it provides a procedure for testing

$$H_0 : \phi = 1 \quad \text{versus} \quad H_1 : |\phi| < 1.$$

To see if it is reasonable to assume $\phi - 1 = 0$, an obvious test statistic would be to consider $(\widehat{\phi} - 1)$, appropriately normalized, in the hope to develop an asymptotically normal test statistic, where $\widehat{\phi}$ is one of the optimal estimators discussed in Section 3.4. Note that the distribution in Example 3.22 does not work in this case; if it did, under the null hypothesis, $\widehat{\phi} \dot{\sim} N(1, 0)$, which is nonsense. The theory of Section 3.4 does not work in the null case because the process is not stationary.

However, the test statistic

$$U = n(\widehat{\phi} - 1)$$

can be used, and it is known as the unit root or Dickey-Fuller (DF) statistic, although the actual DF test statistic is normalized a little differently. In this case, the distribution of the test statistic does not have a closed form and quantiles of the distribution must be computed by numerical approximation or by simulation. The R package `tseries` provides this test along with more general tests that we mention briefly.

Toward a more general model, we note that the DF test was established by noting that if $x_t = \phi x_{t-1} + w_t$, then $\nabla x_t = (\phi - 1)x_{t-1} + w_t = \gamma x_{t-1} + w_t$, and one could test $H_0 : \gamma = 0$ by regressing ∇x_t on x_{t-1}. They formed a Wald statistic and derived its limiting distribution. The test was extended to accommodate AR(p) models, $x_t = \sum_{j=1}^{p} \phi_j x_{t-j} + w_t$, as follows. Subtract x_{t-1} from the model to obtain

$$\nabla x_t = \gamma x_{t-1} + \sum_{j=1}^{p-1} \psi_j \nabla x_{t-j} + w_t, \tag{5.21}$$

where $\gamma = \sum_{j=1}^{p} \phi_j - 1$ and $\psi_j = -\sum_{i=j}^{p} \phi_i$ for $j = 2, \ldots, p$. For a quick check of (5.21) when $p = 2$, note that $x_t = (\phi_1 + \phi_2)x_{t-1} - \phi_2(x_{t-1} - x_{t-2}) + w_t$; now subtract x_{t-1} from both sides. To test the hypothesis that the process has a unit root at 1 (i.e., the AR polynoimial $\phi(z) = 0$ when $z = 1$), we can test $H_0 : \gamma = 0$ by estimating γ in the regression of ∇x_t on $x_{t-1}, \nabla x_{t-1}, \ldots, \nabla x_{t-p+1}$, and forming a Wald test based on $t_\gamma = \widehat{\gamma}/\mathrm{se}(\widehat{\gamma})$. This test leads to the so-called augmented Dickey-Fuller test (ADF). While the calculations for obtaining the asymptotic null distribution change, the basic ideas and machinery remain the same as in the simple case. The choice of p is crucial, and we will discuss some suggestions in the example. For ARMA(p, q) models, the ADF test can be used by assuming p is large enough to capture the essential correlation structure; another alternative is the Phillips-Perron (PP) test, which differs from the ADF tests mainly in how they deal with serial correlation and heteroskedasticity in the errors.

One can extend the model to include a constant, or even non-stochastic trend. For example, consider the model

$$x_t = \beta_0 + \beta_1 t + \phi x_{t-1} + w_t.$$

If we assume $\beta_1 = 0$, then under the null hypothesis, $\phi = 1$, the process is a random walk with drift β_0. Under the alternate hypothesis, the process is a causal AR(1) with mean $\mu_x = \beta_0(1 - \phi)$. If we cannot assume $\beta_1 = 0$, then the interest here is testing the null that $(\beta_1, \phi) = (0, 1)$, simultaneously, versus the alternative that $\beta_1 \neq 0$ and $|\phi| < 1$. In this case, the null hypothesis is that the process is a random walk with drift, versus the alternative hypothesis that the process is stationary around a global trend (consider the chicken price series examined in Example 2.1).

Example 5.4 Testing Unit Roots in the Glacial Varve Series

In this example we use the R package `tseries` to test the null hypothesis that the log of the glacial varve series has a unit root, versus the alternate hypothesis that the process is stationary. We test the null hypothesis using the available DF, ADF and PP tests; note that in each case, the general regression equation incorporates a constant and a linear trend. In the ADF test, the default number of AR components included in the model, say k, is $[\![(n-1)^{\frac{1}{3}}]\!]$, which corresponds to the suggested upper bound on the rate at which the number of lags, k, should be made to grow with the sample size for the general ARMA(p,q) setup. For the PP test, the default value of k is $[\![.04n^{\frac{1}{4}}]\!]$.

```
library(tseries)
adf.test(log(varve), k=0)               # DF test
  Dickey-Fuller = -12.8572, Lag order = 0, p-value < 0.01
   alternative hypothesis: stationary
adf.test(log(varve))                    # ADF test
  Dickey-Fuller = -3.5166, Lag order = 8, p-value = 0.04071
   alternative hypothesis: stationary
pp.test(log(varve))                     # PP test
   Dickey-Fuller Z(alpha) = -304.5376,
    Truncation lag parameter = 6, p-value < 0.01
    alternative hypothesis: stationary
```

In each test, we reject the null hypothesis that the logged varve series has a unit root. The conclusion of these tests supports the conclusion of the previous section that the logged varve series is long memory rather than integrated.

5.3 Long Memory and Fractional Differencing

The conventional ARMA(p,q) process is often referred to as a short-memory process because the coefficients in the representation

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

obtained by solving

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

are dominated by exponential decay. As pointed out in Chapter 3, this result implies the ACF of the short memory process $\rho(h) \to 0$ exponentially fast as $h \to \infty$. When the sample ACF of a time series decays slowly, the advice given in Chapter 3 has been to difference the series until it seems stationary. Following this advice with the glacial varve series first presented in Example 3.21 leads to the first difference of the logarithms of the data being represented as a first-order moving average. In Example 3.29, further analysis of the residuals leads to fitting an ARIMA$(1,1,1)$ model,

$$\nabla x_t = \phi \nabla x_{t-1} + w_t + \theta w_{t-1},$$

where we understand x_t is the log-transformed varve series. In particular, the estimates of the parameters (and the standard errors) were $\widehat{\phi} = .23(.05)$, $\widehat{\theta} = -.89(.03)$, and $\widehat{\sigma}_w^2 = .23$. The use of the first difference $\nabla x_t = (1-B)x_t$

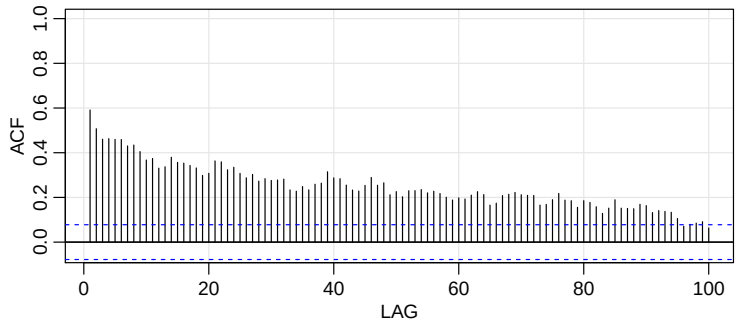


Fig. 5.3. *Sample ACF of the log transformed varve series.*

can be too severe a modification in the sense that the nonstationary model might represent an overdifferencing of the original process.

Long memory (or persistent) time series were considered in Hosking (1981) and Granger and Joyeux (1980) as intermediate compromises between the short memory ARMA type models and the fully integrated nonstationary processes in the Box–Jenkins class. The easiest way to generate a long memory series is to think of using the difference operator $(1-B)^d$ for fractional values of d, say, $0 < d < .5$, so a basic long memory series gets generated as

$$(1-B)^d x_t = w_t, \tag{5.22}$$

where w_t still denotes white noise with variance σ_w^2. The fractionally differenced series (5.22), for $|d| < .5$, is often called *fractional noise* (except when d is zero). Now, d becomes a parameter to be estimated along with σ_w^2. Differencing the original process, as in the Box–Jenkins approach, may be thought of as simply assigning a value of $d = 1$. This idea has been extended to the class of fractionally integrated ARMA, or ARFIMA models, where $-.5 < d < .5$; when d is negative, the term antipersistent is used. Long memory processes occur in hydrology (see Hurst, 1951, and McLeod and Hipel, 1978) and in environmental series, such as the varve data we have previously analyzed, to mention a few examples. Long memory time series data tend to exhibit sample autocorrelations that are not necessarily large (as in the case of $d = 1$), but persist for a long time. Figure 5.3 shows the sample ACF, to lag 100, of the log-transformed varve series, which exhibits classic long memory behavior:

```
acf1(log(varve), 100)
```

To investigate its properties, we can use the binomial expansion ($d > -1$) to write

$$w_t = (1-B)^d x_t = \sum_{j=0}^{\infty} \pi_j B^j x_t = \sum_{j=0}^{\infty} \pi_j x_{t-j} \tag{5.23}$$

where

$$\pi_j = \frac{\Gamma(j-d)}{\Gamma(j+1)\Gamma(-d)} \tag{5.24}$$

with $\Gamma(x+1) = x\Gamma(x)$ being the gamma function. Similarly ($d < 1$), we can write

$$x_t = (1-B)^{-d} w_t = \sum_{j=0}^{\infty} \psi_j B^j w_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} \tag{5.25}$$

where

$$\psi_j = \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)}. \tag{5.26}$$

When $|d| < .5$, the processes (5.23) and (5.25) are well-defined stationary processes (see Brockwell and Davis, 1991, for details). In the case of fractional differencing, however, the coefficients satisfy $\sum \pi_j^2 < \infty$ and $\sum \psi_j^2 < \infty$ as opposed to the absolute summability of the coefficients in ARMA processes.

Using the representation (5.25)–(5.26), and after some nontrivial manipulations, it can be shown that the ACF of x_t is

$$\rho(h) = \frac{\Gamma(h+d)\Gamma(1-d)}{\Gamma(h-d+1)\Gamma(d)} \sim h^{2d-1} \tag{5.27}$$

for large h. From this we see that for $0 < d < .5$

$$\sum_{h=-\infty}^{\infty} |\rho(h)| = \infty$$

and hence the term *long memory*.

In order to examine a series such as the varve series for a possible long memory pattern, it is convenient to look at ways of estimating d. Using (5.24) it is easy to derive the recursions

$$\pi_{j+1}(d) = \frac{(j-d)\pi_j(d)}{(j+1)}, \tag{5.28}$$

for $j = 0, 1, \ldots$, with $\pi_0(d) = 1$. Maximizing the joint likelihood of the errors under normality, say, $w_t(d)$, will involve minimizing the sum of squared errors

$$Q(d) = \sum w_t^2(d).$$

The usual Gauss–Newton method, described in §3.6, leads to the expansion

$$w_t(d) = w_t(d_0) + w_t'(d_0)(d - d_0),$$

where

$$w_t'(d_0) = \left.\frac{\partial w_t}{\partial d}\right|_{d=d_0}$$

and d_0 is an initial estimate (guess) at to the value of d. Setting up the usual regression leads to

$$d = d_0 - \frac{\sum_t w_t'(d_0) w_t(d_0)}{\sum_t w_t'(d_0)^2}. \tag{5.29}$$

The derivatives are computed recursively by differentiating (5.28) successively with respect to d: $\pi_{j+1}'(d) = [(j-d)\pi_j'(d) - \pi_j(d)]/(j+1)$, where $\pi_0'(d) = 0$. The errors are computed from an approximation to (5.23), namely,

$$w_t(d) = \sum_{j=0}^{t} \pi_j(d) x_{t-j}. \tag{5.30}$$

It is advisable to omit a number of initial terms from the computation and start the sum, (5.29), at some fairly large value of t to have a reasonable approximation.

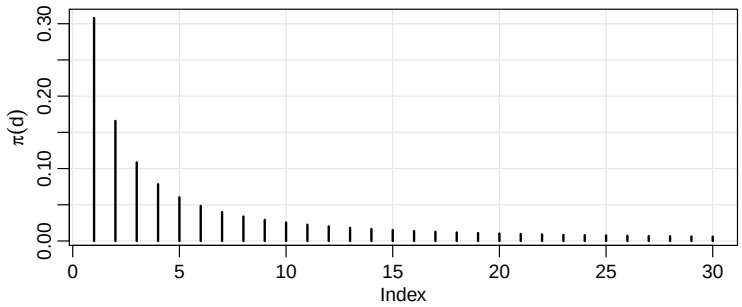


Fig. 5.4. *Coefficients* $\pi_j(.373)$, $j = 1, 2, \ldots, 30$ *in the representation* (5.28).

Example 5.5 Long Memory Fitting of the Glacial Varve Series

We consider analyzing the glacial varve series discussed in Example 2.7 and Example 3.21. Figure 2.7 shows the original and log-transformed series (which we denote by x_t). In Example 3.29, we noted that x_t could be modeled as an ARIMA$(1,1,1)$ process. We fit the fractionally differenced model, (5.22), to the mean-adjusted series, $x_t - \bar{x}$. Applying the Gauss–Newton iterative procedure previously described leads to a final value of $d = .373$, which implies the set of coefficients $\pi_j(.373)$, as given in Figure 5.4 with $\pi_0(.373) = 1$. We can compare roughly the performance of the fractional difference operator with the ARIMA model by examining the autocorrelation functions of the two residual series as shown in Figure 5.5. The ACFs of the two residual series are roughly comparable with the white noise model.

To perform this analysis in R, first download and install the `arfima` package. Then use

```
library(arfima)
summary(varve.fd <- arfima(log(varve), order = c(0,0,0)))
  d.hat = 0.3728, se(d,hat) = 0.0273   (summary of the results)
# residual stuff
innov = resid(varve.fd)
tsplot(innov[[1]])
acf1(innov[[1]])
```

Forecasting long memory processes is similar to forecasting ARIMA models. That is, (5.23) and (5.28) can be used to obtain the truncated forecasts

$$\widetilde{x}^n_{n+m} = -\sum_{j=1}^{n} \pi_j(\widehat{d})\, \widetilde{x}^n_{n+m-j}, \tag{5.31}$$

for $m = 1, 2, \ldots$. Error bounds can be approximated by using

$$P^n_{n+m} = \widehat{\sigma}^2_w \left(\sum_{j=0}^{m-1} \psi^2_j(\widehat{d}) \right) \tag{5.32}$$

where, as in (5.28),

$$\psi_j(\widehat{d}) = \frac{(j+\widehat{d})\psi_j(\widehat{d})}{(j+1)}, \tag{5.33}$$

with $\psi_0(\widehat{d}) = 1$.

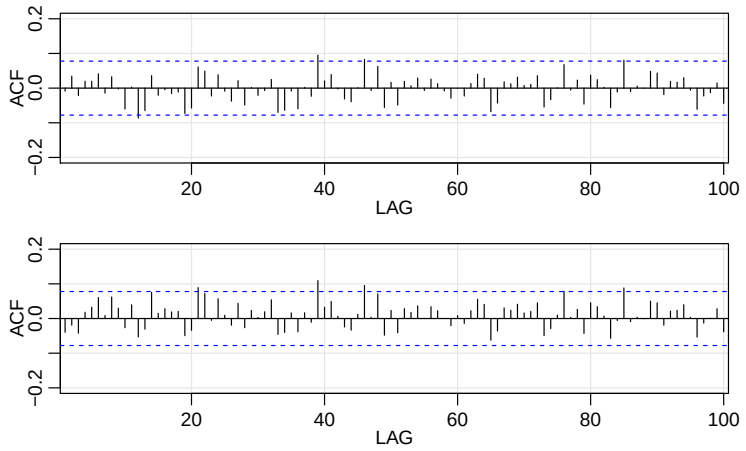


Fig. 5.5. *ACF of residuals from the ARIMA*$(1,1,1)$ *fit to the logged varve series (top) and of the residuals from the long memory model fit,* $(1-B)^d x_t = w_t$*, with* $d = .373$ *(bottom).*

No obvious short memory ARMA-type component can be seen in the ACF of the residuals from the fractionally differenced varve series shown in Figure 5.5. It is natural, however, that cases will exist in which substantial short memory-type components will also be present in data that exhibits long memory. Hence, it is natural to define the general ARFIMA(p,d,q), $-.5 < d < .5$ process as

$$\phi(B)\nabla^d(x_t - \mu) = \theta(B)w_t, \tag{5.34}$$

where $\phi(B)$ and $\theta(B)$ are as given in Chapter 3. Writing the model in the form

$$\phi(B)\pi_d(B)(x_t - \mu) = \theta(B)w_t \tag{5.35}$$

makes it clear how we go about estimating the parameters for the more general model. Forecasting for the ARFIMA(p,d,q) series can be easily done, noting that we may equate coefficients in

$$\phi(z)\psi(z) = (1-z)^{-d}\theta(z) \tag{5.36}$$

and

$$\theta(z)\pi(z) = (1-z)^d\phi(z) \tag{5.37}$$

to obtain the representations

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

and

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

We then can proceed as discussed in (5.31) and (5.32).

Problems

5.1 Weekly crude oil spot prices in dollars per barrel are in `oil`. Investigate whether the growth rate of the weekly oil price exhibits GARCH behavior. If so, fit an appropriate model to the growth rate.

5.2 The `stats` package of R contains the daily closing prices of four major European stock indices; type `help(EuStockMarkets)` for details. Fit a GARCH model to the returns of one of these series and discuss your findings. (Note: The data set contains actual values, and not returns. Hence, the data must be transformed prior to the model fitting.)

5.3 The data set `arf` is 1000 simulated observations from an ARFIMA$(1,1,0)$ model with $\phi = .75$ and $d = .4$.

(a) Plot the data and comment.
(b) Plot the ACF and PACF of the data and comment.
(c) Estimate the parameters and test for the significance of the estimates $\widehat{\phi}$ and $\widehat{d}$.
(d) Explain why, using the results of parts (a) and (b), it would seem reasonable to difference the data prior to the analysis. That is, if x_t represents the data, explain why we might choose to fit an ARMA model to ∇x_t.
(e) Plot the ACF and PACF of ∇x_t and comment.
(f) Fit an ARMA model to ∇x_t and comment.

5.4 Compute the sample ACF of the absolute values of the NYSE returns displayed in Figure 1.4 up to lag 200, and comment on whether the ACF indicates long memory. Fit an ARFIMA model to the absolute values and comment.

5.5 Plot the global temperature series, `globtemp`, and then test whether there is a unit root versus the alternative that the process is stationary using the three tests, DF, ADF, and PP, discussed in Example 5.4. Comment.

5.6 Plot the GNP series, `gnp`, and then test for a unit root against the alternative that the process is explosive. State your conclusion.

5.7 Verify (5.21).

Appendix R

R Supplement

R.1 First Things First

R is an open source programming language and software environment for statistical computing and graphics that runs on most operating systems. It is an interpreted language and is accessed through a command-line interpreter. A user types a command, presses enter, and the answer is returned.

To obtain R, point your browser to the Comprehensive R Archive Network (CRAN), http://cran.r-project.org/ and download and install it. The installation includes help files and some user manuals. An internet search can pull up various short tutorials and YouTube videos.

RStudio (https://www.rstudio.com/) can make using R much easier and we recommend using it. It is an open source integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. This tutorial does not assume you are using RStudio; if you do use it, a number of the command driven tasks can be accomplished by pointing and clicking.

There are 15 simple exercises in this appendix that will help you get used to using R. For example,

▶ **Exercise 1:** Install R and RStudio (optional) now.

Solution: Follow the directions above.

R.2 Packages

At this point, you should have R (or RStudio) up and running. The capabilities of R are extended through packages. R comes with a number of preloaded packages that are available immediately. There are "base" packages that install with R and load automatically. Then there are "priority" packages that are installed with R, but not loaded automatically. Finally, there are user-created packages that must be installed and loaded into R before use. If you are using RStudio, there is a Packages tab to help you manage your packages.

Most packages can be obtained from CRAN and its mirrors. For example, in Chapter 1, we will use the eXtensible Time Series package `xts`. To install `xts`, start R and type

```
install.packages("xts")
```

If you are using RStudio, then use Install from the Packages tab. If asked to choose a repository, select 0-Cloud, the first choice, and that will find your closest repository. To use the package, you first load it by issuing the command

```
library(xts)
```

If you're using RStudio, just click the box next to the package name.

R.2.1 Latest Version of ASTSA

The package for this course is called `astsa`. The latest version of the package will not always be at CRAN, but will be available from GitHub. If you want to install or update the package to the most recent version, you just need the following two lines:

```
install.packages("devtools")
devtools::install_github("nickpoison/astsa")
```

Details about the updates and the current version of the package are on the `astsa news page`.

▶ Exercise 2: Install the most recent version of `astsa`.

Solution: Start R or RStudio, paste in the following two lines. Easy.

```
install.packages("devtools")
devtools::install_github("nickpoison/astsa")  # reissue if you get an error
```

If you don't use RStudio, you may want to create a `.First` function as follows,

```
.First <- function(){library(astsa)}
```

and save the workspace when you quit; `astsa` will be loaded at every start until you change `.First`.

R.3 Getting Help

In RStudio, there is a Help tab. Otherwise, the R html help system can be started by issuing the command

```
help.start()
```

The help files for installed packages can also be found there. *Notice the parentheses* in all the commands above; they are necessary to run scripts. If you simply type

```
help.start
```

nothing will happen and you will just see the commands that make up the script. To get help for a particular command, say `library`, which we have already used, do this:

```
help(library)
?library        # same thing
```

And we state the obvious (well, not obvious to all):

If you can't figure out how to do something, do an internet search.

R.4 Basics

The convention throughout the text is that R code is in `blue`, output is `purple` and comments are `# green`. Get comfortable, then start her up and try some simple tasks.

```
2+2           # addition
 [1] 5
5*5 + 2       # multiplication and addition
 [1] 27
5/5 - 3       # division and subtraction
 [1] -2
log(exp(pi))  # log, exponential, pi
 [1] 3.141593
sin(pi/2)     # sinusoids
 [1] 1
2^(-2)        # power
 [1] 0.25
sqrt(8)       # square root
 [1] 2.828427
-1:5           # sequences
 [1] -1  0  1  2  3  4  5
seq(1, 10, by=2)   # sequences
 [1] 1 3 5 7 9
rep(2, 3)     # repeat 2 three times
 [1] 2 2 2
```

▶ Exercise 3: In one line, add 3 to 2 and multiply the result by 5.

Solution: The answer is not a teen.

```
2 + 3 * 5    # wrong
 [1] 17
(2 + 3) * 5  # right
 [1] 25
```

▶ Exercise 4: Verify that $1/i = -i$ where $i = \sqrt{-1}$.

Solution: The complex number i is written as `1i` in R.

```
1/1i
 [1] 0-1i    # complex numbers are displayed as a+bi
```

▶ Exercise 5: Calculate i^2.

Solution: Easy.

▶ Exercise 6: Calculate $\cos(\pi/2)$.

Solution: You won't get 0 exactly, but you will get machine precision 0. Here you'll see what it looks like.

Next, we'll use *assignment* to make some vector *objects*:

```
x <- 1 + 2  # put 1 + 2 in object x
x = 1 + 2   # same as above with fewer keystrokes
1 + 2 -> x  # same
x           # view object x
 [1]  3
(y = 9 * 3)   # put 9 times 3 in y and view the result
 [1] 27
(z = rnorm(5))  # put 5 standard normals into z and print z
 [1]  0.96607946  1.98135811 -0.06064527  0.31028473  0.02046853
```

Vectors can be of various types, and they can be put together using `c()` [concatenate or combine]; for example

```
x <- c(1, 2, 3)             # numeric vector
y <- c("one","two","three") # character vector
z <- c(TRUE, TRUE, FALSE)   # logical vector
```

Missing values are represented by the symbol `NA`, ∞ by `Inf` and impossible values are `NaN`. Here are some examples:

```
( x = c(0, 1, NA) )
 [1]  0  1 NA
2*x
 [1]  0  2 NA
is.na(x)
 [1] FALSE FALSE  TRUE
x/0
 [1] NaN Inf  NA
```

It is worth pointing out R's *recycling rule* for doing arithmetic. Note the use of the semicolon for multiple commands on one line.

```
x = c(1, 2, 3, 4);  y = c(2, 4, 6, 8)
z = c(10, 20);      w = c(8, 3, 2)
x * y    # 1*2, 2*4, 3*6, 4*8
 [1]  2  8 18 32
x + z    # 1+10, 2+20, 3+10, 4+20
 [1] 11 22 13 24
y + w    # oops
 [1] 10  7  8 16
 Warning message:
  In y + w : longer object length is not a multiple of
   shorter object length
```

▶ Exercise 7: Why was `y+w` above the vector (10, 7, 8, 16) and why is there a warning?

Solution: To get started, `y+w = (2+8, 4+3, ...)` ...

The following commands are useful:

```
ls()                # list all objects
 "dummy" "mydata" "x" "y" "z"
ls(pattern = "my")  # list every object that contains "my"
 "dummy" "mydata"
rm(dummy)           # remove object "dummy"
rm(list=ls())       # remove almost everything (use with caution)
data()              # list of available data sets
help(ls)            # specific help (?ls is the same)
getwd()             # get working directory
setwd()             # change working directory
q()                 # end the session (keep reading)
```

and a reference card may be found here:
https://cran.r-project.org/doc/contrib/Short-refcard.pdf.

When you quit, R will prompt you to save an image of your current workspace. Answering *yes* will save the work you have done so far, and load it when you next start R. We have never regretted selecting *yes*, but we have regretted answering *no*.

If you want to **keep your files separated for different projects**, then having to set the working directory each time you run R is a pain. If you use RStudio, then you can easily create separate projects (from the menu File):
https://support.rstudio.com/hc/en-us/articles/200526207.

Otherwise, there are easy work-arounds, but it depends on your OS. In Windows, copy the R or RStudio shortcut into the directory you want to use for your project. Right click on the shortcut icon, select Properties, and remove the text in the Start in: field; leave it blank and press OK. Then start R or RStudio from that shortcut.

▶ Exercise 8: Create a directory that you will use for the course and use the tricks previously mentioned to make it your working directory (or use the default if you don't care). Load `astsa` and use help to find out what's in the data file `cpg`. Write `cpg` as text to your working directory.
Solution: Assuming you started R in the working directory:

```
library(astsa)
help(cpg)     # or ?cpg
  Median ...
write(cpg, file="zzz.txt", ncolumns=1)  # zzz so it's easy to find
```

▶ Exercise 9: Find the file `zzz.txt` previously created (leave it there for now).
Solution: In RStudio, use the Files tab. Otherwise, go to your working directory:

```
getwd()
 "C:\TimeSeries"
```

Now find the file and look at it; there should be 29 numbers in one column.

To create your own data set, you can make a data vector as follows:

```
mydata = c(1,2,3,2,1)
```

Now you have an object called `mydata` that contains five elements. R calls these objects *vectors* even though they have no dimensions (no rows, no columns); they do have order and length:

```
mydata         # display the data
 [1] 1 2 3 2 1
mydata[3]      # 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
```

If you have an external data set, you can use `scan` or `read.table` (or some variant) to input the data. For example, suppose you have an ASCII (text) data file called `dummy.txt` in your working directory, and the file looks like this:

```
1 2 3 2 1
9 0 2 1 0
```

```
(dummy = scan("dummy.txt") )        # scan and view it
 Read 10 items
  [1] 1 2 3 2 1 9 0 2 1 0
(dummy = read.table("dummy.txt") )  # read and view it
 V1 V2 V3 V4 V5
  1  2  3  2  1
  9  0  2  1  0
```

There is a difference between `scan` and `read.table`. The former produced a data vector of 10 items while the latter produced a *data frame* with names `V1` to `V5` and two observations per variate.

▶ Exercise 10: Scan and view the data in the file `zzz.txt` that you previously created.

Solution: Hopefully it's in your working directory:

```
(cost_per_gig = scan("zzz.txt") )  # read and view
 Read 29 items
 [1] 2.13e+05 2.95e+05 2.60e+05 1.75e+05 1.60e+05
 [6] 7.10e+04 6.00e+04 3.00e+04 3.60e+04 9.00e+03
[11] 7.00e+03 4.00e+03 ...
```

When you use `read.table` or similar, you create a data frame. In this case, if you want to list (or use) the second variate, `V2`, you would use

```
dummy$V2
 [1] 2 0
```

and so on. You might want to look at the help files `?scan` and `?read.table` now. Data frames (`?data.frame`) are "used as the fundamental data structure by most of R's modeling software." Notice that R gave the columns of `dummy` generic names, `V1, ..., V5`. You can provide your own names and then use the names to access the data without the use of `$` as above.

```
colnames(dummy) = c("Dog", "Cat", "Rat", "Pig", "Man")
attach(dummy)    # this can cause problems; see ?attach
Cat
 [1] 2 0
Rat*(Pig - Man)  # animal arithmetic
 [1] 3 2
head(dummy)      # view the first few lines of a data file
detach(dummy)    # clean up
```

R is case sensitive, thus `cat` and `Cat` are different. Also, `cat` is a reserved name (`?cat`) in R, so using `"cat"` instead of `"Cat"` may cause problems later. It is noted that `attach` can lead to confusion: *The possibilities for creating errors when using attach are numerous. Avoid.* If you use it, it's best to clean it up when you're done.

You may also include a *header* in the data file to avoid `colnames()`. For example, if you have a *comma separated values* file `dummy.csv` that looks like this,

```
Dog,Cat,Rat,Pig,Man
1,2,3,2,1
9,0,2,1,0
```

then use the following command to read the data.

```
(dummy = read.csv("dummy.csv"))
  Dog Cat Rat Pig Man
1   1   2   3   2   1
2   9   0   2   1   0
```

The default for `.csv` files is `header=TRUE`; type `?read.table` for further information on similar types of files.

Two commands that are used frequently to manipulate data are `cbind` for *column binding* and `rbind` for *row binding*. The following is an example.

```
x = runif(4)  # generate 4 values from a uniform(0,1) into object x
y = runif(4)  # generate 4 more and put them into object y
cbind(x,y)    # column bind the two vectors (4 by 2 matrix)
          x         y
[1,] 0.6547304 0.7503984
[2,] 0.8222048 0.1335557
[3,] 0.4555755 0.2151735
[4,] 0.9843289 0.8483795
```

```
rbind(x,y)    # row bind the two vectors (2 by 4 matrix)
          [,1]      [,2]      [,3]      [,4]
  x  0.6547304 0.8222048 0.4555755 0.9843289
  y  0.7503984 0.1335557 0.2151735 0.8483795
```

Summary statistics are fairly easy to obtain. We will simulate 25 normals with $\mu = 10$ and $\sigma = 4$ and then perform some basic analyses. The first line of the code is `set.seed`, which fixes the seed for the generation of pseudorandom numbers. Using the same seed yields the same results; to expect anything else would be insanity.

```
set.seed(90210)          # so you can reproduce these results
x = rnorm(25, 10, 4)     # generate the data
c( mean(x), median(x), var(x), sd(x) )  # guess
 [1]  9.473883  9.448511   13.926701  3.731850
c( min(x), max(x) )  # smallest and largest values
 [1]  2.678173 17.326089
which.max(x)   # index of the max (x[25] in this case)
 [1] 25
summary(x)     # a five number summary with six numbers
     Min. 1st Qu. Median   Mean  3rd Qu.    Max.
   2.678   7.824   9.449  9.474   11.180  17.330
boxplot(x);  hist(x);  stem(x)   # visual summaries (not shown)
```

▶ Exercise 11: Generate 100 standard normals and draw a boxplot of the results.

Solution: You can do it all in one line.

```
boxplot(rnorm(100))
```

It can't hurt to learn a little about programming in R because you will see some of it along the way. First, let's try a simple example of a function that returns the reciprocal of a number:

```
oneover <- function(x){ 1/x }
oneover(0)
 [1] Inf
oneover(-4)
 [1] -0.25
```

Now 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 standard Cauchy distribution (the ratio of independent standard normals).

```
1 crazy <- function(num) {
2   x <- c()
3   for (n in 1:num) { x[n] <- mean(rcauchy(n)) }
4   plot(x, type="l", xlab="sample size", ylab="sample mean")
5   }
```

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 an empty vector, `x`, 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. The fifth line closes the function. To use `crazy` ending with sample of size of 200, type

```
crazy(200)
```

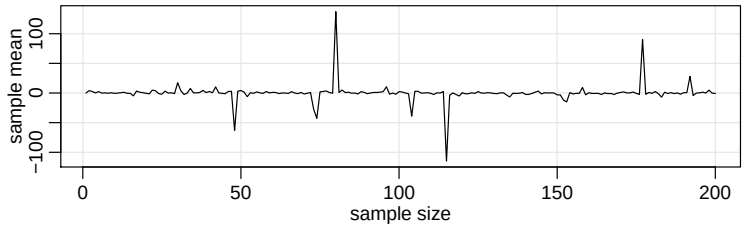


Fig. R.1. *Crazy Cauchy example.*

and you will get a graphic that looks like Figure R.1.

A script can have multiple inputs, for example, guess what this does:

```
pwr <- function(x,y){ x^y }
pwr(25, .5)  # and try it
  [1] 5
```

▶ Exercise 12: Write a simple function to return the sum of two given numbers and then use it to see if it works.

Solution: It's similar to the previous example (avoid using reserved names such as `sum`).

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 `T = 9` so that `T` is no longer `TRUE`.

R.5 Regression and Time Series Primer

These topics run throughout the text, but we'll give a brief introduction here.

The workhorse for regression in R is `lm()`. Suppose we want to fit a simple linear regression, $y = \alpha + \beta x + \epsilon$. In R, the formula is written as `y~x`: We'll simulate our own data and do a simple example first.

```
set.seed(666)           # fixes initial value of generation algorithm
x = rnorm(10)           # generate 10 standard normals
y = 1 + 2*x + rnorm(10) # generate a simple linear model
summary(fit <- lm(y~x)) # fit the model - gets results
 Coefficients:
             Estimate Std. Error t value Pr(>|t|)
  (Intercept)   1.0405     0.2594   4.012  0.00388
  x             1.9611     0.1838  10.672 5.21e-06
  ---
  Residual standard error: 0.8183 on 8 degrees of freedom
  Multiple R-squared:  0.9344,    Adjusted R-squared:  0.9262
  F-statistic: 113.9 on 1 and 8 DF,  p-value: 5.214e-06
plot(x, y)              # scatterplot of generated data
abline(fit, col=4)      # add fitted blue line to the plot
```

Note that we put the results of `lm(y~x)` into an object we called `fit`; this object contains all of the information about the regression. Then we used `summary` to display some of the results and used `abline` to plot the fitted line. The command `abline` is useful for drawing horizontal and vertical lines also.

▶ Exercise 13: Add red horizontal and vertical dashed lines to the previously generated graph to show that the fitted line goes through the point $(\bar{x}, \bar{y})$.
Solution: Add the following two lines to the above graph:

```
abline(h=mean(y), col=2, lty=2)  # color 2 is 'red' and lty 2 is 'dashed'
abline( ?? )   # your turn
# now use the graphical device to save your graph; see Figure R.2.
```

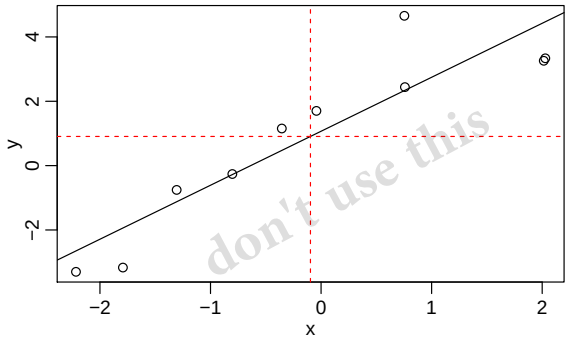


Fig. R.2. *Full plot for Exercise 13.*

All sorts of information can be extracted from the `lm` object, which we called `fit`. For example,

```
plot(resid(fit))    # will plot the residuals (not shown)
fitted(fit)         # will display the fitted values (not shown)
```

We'll get back to regression later after we focus a little on time series. To create a time series object, use the command `ts`. Related commands are `as.ts` to coerce an object to a time series and `is.ts` to test whether an object is a time series. First, make a small data set:

```
(mydata = c(1,2,3,2,1) ) # make it and view it
  [1] 1 2 3 2 1
```

Make it an annual time series that starts in 1990:

```
(mydata = ts(mydata, start=1990) )
  Time Series:
  Start = 1990
  End = 1994
  Frequency = 1
  [1] 1 2 3 2 1
```

Now make it a quarterly time series that starts in 1990-III:

```
(mydata = ts(mydata, start=c(1990,3), frequency=4) )
       Qtr1 Qtr2 Qtr3 Qtr4
  1990              1    2
  1991    3    2    1
time(mydata)  # view the sampled times
          Qtr1    Qtr2    Qtr3    Qtr4
  1990                  1990.50 1990.75
  1991 1991.00 1991.25 1991.50
```

To use part of a time series object, use `window()`:

```
(x = window(mydata, start=c(1991,1), end=c(1991,3) ))
       Qtr1 Qtr2 Qtr3
  1991    3    2    1
```

Next, we'll look at lagging and differencing, which are fundamental transformations used frequently in the analysis of time series. For example, if I'm interested in predicting todays from yesterdays, I would look at the relationship between x_t and its lag, x_{t-1}.

First make a simple series, x_t:

```
x = ts(1:5)
```

Now, column bind (`cbind`) lagged values of x_t and you will notice that `lag(x)` is *forward* lag, whereas `lag(x, -1)` is *backward* lag.

```
cbind(x, lag(x), lag(x,-1))
     x   lag(x)   lag(x, -1)
0   NA      1        NA
1    1      2        NA
2    2      3         1
3    3      4         2 <- in this row, for example, x is 3,
4    4      5         3     lag(x) is ahead at 4, and
5    5     NA         4     lag(x,-1) is behind at 2
6   NA     NA         5
```

Compare `cbind` and `ts.intersect`:

```
ts.intersect(x, lag(x,1), lag(x,-1))
  Time Series:  Start = 2  End = 4  Frequency = 1
     x   lag(x, 1)   lag(x, -1)
2    2          3           1
3    3          4           2
4    4          5           3
```

For discrete-time series, finite differences are used like differentials. To difference a series, $\nabla x_t = x_t - x_{t-1}$, use

```
diff(x)
```

but note that

```
diff(x, 2)
```

is $x_t - x_{t-2}$ and *not* second order differencing. For second order differencing, that is, $\nabla^2 x_t = \nabla(\nabla x_t)$, do one of these:

```
diff(diff(x))
diff(x, diff=2)   # same thing
```

and so on for higher order differencing.

You have to be careful if you use `lm()` for lagged values of a time series. If you use `lm()`, then what you have to do is align the series using `ts.intersect`. Please read the warning *Using time series* in the `lm()` help file [`help(lm)`]. Here is an example regressing `astsa` data, weekly cardiovascular mortality (M_t `cmort`) on particulate pollution (P_t `part`) at the present value and lagged four weeks (P_{t-4} `part4`). The model is

$$M_t = \alpha + \beta_1 P_t + \beta_2 P_{t-4} + w_t\,,$$

where we assume w_t is the usual normal regression error term. First, we create `ded`, which consists of the intersection of the three series:

```
ded = ts.intersect(cmort, part, part4=lag(part,-4))
```

Now the series are all aligned and the regression will work.

```
summary(fit <- lm(cmort~part+part4, data=ded, na.action=NULL) )
 Coefficients:
             Estimate Std. Error t value Pr(>|t|)
 (Intercept) 69.01020    1.37498  50.190  < 2e-16
 part         0.15140    0.02898   5.225 2.56e-07
 part4        0.26297    0.02899   9.071  < 2e-16
```

```
---
Residual standard error: 8.323 on 501 degrees of freedom
Multiple R-squared:  0.3091,    Adjusted R-squared:  0.3063
F-statistic: 112.1 on 2 and 501 DF,  p-value: < 2.2e-16
```

There was no need to rename `lag(part,-4)` to `part4`, it's just an example of what you can do. Also, `na.action=NULL` is necessary to retain the time series attributes. It should be there whenever you do time series regression.

▶ Exercise 14: Rerun the previous example of mortality on pollution but without making a data frame. In this case, the lagged pollution value gets kicked out of the regression because R sees `part` and `part4` as the same thing.

Solution: First lag particulates and then put it in to the regression.

```
part4 <- lag(part, -4)
summary(fit <- lm(cmort~ part + part4, na.action=NULL) )
```

In Problem 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(cycle(jj) )     # make (Q)uarter factors
reg   = lm(log(jj)~ 0 + trend + Q, na.action=NULL)  # no intercept
model.matrix(reg)              # view the model design matrix
        trend Q1 Q2 Q3 Q4
    1  -10.00  1  0  0  0
    2   -9.75  0  1  0  0
    3   -9.50  0  0  1  0
    .     .    .  .  .  .
    .     .    .  .  .  .
summary(reg)                    # view the results (not shown)
```

R.6 Graphics

We introduced some graphics without saying much about it. There are various packages available for producing fabulous graphics, but for quick and easy graphing of time series, the R base graphics does fine wit a little help from `tsplot`, which is available in the `astsa` package. As seen in Chapter 1, a time series may be plotted in a few lines, such as

```
tsplot(globtemp)
```

in Example 1.2, or the multifigure plot

```
plot.ts( cbind(soi, rec) )
```

which we made little fancier in Example 1.4:

```
par(mfrow = c(2,1))   # ?par for details
tsplot(soi, ylab='', xlab='', main='Southern Oscillation Index')
tsplot(rec, ylab='', xlab='', main='Recruitment')
```

If you are using a word processor and you want to be able to paste the graphic in the document, then you can print directly to a png by replacing line 1 with something like

```
png(file="globtemp.png", width=480, height=360)  # default is 480 x 480 px
```

but you have to turn the device off to complete the file save:

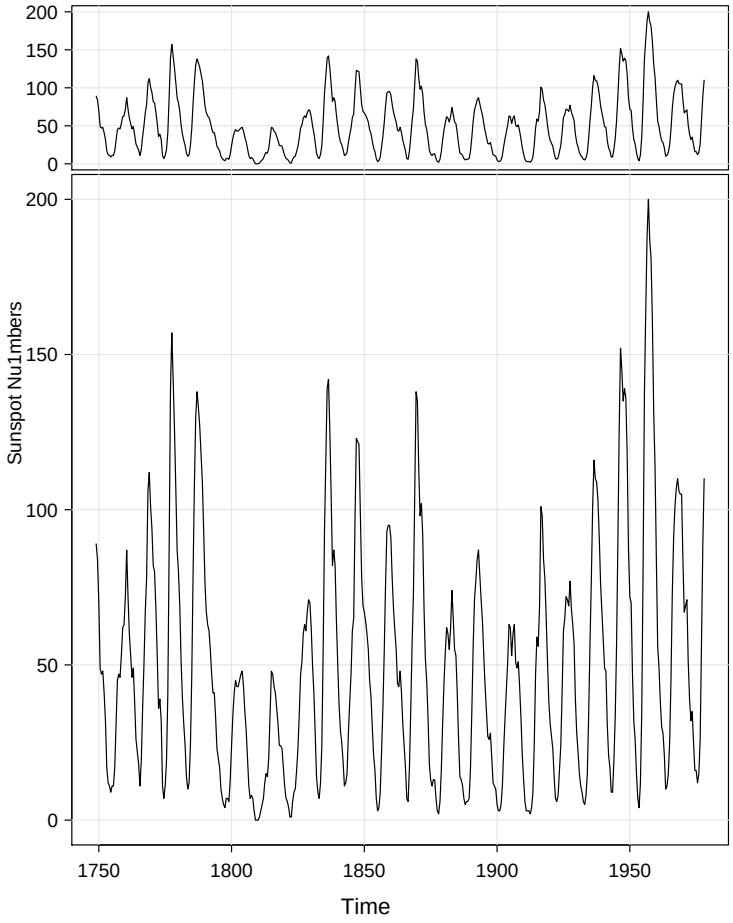


Fig. R.3. *The sunspot numbers plotted in different-sized boxes, demonstrating that the dimensions of the graphic matters when displaying time series data.*

```
dev.off()
```

In RStudio, simply use the Export tab under Plots.

For plotting many time series, `plot.ts` and `ts.plot` are also available using R base graphics. If the series are all on the same scale, it might be useful to do the following:

```
ts.plot(cmort, tempr, part, col=2:4)
legend('topright', legend=c('M','T','P'), lty=1, col=2:4)
```

This produces a plot of all three series on the same axes with different colors, and then adds a legend. We are not restricted to using basic colors; an internet search on 'R colors' is helpful. The following code gives separate plots of each different series (with a limit of 10):

```
plot.ts(cbind(cmort, tempr, part) )
plot.ts(eqexp)                          # you will get a warning
plot.ts(eqexp[,9:16], main='Explosions') # but this works
```

Finally, we mention that size matters when plotting time series. Figure R.3 shows the sunspot numbers discussed in Problem 4.6 plotted with varying dimension size as follows.

```
dev.new(height=8.75)
layout(matrix(1:2), height=c(.2,.8))
par(mar=c(.2,3.5,0,.5), oma=c(3.5,0,.5,0), mgp=c(2,.6,0))
plot(sunspotz, type='n', xaxt='no', ylab='')
  grid(lty=1, col=gray(.9))
  lines(sunspotz)
plot(sunspotz, type='n', ylab='')
  grid(lty=1, col=gray(.9))
  lines(sunspotz)
title(xlab="Time", outer=TRUE, cex.lab=1.2)
mtext(side=2, "Sunspot Numbers", line=2, adj=.75)
```

The result is shown in Figure R.3. The top plot is wide and narrow, revealing the fact that the series rises quickly ↑ and falls slowly ↘. The bottom plot, which is more square, obscures this fact. You will notice that in the main part of the text, we never plotted a series in a square box. The ideal shape for plotting time series, in most instances, is when the time axis is much wider than the value axis.

▶ Exercise 15: There is an R data set called `lynx` that is the annual numbers of lynx trappings for 1821–1934 in Canada. The data are typical of predator-prey series. Produce two graphs in a multifigure plot, one of the sunspot numbers, and one of the lynx series. What attribute does the lynx plot reveal?
Solution: We'll get you started. Are the data doing this: ↑↘ as the sunspot numbers, or is are they doing this: ↗↓?

```
par(mfrow=c(2,1))
tsplot(sunspotz)
tsplot( ___ )
```

Note: Any resizing command such as `dev.new(height=8.75)` does not work with RStudio. Their official statement is:

> Unfortunately there's no way to explicitly set the plot pane size itself right now - however, you can explicitly set the size of a plot you're saving using the Export Plot feature of the Plots pane. Choose Save Plot as PDF or Image and it will give you an option to set the size of the plot by pixel or inch size.

Because size matters when plotting time series, producing graphs interactively in RStudio can be a bit of a pain.

Index